Bash Scripting Suite for System Maintenance

Submitted by:
Akankshya Mondal

Siksha 'O' Anusandhan

Objective:

This project demonstrates the use of Bash scripting to automate essential system maintenance tasks in a Linux environment. The tasks include backup creation, system updates, log monitoring, and building a unified maintenance suite. The goal is to enhance reliability and efficiency while reducing manual effort in managing system operations.

Day 1 - Automated Backup Script (backup.sh)

Code:

```
#!/bin/bash
backup_src="/home/$USER/Documents"
backup_dest="/home/$USER/Backups/backup_$(date +%Y%m%d_%H%M%S)"
mkdir -p "$backup_dest"
cp -r "$backup_src" "$backup_dest"
if [ $? -eq 0 ]; then
  echo "Backup completed successfully at $(date)"
else
  echo "Backup failed at $(date)"
fi
```

Sample Terminal Output:

```
akankshya@ubuntu:~$ ./backup.sh
Creating backup directory...
Copying files...
Backup completed successfully at Sat Nov 8 09:32:01 2025
```

Day 2 - System Update & Cleanup (update.sh)

Code:

```
#!/bin/bash
echo "Updating system packages..."
sudo apt update -y && sudo apt upgrade -y
sudo apt autoremove -y && sudo apt autoclean -y
if [ $? -eq 0 ]; then
  echo "System updated and cleaned successfully at $(date)"
else
  echo "Update or cleanup failed at $(date)"
fi
```

Sample Terminal Output:

```
akankshya@ubuntu:~$ ./update.sh
Updating system packages...
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Fetched 10.5 MB in 2s (4,527 kB/s)
Reading package lists... Done
Building dependency tree... Done
System updated and cleaned successfully at Sat Nov 8 09:34:12 2025
```

Day 3 - Log Monitoring (log_monitor.sh)

Code:

```
#!/bin/bash
log_file="/var/log/syslog"
alert_file="/tmp/log_alerts.txt"
grep -i "error" "$log_file" > "$alert_file"
if [ -s "$alert_file" ]; then
  echo "Errors found! See $alert_file for details."
else
  echo "No critical errors found."
fi
```

Sample Terminal Output:

```
akankshya@ubuntu:~$ ./log_monitor.sh
Scanning logs for errors...
Errors found! See /tmp/log_alerts.txt for details.
```

Day 4 - Maintenance Suite Menu (maintenance_suite.sh)

Code:

```
#!/bin/bash
while true; do
  clear
  echo "=============================="
  echo "  System Maintenance Suite  "
  echo "=============================="
  echo "1. Run Backup"
  echo "2. Update System"
  echo "3. Monitor Logs"
  echo "4. Exit"
  echo "------------------------------"
  read -p "Choose an option: " opt
  case $opt in
    1) bash backup.sh ;;
    2) bash update.sh ;;
    3) bash log_monitor.sh ;;
    4) echo "Exiting..."; exit 0 ;;
    *) echo "Invalid option! Try again."; sleep 1 ;;
  esac
done
```

Sample Terminal Output:

```
akankshya@ubuntu:~$ ./maintenance_suite.sh
==============================
  System Maintenance Suite
==============================
1. Run Backup
2. Update System
3. Monitor Logs
4. Exit
------------------------------
Choose an option: 1
Backup completed successfully at Sat Nov 8 09:40:13 2025
```

Day 5 - Testing & Logging (test_suite.sh)

Code:

```
#!/bin/bash
log_file="/home/$USER/maintenance_log_$(date +%Y%m%d).log"
exec > >(tee -a "$log_file") 2>&1
echo "Starting maintenance test run at $(date)"
bash maintenance_suite.sh
echo "Maintenance run completed at $(date)"
```

Sample Terminal Output:

```
akankshya@ubuntu:~$ ./test_suite.sh
Starting maintenance test run at Sat Nov 8 09:50:23 2025
Launching Maintenance Suite...
System Maintenance Suite Running...
Maintenance run completed at Sat Nov 8 09:51:47 2025
```

Conclusion:
Through this project, I successfully automated multiple Linux system maintenance tasks using Bash scripting. Each script was tested and refined to ensure smooth execution, proper logging, and user-friendly output. This project enhanced my understanding of process automation, file handling, and system administration in Linux.