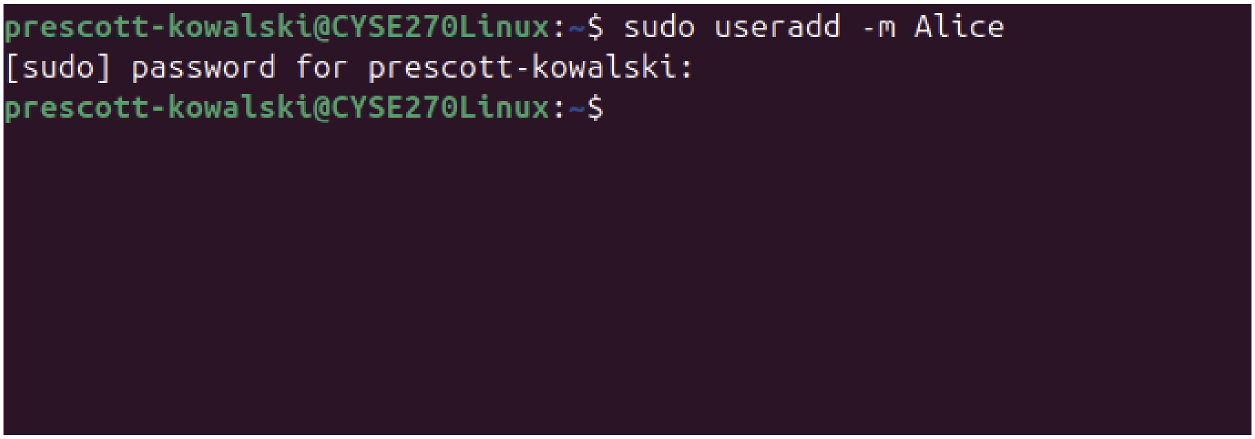
Assignment-9

Task A - Backup your system (Using crontab) [100 points]

Scenario: Performing system backup can be time-consuming, and the process is often overlooked. For this scenario:

1. (10 Points) Create a new user Alice (with home directory).

sudo useradd -m Alice



```
prescott-kowalski@CYSE270Linux:~$ sudo useradd -m Alice
[sudo] password for prescott-kowalski:
prescott-kowalski@CYSE270Linux:~$
```

2. (50 Points) Write a shell script that backups Alice’s home directory by creating a tar file (tape archive), using the following steps:

a. Do the following:

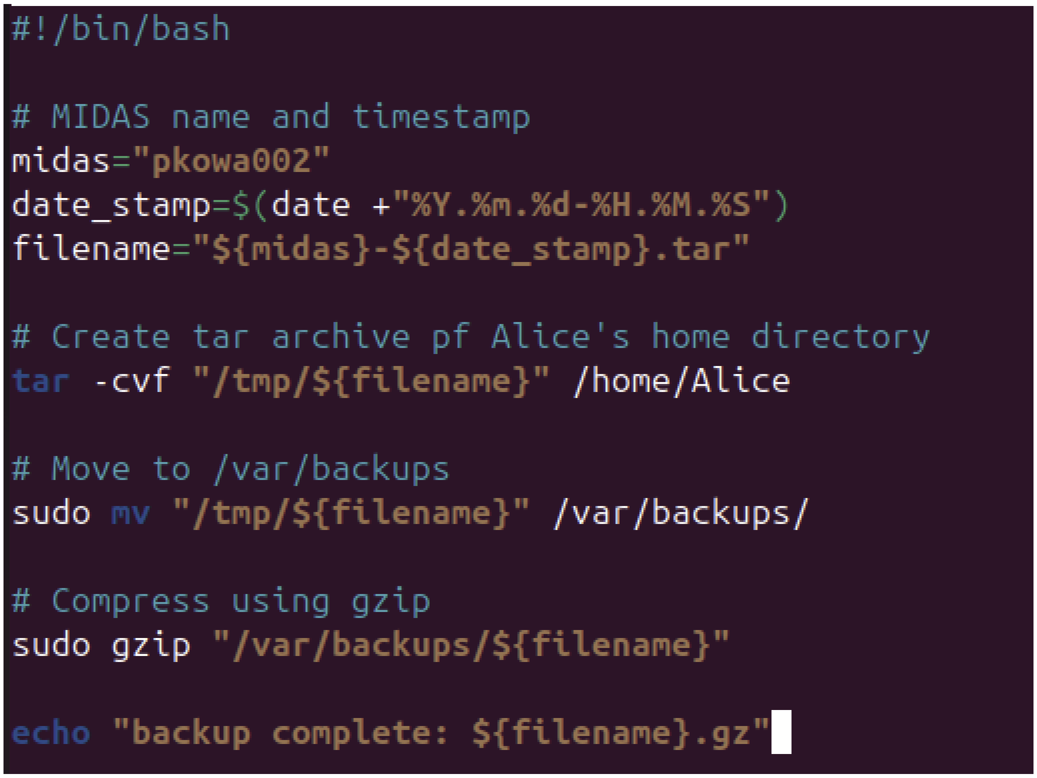
• Take 2 inputs with their values- your MIDAS name and current date (for example, midas=Mohammed).

