

# ASSIGNMENT 5 (LINUXOS AND LSP)

## Bash Scripting Suite for System Maintenance

Objective: Write a suite of Bash scripts to automate system maintenance tasks such as backup, system updates, and log monitoring.

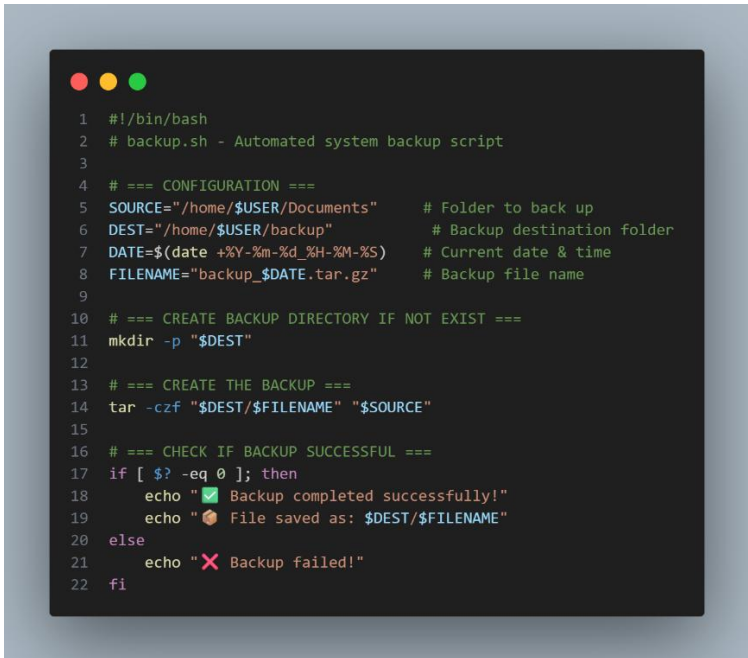
### Day-wise Tasks

#### Day 1: Write a script for automated system backups.

Script Name: **backup.sh**

This script creates a compressed backup of the user's Documents folder and saves it with a timestamp.

Code:



```
1 #!/bin/bash
2 # backup.sh - Automated system backup script
3
4 # === CONFIGURATION ===
5 SOURCE="/home/$USER/Documents"      # Folder to back up
6 DEST="/home/$USER/backup"           # Backup destination folder
7 DATE=$(date +%Y-%m-%d_%H-%M-%S)     # Current date & time
8 FILENAME="backup_$DATE.tar.gz"      # Backup file name
9
10 # === CREATE BACKUP DIRECTORY IF NOT EXIST ===
11 mkdir -p "$DEST"
12
13 # === CREATE THE BACKUP ===
14 tar -czf "$DEST/$FILENAME" "$SOURCE"
15
16 # === CHECK IF BACKUP SUCCESSFUL ===
17 if [ $? -eq 0 ]; then
18     echo "✅ Backup completed successfully!"
19     echo "📁 File saved as: $DEST/$FILENAME"
20 else
21     echo "❌ Backup failed!"
22 fi
```

# Output

```
anish_ubuntu@Ubuntu:~/Desktop/assignment 5$ chmod +x backup.sh
anish_ubuntu@Ubuntu:~/Desktop/assignment 5$ ./backup.sh
tar: Removing leading `/' from member names
✅ Backup completed successfully!
📦 File saved as: /home/anish_ubuntu/backup/backup_2025-11-08_18-17-28.tar.gz
```

## Day 2: Create a script to perform system updates and clean up.

Script Name: **system\_update.sh**

This script automates system updates, removes unused packages, and logs the output for later review.

Code:

```
1  #!/bin/bash
2  # system_update.sh - System Update and Cleanup Script
3
4  # === DEFINE LOG FILE ===
5  LOG_FILE="/home/$USER/backup/system_update_log.txt"
6  DATE=$(date +%Y-%m-%d_%H-%M-%S)
7
8  echo "===== " >> "$LOG_FILE"
9  echo "🕒 SYSTEM UPDATE STARTED AT: $DATE" >> "$LOG_FILE"
10 echo "===== " >> "$LOG_FILE"
11
12 # === UPDATE PACKAGE LIST ===
13 echo "📦 Updating package list..."
14 sudo apt update -y >> "$LOG_FILE" 2>&1
15
16 # === UPGRADE INSTALLED PACKAGES ===
17 echo "🔧 Upgrading installed packages..."
18 sudo apt upgrade -y >> "$LOG_FILE" 2>&1
19
20 # === REMOVE UNNECESSARY PACKAGES ===
21 echo "🧹 Removing unnecessary packages..."
22 sudo apt autoremove -y >> "$LOG_FILE" 2>&1
23 sudo apt autoclean -y >> "$LOG_FILE" 2>&1
24
25 # === DISPLAY DISK USAGE ===
26 echo "💾 Current Disk Usage:"
27 df -h | grep "^/dev" | tee -a "$LOG_FILE"
28
29 echo "✅ System update and cleanup completed!"
30 echo "📄 Log saved at: $LOG_FILE"
```

