OLD DOMINION UNIVERSITY

CYSE 270 LINUX SYSTEM FOR CYBERSECURITY

Assignment #10 Automation Tasks and Shell Scripting

John Wilson

01179411

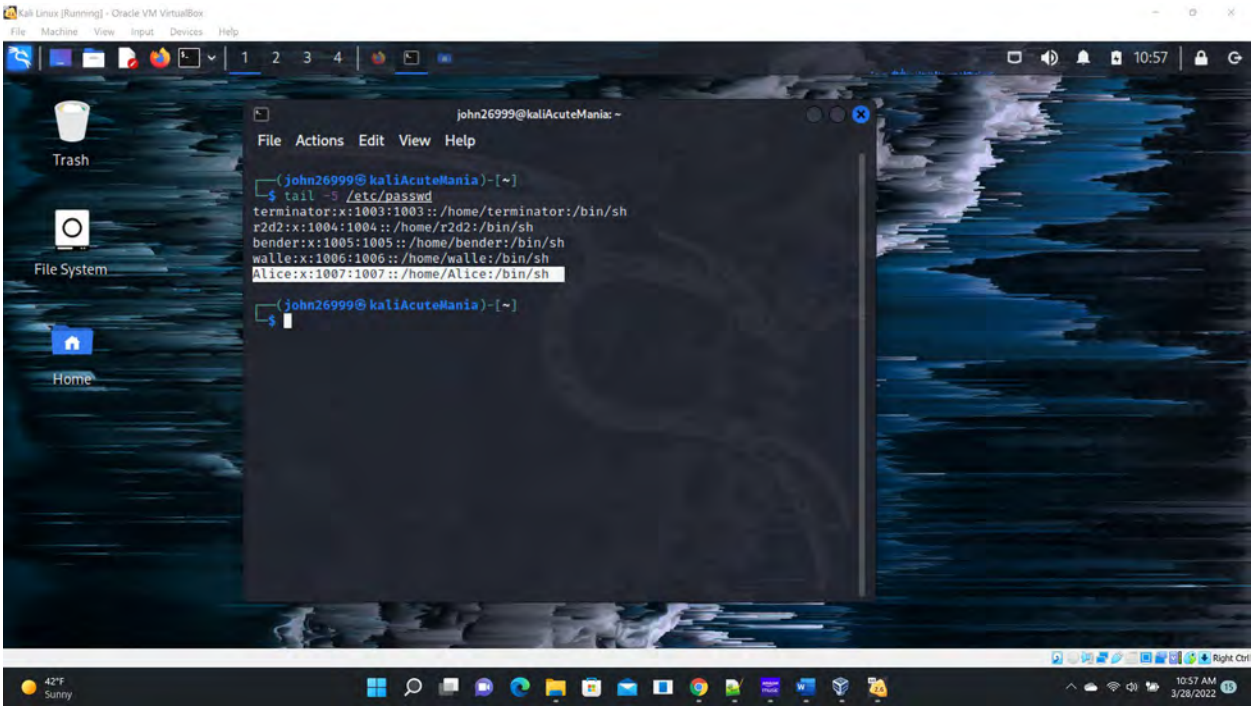
John Wilson
3/28/22

CYSE 270 Assignment #10 Automation Tasks and Shell Scripting

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) Create a new user **Alice (with home directory**) and write a shell script that backups Alice's home directory, using the following steps:
   a) Take your **MIDAS** and **current date** as input
   b) Create a **tar file** in the **/var/backups** directory.
      i) The filename should be, **YourMIDAS-datetime**, where date and time are replaced with the current system time (for example, svatsa2021.3.17- 01.16.430.tar).
      ii) The tar file should include Alice's home directory along with the filename created above (for example, svatsa2021.3.17-01.16.430.tar /home/Alice).
   c) To optimize the disk usage, pick a compression algorithm (bz2, gzip, or xv) to compress the tar file you created.



