

**Name:** Lakshay Dhanda

**SAP ID:** 590029328

**Date:** 2025-11-09

---

## AIM

To create and schedule a shell script that:

- Logs system information daily
  - Stores logs in a dedicated folder
  - Rotates logs older than 7 days
  - Runs automatically using **cron**
- 

## Requirements

- Any Linux distribution (e.g., Linux Mint / Ubuntu / Pop!\_OS)
  - Text editor (Nano, Vim, VS Code, etc.)
  - Cron service enabled
- 

## Theory

System administrators often automate logs to track performance and system health. This experiment focuses on:

1. Collecting system data: processes, memory, disk usage, user info
  2. Saving daily logs with date-based filenames
  3. Deleting old logs automatically
  4. Scheduling tasks using **cron**
- 

## Exercise 1: Creating the Daily Log Script

### Task:

Create a Bash script that saves system info and rotates logs older than 7 days.

### Script (**daily\_log.sh**):

```
#!/bin/bash

LOG_DIR="$HOME/daily_logs"
mkdir -p "$LOG_DIR"

# Create a log file with today's date
LOG_FILE="$LOG_DIR/log_$(date +"%Y-%m-%d").txt"
```

```

# Record system information
{
    echo "===== Daily System Log ====="
    echo "Date: $(date)"
    echo "User: $USER"
    echo

    echo "===== Top 5 Running Processes (by CPU) ====="
    ps -eo pid,user,comm,%cpu,%mem --sort=-%cpu | head -n 6
    echo

    echo "===== Disk Usage ====="
    df -h
    echo

    echo "===== Memory Usage ====="
    free -h
} > "$LOG_FILE"

# Delete logs older than 7 days
find "$LOG_DIR" -type f -name "log_*.txt" -mtime +7 -exec rm {} \;

echo "Log saved to: $LOG_FILE"

```

## Output Screenshot:

```

lakshay42@lakshay42-VirtualBox:~$ ./daily_log.sh
GNU nano 7.2
#!/bin/bash

LOG_DIR="$HOME/daily_logs"
mkdir -p "$LOG_DIR"

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

{
    echo "===== Daily System Log ====="
    echo "Date: $(date)"
    echo "User: $USER"
    echo
    echo "===== Top 5 Running Processes (by CPU) ====="
    ps -eo pid,user,comm,%cpu,%mem --sort=-%cpu | head -n 6
    echo
    echo "===== Disk Usage ====="
    df -h
    echo
    echo "===== Memory Usage ====="
    free -h
} > "$LOG_FILE"

find "$LOG_DIR" -type f -name "log_*.txt" -mtime +7 -exec rm {} \;
echo "Log saved to: $LOG_FILE"

```

## Exercise 2: Scheduling the Script Using Cron

### Task:

Schedule the script to run daily.

## Steps:

1. Open crontab:

```
crontab -e
```

2. Add this line:

```
0 20 * * * /home/lakshay42/daily_log.sh
```

This runs the script.

## Cron Format Reminder:

```
m h dom mon dow command
```

## Cron Screenshot:

The screenshot shows a terminal window titled 'lakshay42@lakshay42-VirtualBox: ~'. The command entered is:

```
0 20 * * * /home/lakshay42/daily_log.sh
```

The terminal also displays the man page for crontab(5), which includes the following text about the cron format:

```
Each task to run has to be defined through a single line indicating with different fields when the task will be run and what command to run for the task. To define the time you can provide concrete values for minute (m), hour (h), day of month (dom), month (mon), and day of week (dow) or use '*' in these fields (for 'any'). Notice that tasks will be started based on the cron's system daemon's notion of time and timezone. Output of the crontab jobs (including errors) is sent through email to the user the crontab file belongs to (unless redirected). For example, you can run a backup of all your user accounts at 5 a.m every week with:
```

```
0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
```

