Task 6: Crontab Practice – PoC

This PoC demonstrates how to automate tasks in Linux using **cron jobs**.

Objective

Create **three cron jobs** to:

`1` Print **"Good morning!"** every day at 8 AM.

`2` Backup `/home/studentuser/projectX/` every Sunday.

`3` Delete `.log` files older than 7 days every night at midnight.

Steps

1. Create a monitoring script (optional for backup task)

We created a script called `/usr/local/bin/monitor.sh` for any custom tasks (like backups).

```
sudo nano /usr/local/bin/monitor.sh
```

Example content:

```
#!/bin/bash
echo "Monitoring tasks running..."
```

Then make it executable:

```
sudo chmod +x /usr/local/bin/monitor.sh
```

2. Open the crontab editor

```
sudo crontab -e
```

3. Add the following cron jobs

1 Print 'Good morning!' every day at 8 AM
0 8 * * * echo "Good morning!"

2 Backup /home/studentuser/projectX every Sunday at 2 AM
0 2 * * 0 tar -czf /home/studentuser/projectX_backup_$(date +\%F).tar.gz /home/studentuser/projectX/

3 Delete .log files older than 7 days every night at midnight
0 0 * * * find /home/studentuser/projectX/logs -name "*.log" -type f -mtime + 7 -delete

```
  ┌──(kali㉿kali)-[~]
  └─$ su - studentuser

Password:
  ┌──(studentuser㉿kali)-[~]
  └─$ crontab -e
no crontab for studentuser - using an empty one
Select an editor.  To change later, run select-editor again.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny

Choose 1-3 [1]: 1
crontab: installing new crontab

  ┌──(studentuser㉿kali)-[~]
  └─$ crontab -l
# 1. Print "Good morning!" every day at 8 AM
0 8 * * * echo "Good morning!" >> /home/studentuser/morning_greeting.txt

# 2. Backup '/home/studentuser/projectX/' every Sunday at 1:00 AM
0 1 * * 0 tar -czf /home/studentuser/projectX_backup_$(date +\%F).tar.gz /home/studentuser/projectX/

# 3. Delete '.log' files older than 7 days every Friday at midnight
0 0 * * 5 find /home/studentuser/projectX/ -type f -name "*.log" -mtime +7 -exec rm {} \;
# 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.
```