CYSE 270 Assignment #10 Automation Tasks and Shell Scripting

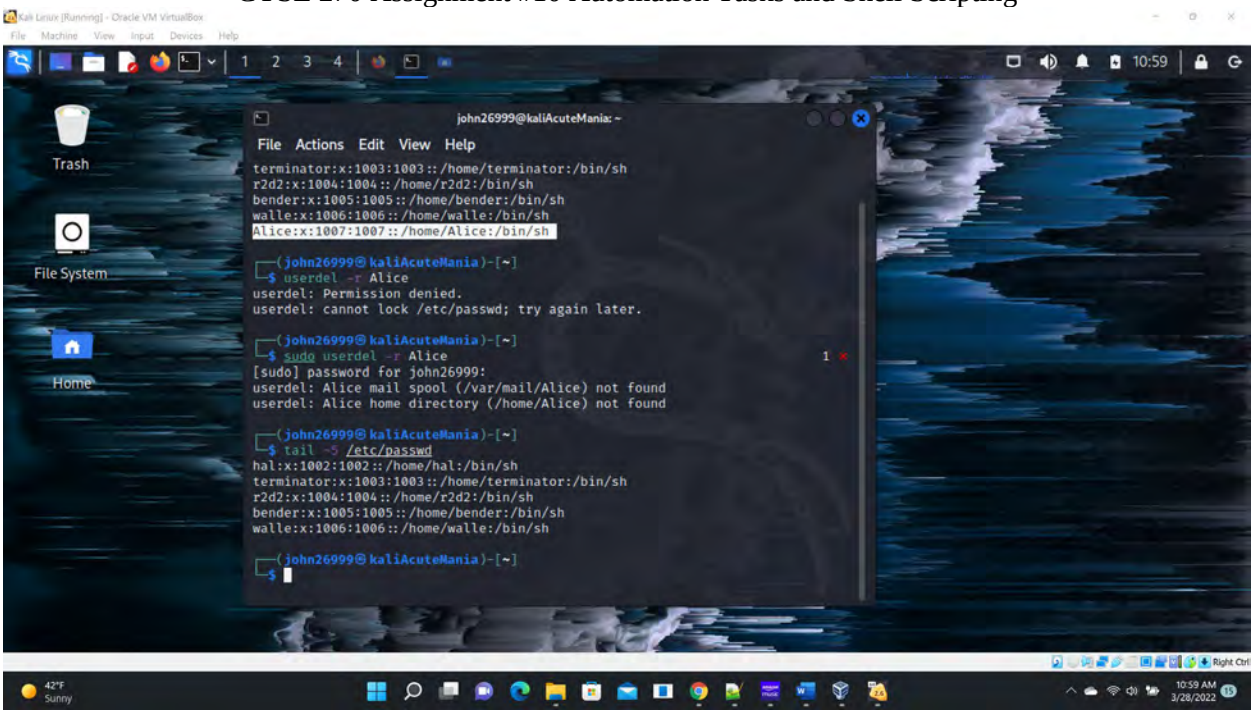


Figure 1 Screenshots of JWILS082 Computer screen prior to Step 1.

Above is the screen shot using the command “tail -5 /etc/passwd” ” to inveistgate if the user Alice existed in the system. Alice does exist.

In the following screenshot I used the command “sudo userdel -r Alice” to delete the user Alice and all of the information pertatiing to the user. Was not sure if I was supposed to do this; however, I decided to do it as a refresher of the procedures and to start the assignment as though Alice did not exist. I also used the command “tail -5 /etc/passwd” ” to show user Alice has ceased to exist in the system.

