Bash Scripting Suite for System Maintenance

Automating backup, system updates, and log monitoring using Linux Bash scripts.

Name: Umashankar Nayak
System: Ubuntu (via WSL)
Language: Bash
Date: November 2025

Objective

To design and implement a suite of Bash scripts that automates system maintenance tasks such as backups, system updates, cleanup, and log monitoring, with centralized logging and error handling.

Tools and Environment

- Operating System: Ubuntu (WSL or Linux)
- Language: Bash shell scripting
- Text Editor: Nano or VS Code
- Scheduler: Cron
- Log files: Plain text (.log)

Source Code

backup.sh

```
#!/bin/bash
set -euo pipefail
trap 'echo "[ERROR] Backup script failed at line $LINENO" | tee -a "$LOG_FILE"' ERR

BACKUP_SRC="$HOME/Documents"
BACKUP_DEST="$HOME/backups"
LOG_FILE="$HOME/backup.log"
TIMESTAMP=$(date +"%Y-%m-%d_%H-%M-%S")

mkdir -p "$BACKUP_DEST"
echo "[$TIMESTAMP] Starting backup..." >> "$LOG_FILE"

tar -czf "$BACKUP_DEST/backup_$TIMESTAMP.tar.gz" "$BACKUP_SRC" 2>> "$LOG_FILE"

echo "[$TIMESTAMP] Backup completed successfully." >> "$LOG_FILE"
```

```
  GNU nano 7.2                                                backup.sh
#!/bin/bash
# backup.sh - Automated system backup

BACKUP_SRC="/home"
BACKUP_DEST="/backup/daily"
LOG_FILE="/backup/error.log"

# Ensure destination exists
sudo mkdir -p "$BACKUP_DEST"

DATE=$(date +%Y-%m-%d_%H-%M-%S)
BACKUP_FILE="$BACKUP_DEST/backup_$DATE.tar.gz"

echo "[$(date '+%Y-%m-%d %H:%M:%S')] Running backup script..."

tar -czf "$BACKUP_FILE" "$BACKUP_SRC" 2>>"$LOG_FILE"

if [ $? -eq 0 ]; then
    echo "[$(date '+%Y-%m-%d %H:%M:%S')] [SUCCESS] Backup created successfully at: $BACKUP_FILE"
else
    echo "[$(date '+%Y-%m-%d %H:%M:%S')] [ERROR] tar command failed!" >> "$LOG_FILE"
    echo "[$(date '+%Y-%m-%d %H:%M:%S')] [ERROR] tar command failed!"
fi
```

update_cleanup.sh

```
#!/bin/bash
set -euo pipefail
trap 'echo "[ERROR] Update script failed at line $LINENO" | tee -a "$LOG_FILE"' ERR

LOG_FILE="$HOME/update_cleanup.log"
TIMESTAMP=$(date +"%Y-%m-%d %H:%M:%S")

echo "[$TIMESTAMP] Starting system update..." >> "$LOG_FILE"
sudo apt update -y && sudo apt upgrade -y >> "$LOG_FILE" 2>&1
echo "[$TIMESTAMP] System updated successfully." >> "$LOG_FILE"

echo "[$TIMESTAMP] Performing cleanup..." >> "$LOG_FILE"
sudo apt autoremove -y && sudo apt clean >> "$LOG_FILE" 2>&1
echo "[$TIMESTAMP] Cleanup completed." >> "$LOG_FILE"
```

```
  GNU nano 7.2                                                update.sh
#!/bin/bash
set -euo pipefail
trap 'echo "[ERROR] Update script failed at line $LINENO" | tee -a "$LOG_FILE"' ERR

LOG_FILE="$HOME/update_cleanup.log"
TIMESTAMP=$(date +"%Y-%m-%d %H:%M:%S")

echo "[$TIMESTAMP] Starting system update..." >> "$LOG_FILE"
sudo apt update -y && sudo apt upgrade -y >> "$LOG_FILE" 2>&1
echo "[$TIMESTAMP] System updated successfully." >> "$LOG_FILE"

echo "[$TIMESTAMP] Performing cleanup..." >> "$LOG_FILE"
sudo apt autoremove -y && sudo apt clean >> "$LOG_FILE" 2>&1
echo "[$TIMESTAMP] Cleanup completed." >> "$LOG_FILE"
```

