

System Logger Script]

Name: Vaishnavi Kumari, Roll No.:590029048 Date: 2025-11-06

AIM:

- To create a shell script that logs current system information, rotates old logs, and schedules itself to run daily.

Requirements:

- Any Linux Distro
- Any text editor (VS Code, Vim, Nano, etc.)
- Cron service for scheduling

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.

Procedure & Observations

Exercise 1: Creating the Daily Log Script

Task Statement:

Write a shell script that logs system info and handles automatic rotation of old logs.

Explanation:

This script:

- Identifies the current user.
- Creates a directory for storing logs.
- Saves daily logs with timestamps.
- Archives logs older than 7 days.
- Can be scheduled using a cron job.

Command(s):

```
bash#!/bin/bash SYS_D="$HOME/daily_logs" ARCHIVE_DIR="$SYS_D/archive" mkdir -p "$ARCHIVE_DIR"
LOG_FILE="$SYS_D/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"

find "$SYS_D" -name "log_*txt" -mtime +7 -exec mv {} "$ARCHIVE_DIR";
if [ "$(date +%-u)" -eq 7 ]; then tar -czf "$ARCHIVE_DIR/weeklylogs_$(date +%-Y-%m-%d).tar.gz" -C "$ARCHIVE_DIR" . fi
```

Output :

log_2025-1

GNU nano 6.2

System Log for: Saturday 18 October 2025 11:52:19 PM IST

User: Vaishnavi

Uptime:

23:52:19 up 16 min, 1 user, load average: 0.41, 0.78, 0.88

Top 5 CPU-consuming processes:

PID	COMMAND	%EM	%CPU
2859	OtWebEngineProc	8.2	4.57
2505	zapzap	5.3	2.43
4619	discord	6.9	7.0
3259	io.elementary.a	6.1	5.9
2272	firefox-bin	6.1	5.9

Disk Usage:	Size	Used	Avail	Use%	Mounted on
tmpfs	772M	2.2M	770M	1%	/run
efivars	384K	90K	290K	24%	/sys/firmware/e
/dev/nvme0n1p8	47G	24G	21G	54%	/
tmpfs	3.8G	0.0	3.8G	0%	/dev/shm
tmpfs	5.0M	0.0	5.0M	0%	/run/lock
total	85.1G	9.53M	75.7G	11%	/home/vaishnavi

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):

bash

crontab -e 0 8 * * * /home/vaishnavi/midsem.sh

Output :

vaishnavi@pop-os:~



GNU nano 6.2 /tmp/crontab.AOPZPR/crontab
3 * * * /home/vaishnavi/misdem.sh

Edit this file to introduce tasks to be run by cron

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 timezones.

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

[Read 26 lines]

Help ^O Write Out \W Where Is ^K Cut ^J Execute ^J
Exit ^R Read File \U Replace ^U Paste ^I Justify ^C

```

getshrpVLpoer@etshrpVLpoer: ~ 8 push aidcem in
y das! 0t8mmrc.9 Qf RSU /!home/Vaishnavi@aly,logs/log V006-10-9,tst
las, file changed as read 1
Nockly archivd'rented. Vattslavidecalhost
misser.tu lime.st -i command not found
misser.khpp.cat | munt
| I REGIMENT. 1 message 1 new
vaishnav: fo: Sun Oct 09 205 11:12:7 Daily System Log - 2025-10-18

Return-Fath, <vaishnavi>
D21le@etus: hume@warchihost
& Priginal. To ront Subject
Reteeved: By roday @>@to// from userid 1000
Date: @un, to 001150UHTS622 Sun 2025 22:52:17 -0530 (IST)
Subject: @tvehargio@Velive ccru@9@0/
Subject: sietav)and JS. UD@VHoVtl's
Ning Version: N ARCCIMENTEx ve+5828@epop=0s>
Proh: @cEN@trv-ovaishnavi>

Sunday System Log Sunday 18 October 2028 11:12:17 PM

Vaishnavi
User: Vaishnavi

User: Vaishnavi 5:11, 1 wsar. Load average: 0.71, 0.48, 0.29

IOP. S-CPU-consuming processes:
    PIR CONHAND      SME@ XZPU
    hfNe@erghtheprincproc  8.3  ±.57
    la @lanetary-video    0.9   7.0
    fifeox Din        0.1   3.9
    code            6.1   5.9

Disk Usase:  Size  Used  Avail  Use%  Mounted on
tmps       772M  2.2M  700M  7%./run
efivars    36GK  80K  200K  28%./bys/firmware/efi/efivars
/dev/rvn0n1p8  47G  240  20G  58%./7
tmps       8.8G   0  3.8G  0%./dev/Spm
tmps       5.9M   M  6.0M  0%./run/lock
/dev/nyme@n1p8 958M  297M  656M  32%./bovt/efi
tmps       5.8G   0  5.8G  0%./run/genu
tmps       772M  7.9M  764M  2%./run/user/1000

Saved 1 message in /home/vaishnavi/mbox
Held 1 message in /var/hail/vaishnavi
Trionmon@ansclear ~ 11 %

```

Result: The script successfully logs daily system information, archives logs older than 7 days, and schedules itself to run daily using a cron job.

Exercise 3: Enhancements and Exploration

Task Statement: Explore additional functionalities that can be added to the Daily System Logger script to improve automation, usability, and robustness.

1. Send an Email with Log Attachment

You can send the latest log file via email using the mail command. This feature is useful for system administrators who want daily reports in their inbox.

Example Implementation:

```

# Send the latest log file as an email attachment
LOG_FILE="$LOG_DIR/log_$(date +%Y-%m-%d).txt"
mail -s "Daily System Log - $(date)" -a "$LOG_FILE" admin@example.com < /dev/null

# Check if directories exist
if [ ! -d "$LOG_DIR" ]; then
    echo "Error: Log directory not found! Creating it now..."
    mkdir -p "$LOG_DIR"
fi

if [ ! -d "$ARCHIVE_DIR" ]; then
    echo "Error: Archive directory not found! Creating it now..."
    mkdir -p "$ARCHIVE_DIR"

```

```
fi

# Verify log file creation
if [ ! -f "$LOG_FILE" ]; then
    echo "Error: Failed to create log file." >&2
    exit 1
fi

#!/bin/bash
echo "===== Daily Log Menu ====="
echo "1. View latest log"
echo "2. Archive old logs"
echo "3. Clean up old logs"
echo "4. Exit"
echo "====="
read -p "Enter your choice [1-4]: " choice

case $choice in
    1)
        echo "Showing latest log:"
        cat "$LOG_DIR/log_$(date +%Y-%m-%d).txt"
        ;;
    2)
        echo "Archiving old logs..."
        tar -czf "$ARCHIVE_DIR/manual_archive_$(date +%Y-%m-%d).tar.gz" "$LOG_DIR"/log_*.txt
        echo "Archive created successfully."
        ;;
    3)
        echo "Deleting logs older than 7 days..."
        find "$LOG_DIR" -name "log_*.txt" -mtime +7 -exec rm {} \;
        echo "Old logs removed."
        ;;
    4)
        echo "Exiting..."
        exit 0
        ;;
    *)
        echo "Invalid option. Please try again."
        ;;
esac
```