Experiment: [Daily System Logger Script]

Name: vinit kumar, Roll No.: 590029353 , Date: 2025-10-18

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:

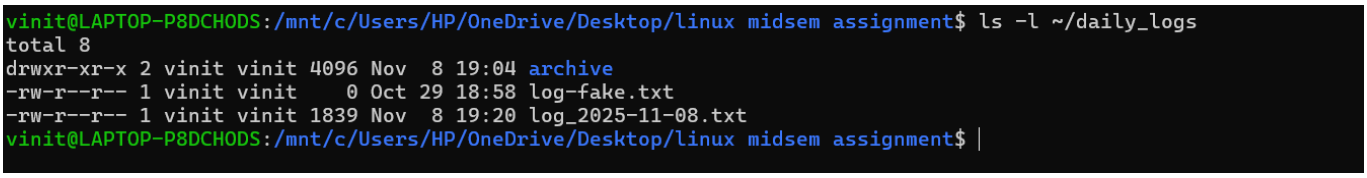
This script:

- 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 - $(date +%Y-%m-%d)" vinitkumar_vk@outlook.com < "$LOG_FILE"
    echo "Log emailed to local mailbox: vinitkumar_vk@outlook.com"
else
    echo "No log file found for today!"
fi
```

Output:



```
vinit@LAPTOP-P8DCHODS:/mnt/c/Users/HP/OneDrive/Desktop/linux midsem assignment$ ls -l ~/daily_logs
total 8
drwxr-xr-x 2 vinit vinit 4096 Nov  8 19:04 archive
-rw-r--r-- 1 vinit vinit    0 Oct 29 18:58 log-fake.txt
-rw-r--r-- 1 vinit vinit 1839 Nov  8 19:20 log_2025-11-08.txt
vinit@LAPTOP-P8DCHODS:/mnt/c/Users/HP/OneDrive/Desktop/linux midsem assignment$
```

Exercise 2: Scheduling the Script

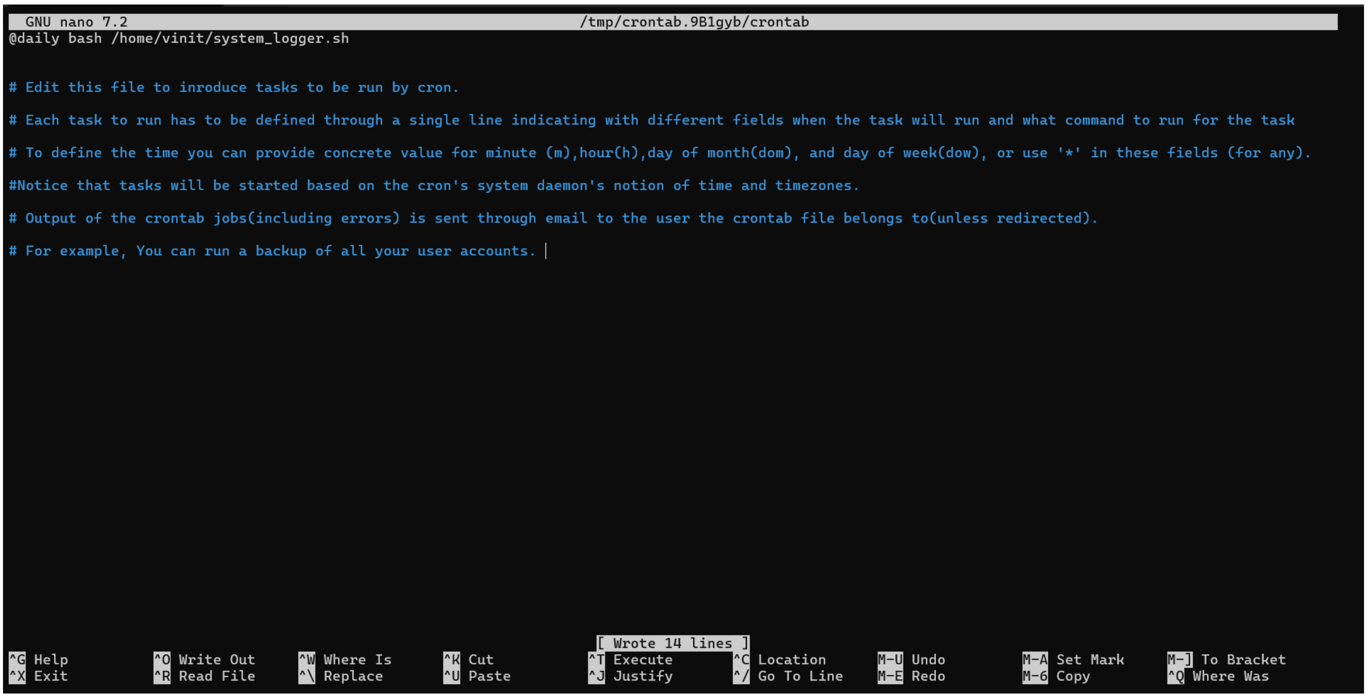
Task Statement: Schedule the above script to run daily using cron.

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

Command(s):

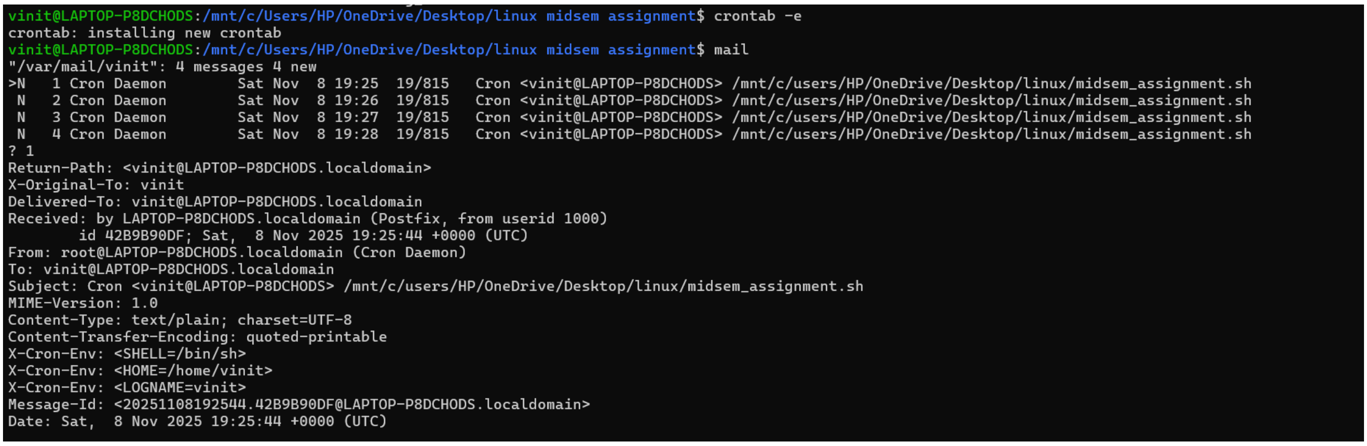
```
crontab -e
0 8 * * * /c/users/HP/onedrive/desktop/linux/midsem_assignment.sh
```

Output:



```
  GNU nano 7.2                                  /tmp/crontab.9B1gyb/crontab
@daily bash /home/vinit/system_logger.sh


# Edit this file to inroduce 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 run and what command to run for the task

# To define the time you can provide concrete value for minute (m),hour(h),day of month(dom), 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.

                                               [ Wrote 14 lines ]
```

Final Output:



```
vinit@LAPTOP-P8DCHODS:/mnt/c/Users/HP/OneDrive/Desktop/linux midsem assignment$ crontab -e
crontab: installing new crontab
vinit@LAPTOP-P8DCHODS:/mnt/c/Users/HP/OneDrive/Desktop/linux midsem assignment$ mail
"/var/mail/vinit": 4 messages 4 new
>N   1 Cron Daemon        Sat Nov  8 19:25  19/815   Cron <vinit@LAPTOP-P8DCHODS> /mnt/c/users/HP/OneDrive/Desktop/linux/midsem_assignment.sh
 N   2 Cron Daemon        Sat Nov  8 19:26  19/815   Cron <vinit@LAPTOP-P8DCHODS> /mnt/c/users/HP/OneDrive/Desktop/linux/midsem_assignment.sh
 N   3 Cron Daemon        Sat Nov  8 19:27  19/815   Cron <vinit@LAPTOP-P8DCHODS> /mnt/c/users/HP/OneDrive/Desktop/linux/midsem_assignment.sh
 N   4 Cron Daemon        Sat Nov  8 19:28  19/815   Cron <vinit@LAPTOP-P8DCHODS> /mnt/c/users/HP/OneDrive/Desktop/linux/midsem_assignment.sh
? 1
Return-Path: <vinit@LAPTOP-P8DCHODS.localdomain>
X-Original-To: vinit
Delivered-To: vinit@LAPTOP-P8DCHODS.localdomain
Received: by LAPTOP-P8DCHODS.localdomain (Postfix, from userid 1000)
        id 42B9B90DF; Sat,  8 Nov 2025 19:25:44 +0000 (UTC)
From: root@LAPTOP-P8DCHODS.localdomain (Cron Daemon)
To: vinit@LAPTOP-P8DCHODS.localdomain
Subject: Cron <vinit@LAPTOP-P8DCHODS> /mnt/c/users/HP/OneDrive/Desktop/linux/midsem_assignment.sh
MIME-Version: 1.0
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: quoted-printable
X-Cron-Env: <SHELL=/bin/sh>
X-Cron-Env: <HOME=/home/vinit>
X-Cron-Env: <LOGNAME=vinit>
Message-Id: <20251108192544.42B9B90DF@LAPTOP-P8DCHODS.localdomain>
Date: Sat,  8 Nov 2025 19:25:44 +0000 (UTC)
```

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