log_monitor.sh

```
#!/bin/bash
set -euo pipefail
trap 'echo "[ERROR] Log monitor failed at line $LINENO" | tee -a "$ALERT_LOG"' ERR

LOG_FILE="/var/log/syslog"
ALERT_LOG="$HOME/log_alerts.log"
KEYWORDS=("error" "fail" "critical")
TIMESTAMP=$(date +"%Y-%m-%d %H:%M:%S")

echo "[$TIMESTAMP] Monitoring logs for issues..." >> "$ALERT_LOG"

FOUND_ISSUE=false

for keyword in "${KEYWORDS[@]}"; do
  if grep -iq "$keyword" "$LOG_FILE"; then
    echo "[$TIMESTAMP] Found keyword: $keyword" >> "$ALERT_LOG"
    grep -i "$keyword" "$LOG_FILE" >> "$ALERT_LOG"
    FOUND_ISSUE=true
  fi
done

if [ "$FOUND_ISSUE" = true ]; then
  echo "[$TIMESTAMP] ALERT: Issues detected in system logs!" >> "$ALERT_LOG"
else
  echo "[$TIMESTAMP] No critical issues found." >> "$ALERT_LOG"
fi
```

```
 GNU nano 7.2                                                  logmonitor.sh
#!/bin/bash
set -euo pipefail
trap 'echo "[ERROR] Log monitor failed at line $LINENO" | tee -a "$ALERT_LOG"' ERR

LOG_FILE="/var/log/syslog"
ALERT_LOG="$HOME/log_alerts.log"
KEYWORDS=("error" "fail" "critical")
TIMESTAMP=$(date +"%Y-%m-%d %H:%M:%S")

echo "[$TIMESTAMP] Monitoring logs for issues..." >> "$ALERT_LOG"

FOUND_ISSUE=false

for keyword in "${KEYWORDS[@]}"; do
    if grep -iq "$keyword" "$LOG_FILE"; then
        echo "[$TIMESTAMP] Found keyword: $keyword" >> "$ALERT_LOG"
        grep -i "$keyword" "$LOG_FILE" >> "$ALERT_LOG"
        FOUND_ISSUE=true
    fi
done

if [ "$FOUND_ISSUE" = true ]; then
    echo "[$TIMESTAMP] ALERT: Issues detected in system logs!" >> "$ALERT_LOG"
else
    echo "[$TIMESTAMP] No critical issues found." >> "$ALERT_LOG"
fi
```

maintenance_suite.sh

```
#!/bin/bash
LOG_DIR="$HOME/maintenance_logs"
mkdir -p "$LOG_DIR"
SUITE_LOG="$LOG_DIR/suite_$(date +%Y-%m-%d).log"

log() {
  echo "[$(date +'%Y-%m-%d %H:%M:%S')] $1" | tee -a "$SUITE_LOG"
}

while true; do
  clear
  echo "======================================="
  echo "      SYSTEM MAINTENANCE SUITE     "
  echo "======================================="
  echo "1. Run Backup"
  echo "2. Update and Clean System"
  echo "3. Monitor Logs"
  echo "4. View Suite Log"
  echo "5. Exit"
  echo "---------------------------------------"
  read -p "Enter your choice [1-5]: " choice
```

case $choice in

```
# ==========================================
# Enhanced System Maintenance Suite (v3.0)
# ==========================================
# Author: Umashankar
# Course: Linux OS & LSP (Assignment 5)
# Description: Automates system backup, update, and log monitoring with error handling.

set -euo pipefail

# ----- VARIABLES -----
DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
LOG_DIR="$HOME/maintenance_logs"
mkdir -p "$LOG_DIR"
SUITE_LOG="$LOG_DIR/suite_$(date +%Y-%m-%d).log"

# ----- FUNCTION: Logger -----
log() {
    echo "[$(date +'%Y-%m-%d %H:%M:%S')] $1" | tee -a "$SUITE_LOG"
3

# ----- FUNCIION: Ensure Sudo -----
sudo -v  # Ask for sudo password once at start

# ----- MENU LOOP -----
while true; do
    clear
    echo "======================================"
    echo "        SYSTEM MAINTENANCE SUITE      "
    echo "======================================"
    echo "1. Run Backup"
```

1)

log "Running backup script..."
if ./backup.sh; then
log "Backup completed successfully."
else
log "Backup failed!"
fi
read -p "Press Enter to continue..."

