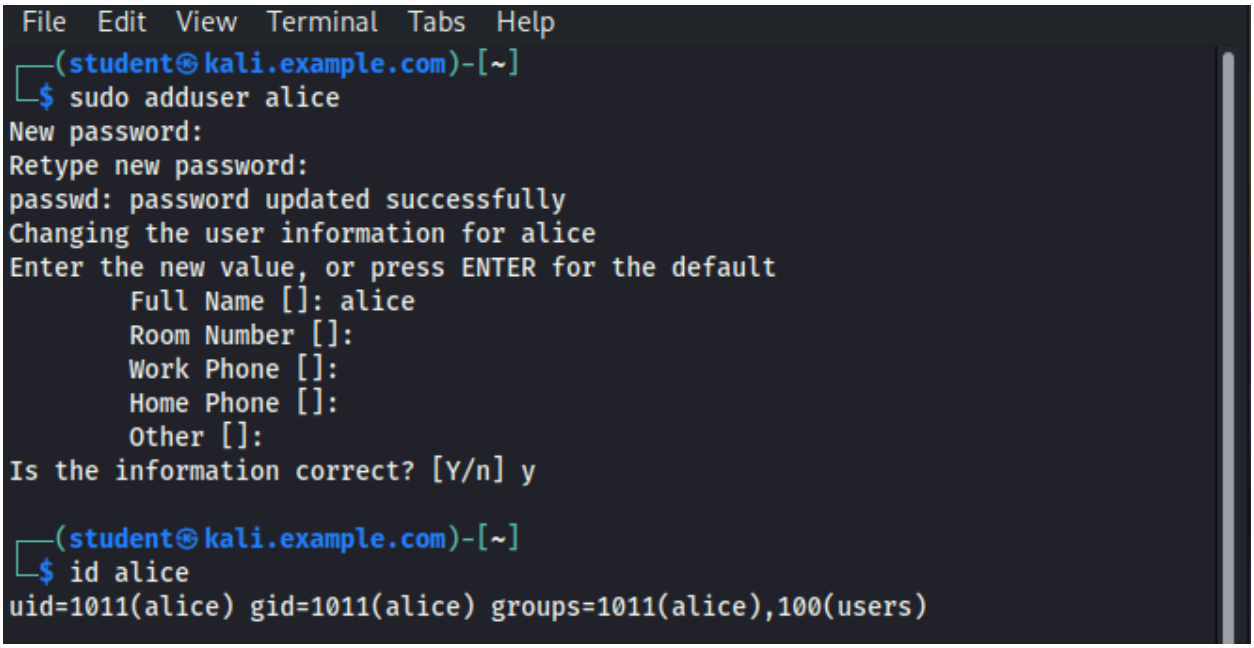
Bryce Baxter

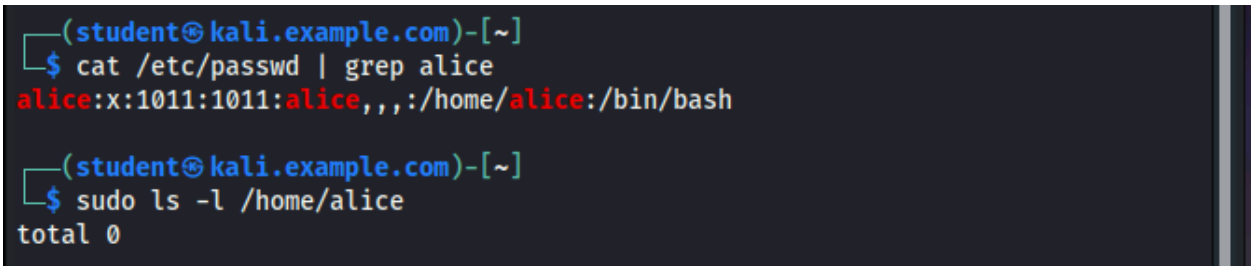
11/04/25

CYSE 270

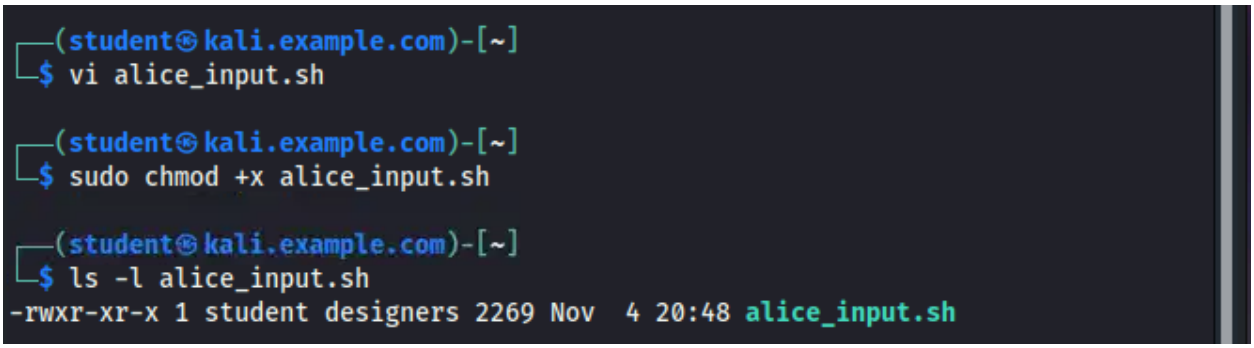
Assignment 9 – Task Automation

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




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



```
File  Edit  View  Terminal  Tabs  Help
#!/bin/bash
# alice_input.sh
#
#Instructions:
# (1) Write a shell script
#
```

a. Do the following:

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

```
# -- Step 2a i:Take two inputs with their values: your Midas name and the curren
t date.
~ MIDAS_NAME=bbaxt003
 echo "Current midas name is $bbaxt003."

#Using | tr -d '\n' sed to remove unwanted potential characters from the date ou
tput.
 CURRENT_DATE=$(date +%Y.%m.%d-%H.%M.%S) #| tr -d '\n'
 echo "Current date and time is $November 4th, 3:06pm."
```

- 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).

```
# -- 2a ii: Create a variable name as 'filename' that should be assigned to the
value as 'MIDAS_date.'
 filename="${MIDAS_NAME}-${CURRENT_DATE}.tar"
# COMPRESSED_FILENAME will be used for the final output display, after gzip adds
.gz
 COMPRESSED_FILENAME="${filename}.gz"
 echo ""

 echo "Preparing to backup /home/alice..."
 echo "Archive will be named: ${filename}."
 echo "The compressed file version is ${COMPRESSED_FILENAME}."
 echo ""
```

- 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)

```
# -- 2a iii: Using tar command, create a tape archive for Alice's home directory
and the file name created above.
 echo "Creating tar archive of /home/alice..."
 tar -cvf "${filename}" /home/alice

# -- 2a iii: Check if your tar command was successful.
 if [ $? -ne 0 ]; then
         echo "Error: Failed to crate a tar archive/file. Exiting Script."
         exit 1
 fi #This 'fi' closes the if statement for the tar command.

