Linux Lab Midterm Project

Daily User Log Archiver

Student Name: Priyadarshi Prabhakar
Course: B.Tech CSE � Linux Lab
Project Type: Shell Scripting
Date: October 2025

Goal

Create a **shell script** that:

- Logs current system information (user, date, processes, disk usage)
- Rotates and archives old logs weekly
- Runs automatically every day using a **cron job**

Implementation Details

1. Identify User

The script identifies the user executing it using:

```
whoami
```

Example:

```
echo "User: $(whoami)"
```

2. File Management

Logs are stored in:

```
~/daily_logs/log_YYYY-MM-DD.txt
```

A new file is created every day, containing:

- Current date and time
- Logged-in user
- System uptime
- Top 5 CPU-consuming processes
- Disk usage summary
- Kernel/system messages (via `dmesg`)

3. Archiving

- Logs older than **7 days** are moved to `~/daily_logs/archive`
- Every **Sunday**, the archive directory is compressed into:

```
weekly_logs_YYYY-WW.tar.gz
```

4. Loop and Conditions

The script uses a loop to check each log�s age and move it if older than 7 days:

```
for file in log_*.txt; do
  if [ condition-to-check-age ]; then
    mv "$file" archive/
  fi
done
```

Optimized version uses:

```
find "${LOG_DIR}" -maxdepth 1 -type f -name 'log_*.txt' -mtime +7 -print0 | while IFS= read -r
  mv "$file" "${ARCHIVE_DIR}/"
done
```

Full Shell Script

```
#!/bin/bash
# Daily User Log Archiver - Full Version (Downloads folder, Older than 7 Days)


# --- Step 1: Identify user ---
USER_NAME=$(whoami)
echo "Script run by: $USER_NAME"

# --- Step 2: Create log directory and save daily log ---
LOG_DIR=/home/wizzz/Desktop/linux_lab/dailylogs
mkdir -p "$LOG_DIR"

LOGFILE="$LOG_DIR/log_$(date +%Y-%m-%d).txt"

{
  echo "User: $USER_NAME"
  echo "Date: $(date)"
  echo "----------------------------------------"
  echo "System Uptime:"
  uptime
  echo "----------------------------------------"
  echo "Top 5 CPU Consuming Processes:"
  ps -eo pid,comm,%mem,%cpu --sort=-%cpu | head -n 6
  echo "----------------------------------------"
  echo "Disk Usage:"
  df -h
} > "$LOGFILE"

echo " Daily log saved: $LOGFILE"

# --- Step 3: Weekly archive (Monday only) ---
ARCHIVE_DIR=/home/wizzz/Desktop/linux_lab/dailylogs/archive
mkdir -p "$ARCHIVE_DIR"

DAY_OF_WEEK=$(date +%u)  # 1 = Monday
if [ "$DAY_OF_WEEK" -eq 1 ]; then
  tar -czf "$ARCHIVE_DIR/weeklylogs_$(date +%Y-%m-%d).tar.gz" -C "$LOG_DIR" .
  echo " Weekly archive created."
fi
```

```
# --- Step 4: Move logs older than 7 days ---
for file in "$LOG_DIR"/log_*.txt; do
  if [ -f "$file" ] && [ $(find "$file" -mtime +7) ]; then
    mv "$file" "$ARCHIVE_DIR/"
    echo "Moved $file to archive"
  fi
done
```

Cron Job Setup

To schedule the script to run daily at 8 PM:

```
crontab -e
```

Add the line:

```
0 20 * * * /home/user/daily_log.sh >/dev/null 2>&1
```

Confirm with:

```
crontab -l
```

Directory Structure

```
~/daily_logs/
�
+-- logs/
�   +-- log_2025-10-01.txt
�   +-- log_2025-10-02.txt
�   +-- ...
�
+-- archive/
�   +-- log_2025-09-25.txt
�   +-- ...
�
+-- weekly_logs_2025-42.tar.gz
```

Optional Enhancements

1. **Email Logs Automatically**

   ```
   echo "Log attached" | mail -s "Daily Log" -a "$LOGPATH" user@example.com
   ```

2. **Error Handling**
   - Script checks if directories exist before writing.
   - Uses a lock (`.lock`) to avoid concurrent runs.
3. **Interactive Menu**
   - Case-based menu allows manual logging, archiving, and viewing.

Commands Used

Feature	Command	Purpose
Identify User	`whoami`	Get current username
Date Format	`date +%Y-%m-%d`	Create timestamped filenames
Disk Usage	`df -h`	Human-readable disk usage

Feature	Command	Purpose
Process List	`ps -eo pid,comm,%cpu,%mem --sort=-%cpu	head -n 6`
File Search	`find . -name "log_*.txt" -mtime +7`	Find logs older than 7 days
Archiving	`tar -czf`	Compress weekly logs
Scheduling	`crontab -e`	Automate daily execution

Learning Outcomes

- Practical experience with **shell scripting**, **loops**, and **conditions**
- Understanding of **cron jobs** for automation
- Use of **system commands** (`ps` , `df` , `find` , `tar`)
- Managing **file I/O** and **directory structures**
- Basic **error handling** and **process synchronization**

Conclusion

The *Daily User Log Archiver* efficiently:

- Automates daily system monitoring
- Reduces manual log management
- Demonstrates file handling, archiving, scheduling, and automation concepts

This project encapsulates key Linux administration skills through scripting.