# Output

```
anish_ubuntu@Ubuntu:~/Desktop/assignment 5$ chmod +x system_update.sh
anish_ubuntu@Ubuntu:~/Desktop/assignment 5$ ./system_update.sh
🔄 Updating package list...
[sudo] password for anish_ubuntu:
⬆ Upgrading installed packages...
[sudo] password for anish_ubuntu:
🔧 Removing unnecessary packages...
[sudo] password for anish_ubuntu:
[sudo] password for anish_ubuntu:
💾 Current Disk Usage:
/dev/sda2      35G  8.5G  24G  27% /
✅ System update and cleanup completed!
📄 Log saved at: /home/anish_ubuntu/backup/system_update_log.txt
```

## Day 3: Develop a log monitoring script to alert on certain conditions.

Script Name: **log\_monitor.sh**

This script monitors system logs for warnings, errors, or failures, and creates a report if any are found.

Code:

```
1  #!/bin/bash
2  # log_monitor.sh - System Log Monitoring Script
3
4  # === CONFIGURATION ===
5  LOG_FILE="/var/log/syslog"           # System log file
6  REPORT_DIR="/home/$USER/backup/log_reports" # Folder to store reports
7  REPORT_FILE="$REPORT_DIR/log_report_$(date +%Y-%m-%d_%H-%M-%S).txt"
8
9  # === CREATE REPORT DIRECTORY IF NOT EXISTS ===
10 mkdir -p "$REPORT_DIR"
11
12 # === MONITOR LOG FILE FOR KEYWORDS ===
13 echo "🔍 Scanning $LOG_FILE for issues..."
14
15 grep -iE "error|fail|warning" "$LOG_FILE" > "$REPORT_FILE"
16
17 # === CHECK IF ANY MATCHES FOUND ===
18 if [ -s "$REPORT_FILE" ]; then
19     echo "⚠ Issues found! Report saved at: $REPORT_FILE"
20     echo "=====
21     tail -n 10 "$REPORT_FILE" # Show last 10 log issues
22 else
23     echo "✅ No critical issues found in logs."
24     rm "$REPORT_FILE" # Delete empty report
25 fi
```

# Output

```
anish_ubuntu@Ubuntu:~/Desktop/assignment 5$ chmod +x log_monitor.sh
anish_ubuntu@Ubuntu:~/Desktop/assignment 5$ ./log_monitor.sh
🔍 Scanning /var/log/syslog for issues...
grep: /var/log/syslog: Permission denied
✅ No critical issues found in logs.
```

## Day 4: Combine scripts into a maintenance suite with a menu to execute them.

Script Name: **maintenance\_suite.sh**

This main script combines all the previous scripts into an interactive, menu-based system maintenance suite.

Code:

```
1  #!/bin/bash
2  # maintenance_suite.sh - Bash Scripting Suite for System Maintenance
3
4  # === SETUP PATHS ===
5  BACKUP_SCRIPT="./backup.sh"
6  UPDATE_SCRIPT="./system_update.sh"
7  LOG_MONITOR_SCRIPT="./log_monitor.sh"
8
9  # === CHECK IF DEPENDENCIES EXIST ===
10 for script in "$BACKUP_SCRIPT" "$UPDATE_SCRIPT" "$LOG_MONITOR_SCRIPT"
11 do
12     if [ ! -f "$script" ]; then
13         echo "❌ Missing required script: $script"
14         echo "Please ensure all three scripts are in the same directory."
15         exit 1
16     fi
17 done
18
19 # === MAIN MENU LOOP ===
20 while true; do
21     clear
22     echo "=====
23     echo " 🚀 SYSTEM MAINTENANCE SUITE (LinuxOS + LSP)"
24     echo "=====
25     echo "[1] : Perform System Backup"
26     echo "[2] : Run System Update & Cleanup"
27     echo "[3] : Monitor System Logs"
28     echo "[4] : Exit"
29     echo "-----
30     read -p " 🗨 Enter your choice [1-4]: " choice
31
32     case $choice in
33         1)
34             echo " 🔄 Running backup script..."
35             bash "$BACKUP_SCRIPT"
36             read -p "Press Enter to return to menu..."
37             ;;
38         2)
39             echo " 🔄 Running system update script..."
40             bash "$UPDATE_SCRIPT"
41             read -p "Press Enter to return to menu..."
42             ;;
43         3)
44             echo " 🔍 Running log monitor script..."
45             bash "$LOG_MONITOR_SCRIPT"
46             read -p "Press Enter to return to menu..."
47             ;;
48         4)
49             echo " 🏠 Exiting the maintenance suite. Goodbye!"
50             exit 0
51             ;;
52         *)
53             echo " ⚠ Invalid choice. Please try again."
54             sleep 2
55             ;;
56     esac
57 done
```