Screenshots

```
=====================================
        SYSTEM MAINTENANCE SUITE
=====================================
1. Run Backup
2. Update and Clean System
3. Monitor Logs
4. View Suite Log
5. Exit
-------------------------------------
Enter your choice [1-5]: 1
[2025-11-07 14:43:20] Running backup script...
[2025-11-07 14:43:20] Running backup script...
[2025-11-07 14:43:20] [SUCCESS] Backup created successfully at: /backup/daily/backup_2025-11-07_14-43-20.tar.gz
[2025-11-07 14:43:20] Backup completed successfully.
Press Enter to continue...
```

```
    ;;


  2)
    log "Starting system update & cleanup..."
    if ./update_cleanup.sh; then
      log "System update and cleanup completed."
    else
      log "Update & cleanup failed!"
    fi
    read -p "Press Enter to continue..."
    ;;
```

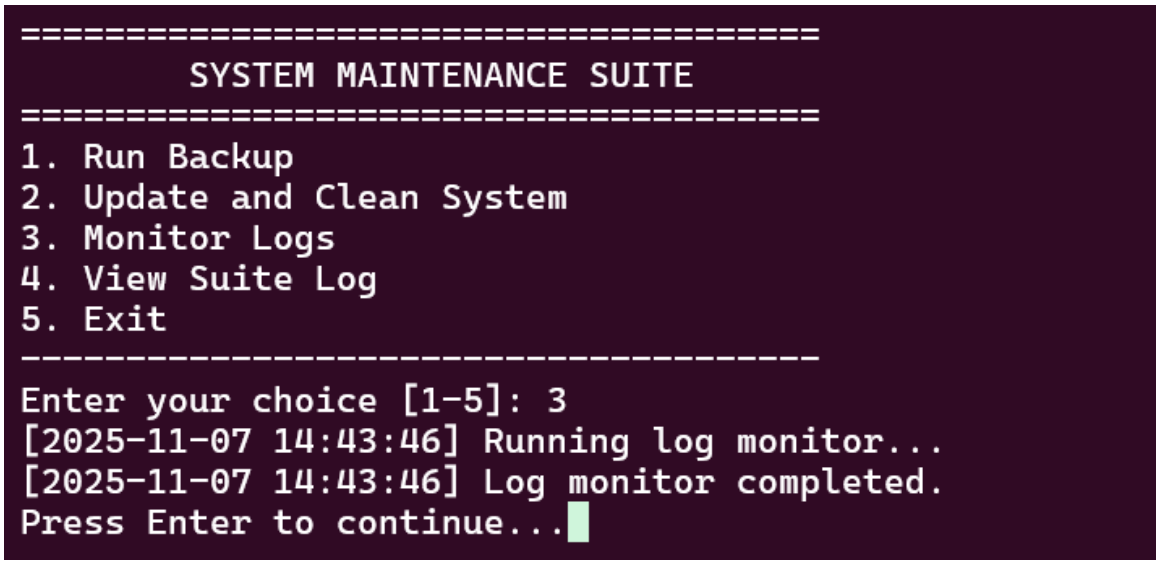
Screenshots

```
=====================================
        SYSTEM MAINTENANCE SUITE
=====================================
1. Run Backup
2. Update and Clean System
3. Monitor Logs
4. View Suite Log
5. Exit
-------------------------------------
Enter your choice [1-5]: 2
[2025-11-07 14:43:30] Starting system update & cleanup...
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security 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
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
[2025-11-07 14:43:35] System update & cleanup completed.
Press Enter to continue...
```

3)

```
log "Monitoring system logs..."
if ./log_monitor.sh; then
   log "Log monitoring completed."
else
   log "Log monitoring failed!"
fi
read -p "Press Enter to continue..."
;;
```