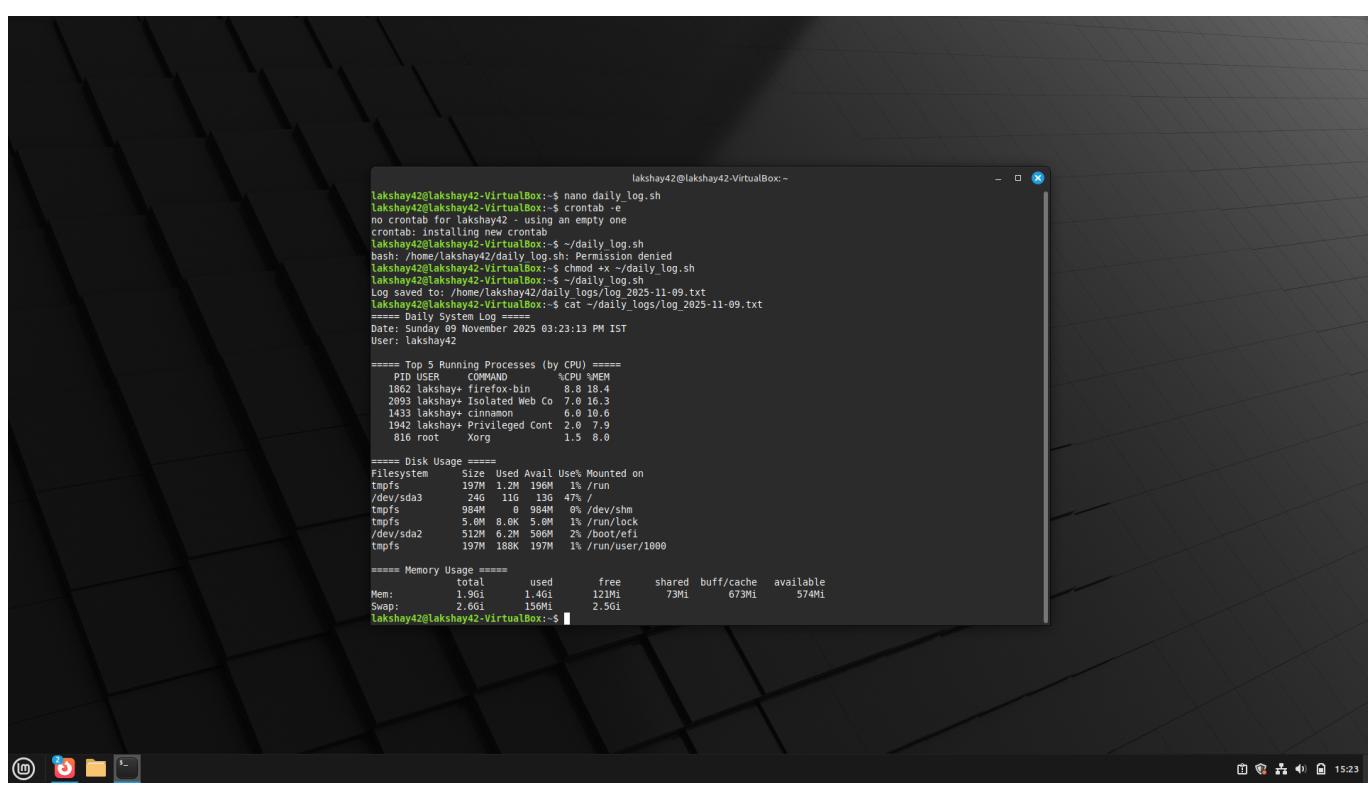
For more information see the manual pages of crontab(5) and cron(8).

At the bottom of the terminal window, there is a menu bar with various icons and keyboard shortcuts for navigating the terminal.

## Final Output

The script successfully:

- Creates a new log file daily
- Shows running processes, memory usage, disk usage
- Deletes old logs (older than 7 days)
- Runs automatically via cron



A screenshot of a Linux desktop environment showing a terminal window. The terminal displays the output of a shell script named 'daily\_log.sh'. The script performs several tasks:

- Creates a new log file daily.
- Show running processes, memory usage, and disk usage.
- Deletes old logs (older than 7 days).
- Runs automatically via cron.

The terminal output includes:

```
lakshay42@lakshay42-VirtualBox:~$ nano daily_log.sh
lakshay42@lakshay42-VirtualBox:~$ crontab -e
no crontab for lakshay42 - using an empty one
crontab: installing new crontab
lakshay42@lakshay42-VirtualBox:~$ ~/daily_log.sh
bash: /home/lakshay42/daily_log.sh: Permission denied
lakshay42@lakshay42-VirtualBox:~$ chmod +x ~/daily_log.sh
lakshay42@lakshay42-VirtualBox:~$ ~/daily_log.sh
Log saved to: /home/lakshay42/daily_logs/log_2025-11-09.txt
lakshay42@lakshay42-VirtualBox:~$ cat ~/daily_logs/log_2025-11-09.txt
=====
Date: Sunday 09 November 2025 03:23:13 PM IST
User: lakshay42

===== Top 5 Running Processes (by CPU) =====
 PID USER      COMMAND   %CPU %MEM
 1862 lakshay+ firefox-bin    8.8 18.4
 2093 lakshay+ Isolated Web Co  7.0 16.3
 1433 lakshay+ cinnamon    6.0 10.6
 1942 lakshay+ Privileged Cont  2.0  7.9
 816 root      Xorg      1.5  8.0

===== Disk Usage =====
Filesystem      Size  Used Avail Use% Mounted on
tmpfs          19M   1M  18M   5% /run
/dev/sda3       24G   1G  13G  47% /
tmpfs          984M   0  984M  0% /dev/shm
tmpfs          8.8M   8.8K  5.8M  1% /run/lock
/dev/sda2       512M   6.1M  506M  2% /boot/efi
tmpfs          197M  188K  197M  1% /run/user/1000

===== Memory Usage =====
total        used        free     shared  buff/cache available
Mem:      1.96Gi  1.46Gi  121Mi    73Mi   673Mi   574Mi
Swap:     2.66Gi  156Mi   2.56Gi
```

## Conclusion

This experiment demonstrates automation using shell scripting and cron. The daily log script efficiently captures system information and maintains a clean log directory by removing older log files. Cron ensures the script runs consistently without manual intervention.