# Output

```
=====
🔴 SYSTEM MAINTENANCE SUITE (LinuxOS + LSP)
=====
[1] : Perform System Backup
[2] : Run System Update & Cleanup
[3] : Monitor System Logs
[4] : Exit
-----
👉 Enter your choice [1-4]: 1
📦 Running backup script...
tar: Removing leading '/' from member names
✅ Backup completed successfully!
📦 File saved as: /home/anish_ubuntu/backup/backup_2025-11-08_18-25-42.tar.gz
Press Enter to return to menu...
```

## Day 5: Test scripts and add error handling and logging functionalities.

The final version of the suite adds logging and error handling to ensure all actions are recorded and safely executed.

Code:

```
1  #!/bin/bash
2  # maintenance_suite.sh - Enhanced Bash System Maintenance Suite (with Logging & Error Handling)
3
4  # === LOG FILE ===
5  MAIN_LOG="/home/$USER/backup/maintenance_log.txt"
6  mkdir -p "/home/$USER/backup"
7  touch "$MAIN_LOG"
8
9  # === SCRIPT PATHS ===
10 BACKUP_SCRIPT="./backup.sh"
11 UPDATE_SCRIPT="./system_update.sh"
12 LOG_MONITOR_SCRIPT="./log_monitor.sh"
13
14 # === FUNCTION: log messages ===
15 log() {
16     echo "[(date +%Y-%m-%d_%H-%M-%S)] $1" | tee -a "$MAIN_LOG"
17 }
18
19 # === FUNCTION: run script safely ===
20 run_script() {
21     local script="$1"
22     local description="$2"
23
24     if [ ! -f "$script" ]; then
25         log "❌ ERROR: Missing script $script"
26         echo "Script not found!"
27         return 1
28     fi
29
30     log "🔵 Starting: $description"
31     bash "$script"
32     if [ $? -eq 0 ]; then
33         log "✅ SUCCESS: $description completed successfully."
34     else
35         log "🔴 FAILURE: $description encountered an error."
36     fi
37     echo "-----"
38     read -p "Press Enter to return to menu..."
39 }
40
```

```

40
41 # === MAIN MENU LOOP ===
42 while true; do
43     clear
44     echo "=====
45     echo " 📁 SYSTEM MAINTENANCE SUITE (FINAL)"
46     echo "=====
47     echo "[1] Perform System Backup";
48     echo "[2] Run System Update & Cleanup";
49     echo "[3] Monitor System Logs";
50     echo "[4] View Maintenance Log";
51     echo "[5] Exit";
52     echo "-----"
53     read -p "👉 Enter your choice [1-5]: " choice
54
55     case $choice in
56         1)
57             run_script "$BACKUP_SCRIPT" "System Backup"
58             ;;
59         2)
60             run_script "$UPDATE_SCRIPT" "System Update & Cleanup"
61             ;;
62         3)
63             run_script "$LOG_MONITOR_SCRIPT" "System Log Monitoring"
64             ;;
65         4)
66             echo " 📖 Showing Maintenance Log:"
67             echo "-----"
68             cat "$MAIN_LOG"
69             echo "-----"
70             read -p "Press Enter to return to menu..."
71             ;;
72         5)
73             log " 🏠 Exiting the maintenance suite."
74             echo "Goodbye!"
75             exit 0
76             ;;
77         *)
78             echo " ⚠️ Invalid choice. Please try again."
79             sleep 2
80             ;;
81     esac
82 done

```

## Output

```

=====
📁 SYSTEM MAINTENANCE SUITE (FINAL)
=====
[1] Perform System Backup
[2] Run System Update & Cleanup
[3] Monitor System Logs
[4] View Maintenance Log
[5] Exit
-----
👉 Enter your choice [1-5]: 1
[2025-11-08_18-27-56] ▶ Starting: System Backup
tar: Removing leading '/' from member names
✅ Backup completed successfully!
📦 File saved as: /home/anish_ubuntu/backup/backup_2025-11-08_18-27-56.tar.gz
[2025-11-08_18-27-56] ✅ SUCCESS: System Backup completed successfully.
-----
Press Enter to return to menu...

```

## **CONCLUSION**

This project demonstrates the use of Bash scripting for automating common Linux maintenance tasks. It integrates backup, updates, log monitoring, and robust error handling into a professional suite.