• Create a variable named as filename that should be assigned the value as MIDAS-date (example output after executing the script would be like, Mohammed-2024.11.04-22.08.01.tar.gz).

• Using tar command, create a tape archive for Alice’s home directory (/home/Alice) and the filename created above (in step-2-ii). (Please learn about tar command in Linux for its usage)

b. Move the tape archive file/tar file (created in step 2-iii) to /var/backups/ directory using correct command in linux.

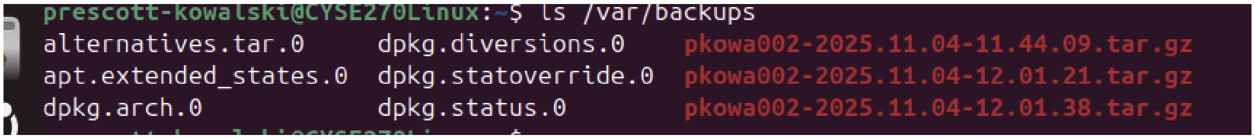
c. To optimize the disk usage, pick a compression algorithm (bz2, gzip, or xv) to compress the tar file you created in /var/backups/ in the previous step-2b.



```
#!/bin/bash

# MIDAS name and timestamp
midas="pkowa002"
date_stamp=$(date +"%Y.%m.%d-%H.%M.%S")
filename="${midas}-${date_stamp}.tar"

# Create tar archive pf Alice's home directory
tar -cvf "/tmp/${filename}" /home/Alice

# Move to /var/backups
sudo mv "/tmp/${filename}" /var/backups/

# Compress using gzip
sudo gzip "/var/backups/${filename}"

echo "backup complete: ${filename}.gz"
```

3. (30 Points) Create a crontab file to keep the scheduled task running for 3 minutes, then check the contents in the /var/backups directory.

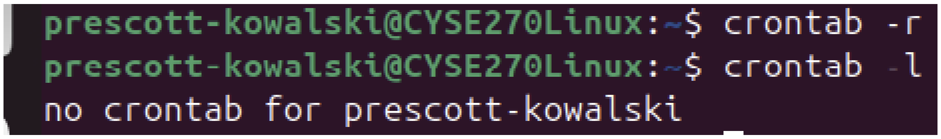
crontab -e



```
prescott-kowalski@CYSE270Linux:~$ ls /var/backups
alternatives.tar.0      dpkg.diversions.0     pkowa002-2025.11.04-11.44.09.tar.gz
apt.extended_states.0   dpkg.statoverride.0   pkowa002-2025.11.04-12.01.21.tar.gz
dpkg.arch.0             dpkg.status.0         pkowa002-2025.11.04-12.01.38.tar.gz
```

4. (10 Points) Cancel the crontab jobs.

Crontab -r



```
prescott-kowalski@CYSE270Linux:~$ crontab -r
prescott-kowalski@CYSE270Linux:~$ crontab -l
no crontab for prescott-kowalski
```

TASK B: SYSTEM CLEANUP (EXTRA CREDIT) [20 Points]

Scenario: In the above scenario, your system disk will be filled up eventually without

cleaning up the old backups. Therefore, in this optional task, create a script that checks the

number of backups you created in Task A. If the number of the backup file is more than a
pre-defined threshold, the script will delete the old archives to maintain the backups under
a reasonable size.
This script should do the following:
1. Count the number of backups created in Task A and determine if this number is larger
than 3.
2. Nothing should happen if the number of backups is less than the threshold, 3.
3. If more backup archives are detected, calculate the number of backups to delete. Then
delete the old archives.
Note: As the script needs to write contents in the “/var/backups” folder, which is owned by
root, you should consider the permission issue properly. (Using sudo to create crontab file)
Reference: How to Format Date for Display or Use In a Shell Script:
https://www.cyberciti.biz/faq/linux-unix-formatting-dates-for-display/
Reference: How to append date timestamp to filename:
https://crunchify.com/shell-script-append-timestamp-to-file-name/