 echo "Tar archive '${filename}' created successfully."
 echo ""
```

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

```
# -- 2b: Move the tape archive file to /var/backups/directory.
 BACKUP_DIR="/var/backups"

 echo "Moving '${filename}' to '${BACKUP_DIR}' ..."
 mv "${filename}" "${BACKUP_DIR}/${filename}"

# -- 2b: Check if the mv command was successful.
 if [ $? -ne 0 ]; then
         echo "Error: Failed to move archive to ${BACKUP_DIR}. Exiting script."
         exit 1
 fi #This 'fi' closes the if statement for the mv command.

 echo "Archive moved to '${BACKUP_DIR}'."
 echo ""z
```

- 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

```
# -- Step 2c i: Compress the tar file using the gzip (optimizing disk usage).
 echo "Compressing '${BACKUP_DIR}/${filename}' using gzip..."
 gzip "${BACKUP_DIR}/${filename}"

 if [ $? -ne 0 ]; then
         echo "Error: Failed to compress archive. Exiting Script."
         exit 1
 fi #This 'fi' closes the if statement for the gzip command.

 echo "Backup completed successfully!"
 echo "The compressed backup file is located at: ${BACKUP_DIR}/${COMPRESSED_FILE
NAME}."
 echo ""
```

Step 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.

```
┌──(root㉿kali.example.com)-[~]
└─# sudo crontab -e
no crontab for root - 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]: 2
crontab: installing new crontab
```

```
┌──(student㉿kali.example.com)-[/root]
└─$ ls /var/backups
alternatives.tar.0  dpkg.diversions.0    dpkg.status.0
dpkg.arch.0         dpkg.statoverride.0
```

Step 4 (10 Points) Cancel the crontab jobs.

```
┌──(student㉿kali.example.com)-[/root]
└─$ sudo crontab -r

┌──(student㉿kali.example.com)-[/root]
└─$ crontab -l
no crontab for student

┌──(student㉿kali.example.com)-[/root]
└─$ sudo crontab -l
no crontab for root
```