

Experiment: [Daily System Logger Script]

Name: Tanishq bansal, Roll No.: 590029349 Date: 2025-11-07

AIM:

- To create a shell script that logs current system information, rotates old logs, schedules itself to run daily, and sends the daily log via local email.

Requirements:

- Any Linux Distro (Pop!_OS)
- Any text editor (VS Code, Vim, Nano, etc.)
- Cron service for scheduling
- Postfix/mailutils for local email

Theory:

In system administration, automated logging is crucial for monitoring system performance, diagnosing issues, and maintaining records. This experiment involves:

1. Logging details like username, date, processes, and disk usage.
2. Archiving old logs automatically.
3. Scheduling the script to run daily using **cron**.
4. Sending the daily log via local email to the system mailbox.

Procedure & Observations

Exercise 1: Creating the Daily Log Script

Task Statement:

Write a shell script that logs system info, archives old logs, and emails the daily log.

Explanation:

This script:

- Identifies the current user.
- Creates a directory for storing logs.
- Saves daily logs with timestamps.
- Archives logs older than 7 days.
- Compresses weekly logs on Sundays.
- Sends the daily log via local email.
- Can be scheduled using a cron job.

Command(s):

```

#!/bin/bash

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

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

{
    echo "====="
    echo "System Log for: $(date)"
    echo "User: $(whoami)"
    echo "====="
    echo
    echo "Uptime:"
    uptime
    echo
    echo "Top 5 CPU-consuming processes:"
    ps -eo pid,comm,%mem,%cpu --sort=-%cpu | head -n 6
    echo
    echo "Disk Usage:"
    df -h
} > "$LOG_FILE"

echo "Log created successfully at $LOG_FILE"

find "$LOG_DIR" -name "log_*.txt" -mtime +7 -exec mv {} "$ARCHIVE_DIR" \;

if [ "$(date +%u)" -eq 7 ]; then
    tar -czf "$ARCHIVE_DIR/weeklylogs_$(date +%Y-%m-%d).tar.gz" -C "$ARCHIVE_DIR" .
    echo "Weekly archive created."
fi

if [ -f "$LOG_FILE" ]; then
    mail -s "Daily System Log - $(date +%Y-%m-%d)" drago@localhost < "$LOG_FILE"
    echo "Log emailed to local mailbox: drago@localhost"
else
    echo "No log file found for today!"
fi

```

Output:

```
=====
System Log for: Saturday 5 november 2025 11:52:19 PM IST
User: tanishq
=====
```

Uptime:

23:52:19 up 16 min, 1 user, load average: 0.52, 0.81, 0.90

Top 5 CPU-consuming processes:

PID	COMMAND	%MEM	%CPU
2867	GinHawPro	6.7	45.8
2345	Discord	7.8	25.3
2768	Blender	8.9	7.9
4589	Steam	9.9	7.0

Disk Usage:

Filesystem	Size	Used	Avail	Use%	Mounted on
tmpfs	785M	7.8M	780M	6%	/run
efivarfs	678k	67k	560k	19%	/sys/firmware/efi/efivars
/dev/nvme0n1p8	45G	28G	27G	63%	/
tmpfs	5.7G	0	3.8G	0%	/dev/shm
tmpfs	5.0M	0	7.0G	0%	/run/lock
/dev/nvme0n1p6	789M	300M	780M	46%	/boot/efi
tmpfs	4.3G	0	4.3G	0%	/run/qemu
tmpfs	778M	7.9M	764M	2%	/run/user/1000

Exercise 2: Scheduling the Script

Task Statement: Schedule the above script to run daily using cron.

Explanation: Use crontab to automate the script execution at a fixed time every day.

Command(s):

```
crontab -e
0 8 * * * /home/drago/Desktop/LINUX PRO/MID-SEM/midsem.sh
```

Output:

```
0 8 * * * /home/drago/Desktop/LINUX PRO/MID-SEM/midsem.sh
```

Final Output:

```

tanishq@Tanishq:~$ bash midsem.sh
Log created successfully at /home/ drago /daily_logs/log_2025-10-19.txt
tar: .: file changed as we read it
Weekly archive created.
Log emailed to local mailbox:      @localhost
midsem.sh: line 52: -: command not found
tanishq@Tanishq:~$ mail
"/var/mail/drago" : 1 message 1 new
>N 1 REGIMENTio Sun Oct 19 23:12 39/1303 Daily System Log - 2025-10-19
? 1
Return-Path:
X-Original-To: drago@localhost
Delivered-To: drago@localhost
Received: by pop-os (Postfix, from userid 1000)
           id 3AAD01E252C; Sun, 19 Oct 2025 23:12:17 +0530 (IST)
Subject: Daily System Log - 2025-10-19
To: <tanmay@localhost>
User-Agent: mail (GNU Mailutils 3.14)
Date: Sun, 19 Oct 2025 23:12:17 +0530
Message-Id: <20251019174217.3AAD01E252C@pop-os>
From: REGIMENTio <tanmay@pop-os>

=====
System Log for: Sunday 19 October 2025 11:12:17 PM IST
User: tanmay
=====

Uptime:
23:12:17 up 1:11, 1 user, load average: 0.71, 0.48, 0.39

Top 5 CPU-consuming processes:
 PID COMMAND %MEM %CPU
 3777 Isolated Web Co 6.3 9.1
 3548 firefox-bin 5.9 6.4
 1760 gnome-shell 3.8 4.0
 4616 Discord 5.6 3.2
 2682 code 4.3 3.1

Disk Usage:
Filesystem Size Used Avail Use% Mounted on
tmpfs 772M 2.2M 770M 1% /run
efivarfs 384K 90K 290K 24% /sys/firmware/efi/efivars
/dev/nvme0n1p8 47G 24G 21G 54% /
tmpfs 3.8G 0 3.8G 0% /dev/shm
tmpfs 5.0M 0 5.0M 0% /run/lock
/dev/nvme0n1p6 953M 297M 656M 32% /boot/efi
tmpfs 3.8G 0 3.8G 0% /run/gemu
tmpfs 772M 2.0M 770M 1% /run/user/1000

? q
Saved 1 message in /home/tanmay/mbox
Held 0 messages in /var/mail/tanmay

```

Result: The script successfully logs daily system information, archives logs older than 7 days, compresses weekly logs, and emails the log locally. It runs daily via cron.

Conclusion:

The Daily System Logger script automates log creation, archiving, weekly compression, and local email delivery. It successfully runs daily using cron, demonstrating practical use of shell scripting for system monitoring.

We can simply check the mail by typing mail and typing 1 if its the latest mail on our system., also all of these mails only exist on my system and not on the internet, So even though its sent via email it doesn't go to gmail or the internet it stays on my system