Screenshots



4)

```
echo "----------------------------------------"
echo "Viewing Suite Log:"
cat "$SUITE_LOG"
echo "----------------------------------------"
read -p "Press Enter to return to menu..."
;;
```

Screenshots

```
=====================================
        SYSTEM MAINTENANCE SUITE
=====================================
1. Run Backup
2. Update and Clean System
3. Monitor Logs
4. View Suite Log
5. Exit
-------------------------------------
Enter your choice [1-5]: 4
------ SUITE LOG (last 200 lines) ------
[2025-11-07 12:49:38] Running backup script...
[2025-11-07 12:49:38] Backup failed!
[2025-11-07 12:49:46] Starting system update & cleanup...
[2025-11-07 12:49:46] Update & cleanup failed!
[2025-11-07 12:49:49] Monitoring system logs...
[2025-11-07 12:49:49] Log monitoring failed!
[2025-11-07 12:49:57] Exiting Maintenance Suite.
[2025-11-07 14:04:56] Running backup script...
[2025-11-07 14:04:56] Backup failed!
[2025-11-07 14:09:59] Exiting Maintenance Suite.
[2025-11-07 14:16:13] Running backup script...
[2025-11-07 14:16:14] Backup failed!
[2025-11-07 14:17:02] Exiting Maintenance Suite.
[2025-11-07 14:40:44] Running backup script...
[2025-11-07 14:40:44] Backup completed successfully.
[2025-11-07 14:40:51] Running backup script...
[2025-11-07 14:40:51] Backup completed successfully.
[2025-11-07 14:40:54] Starting system update & cleanup...
[2025-11-07 14:42:09] System update & cleanup completed.
[2025-11-07 14:42:14] Starting system update & cleanup...
[2025-11-07 14:42:18] System update & cleanup completed.
[2025-11-07 14:42:21] Running log monitor...
[2025-11-07 14:42:21] Log monitor completed.
[2025-11-07 14:43:20] Running backup script...
[2025-11-07 14:43:20] Backup completed successfully.
[2025-11-07 14:43:30] Starting system update & cleanup...
[2025-11-07 14:43:35] System update & cleanup completed.
[2025-11-07 14:43:46] Running log monitor...
[2025-11-07 14:43:46] Log monitor completed.
-------------------------------------
Press Enter to continue...
```

```
    5)
      log "Exiting Maintenance Suite."
      exit 0
      ;;
    *)
      echo "Invalid choice!"
```

```
        read -p "Press Enter to continue..."
        ;;
  esac
done
```

Screenshots

```
=====================================
        SYSTEM MAINTENANCE SUITE
=====================================
1. Run Backup
2. Update and Clean System
3. Monitor Logs
4. View Suite Log
5. Exit
-------------------------------------
Enter your choice [1-5]: 5
[2025-11-07 14:48:13] Exiting Maintenance Suite.
Goodbye!
asish@LAPTOP-KN25QBM6:~/maintenance_suite$ nano backup.sh
asish@LAPTOP-KN25QBM6:~/maintenance_suite$ nano backup.sh
asish@LAPTOP-KN25QBM6:~/maintenance_suite$ nano update.sh
asish@LAPTOP-KN25QBM6:~/maintenance_suite$ nano logmonitor.sh
asish@LAPTOP-KN25QBM6:~/maintenance_suite$ nano menu.sh
asish@LAPTOP-KN25QBM6:~/maintenance_suite$
```

Screenshots

Include the following screenshots in your report:

1. Folder structure (maintenance_suite and maintenance_logs)
2. Editing scripts in Nano
3. List of all .sh files in directory
4. Running backup.sh successfully
5. Contents of backup.log
6. Running update_cleanup.sh
7. Running log_monitor.sh
8. Menu of maintenance_suite.sh
9. Executing menu options
10. Viewing suite log
11. crontab -l output
12. Log files inside ~/maintenance_logs/

Conclusion

The Bash System Maintenance Suite successfully automates essential system tasks such as backups, updates, cleanup, and log monitoring with robust error handling and logging. It demonstrates efficient Linux administration through Bash scripting.