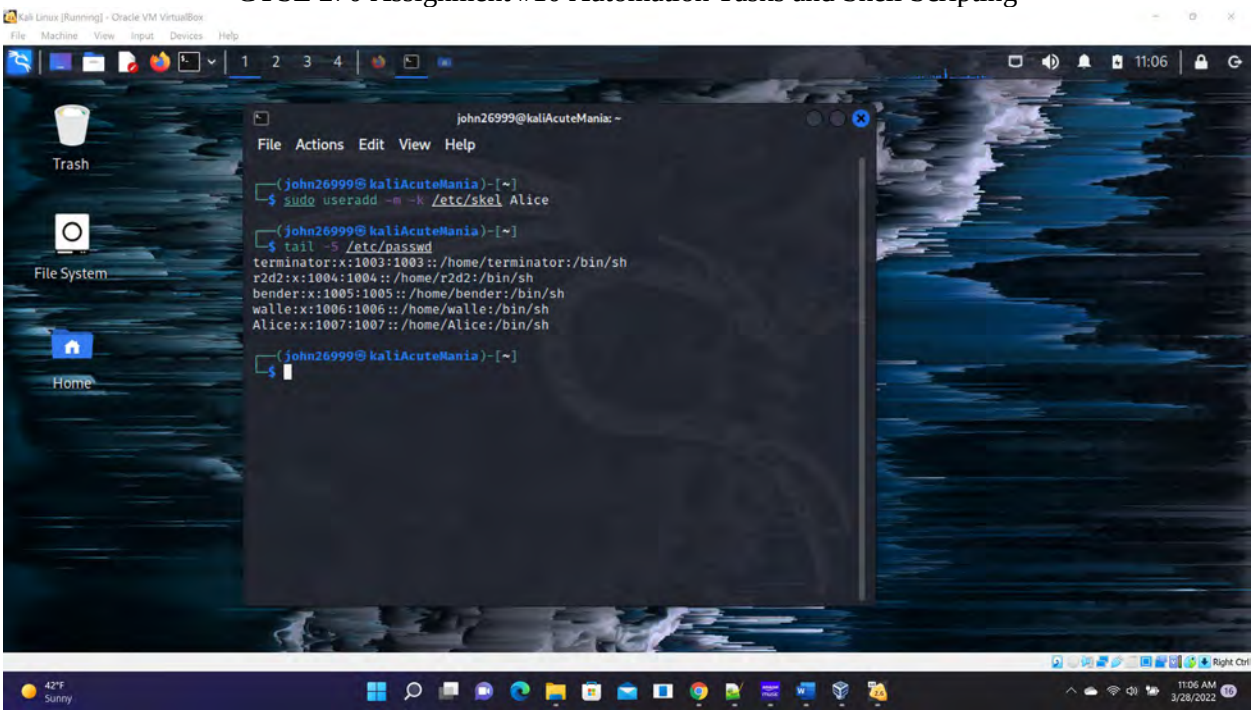
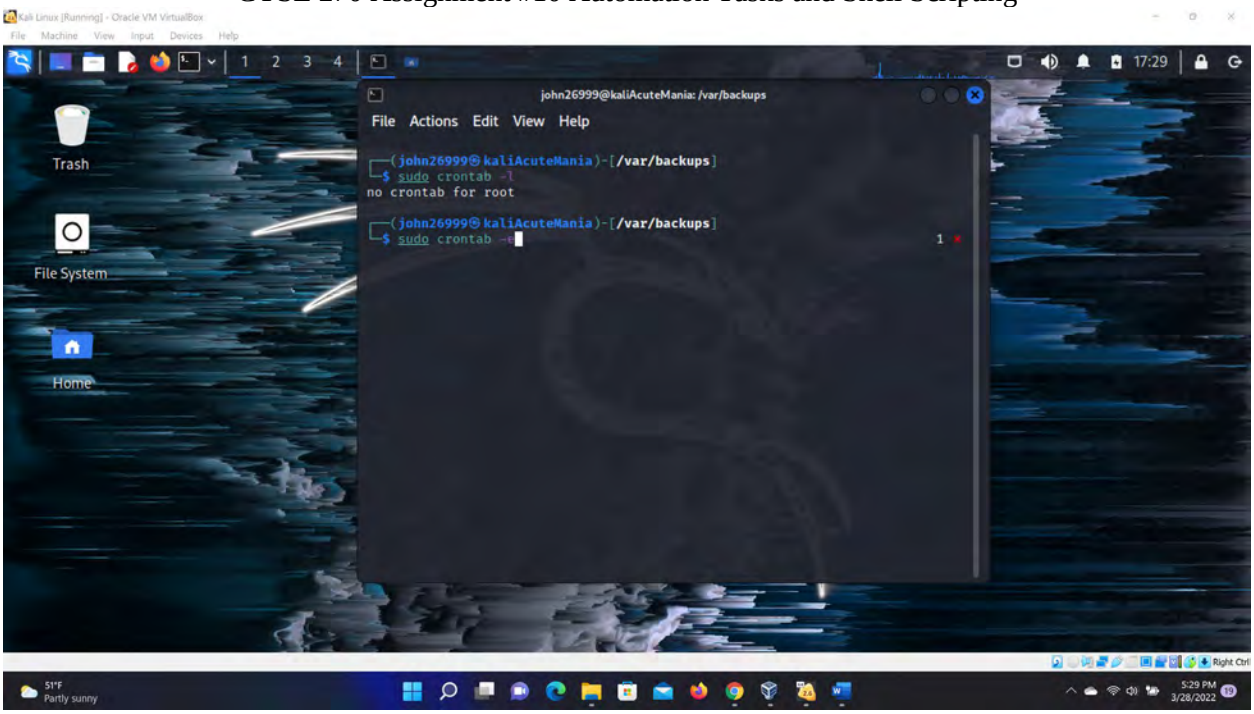
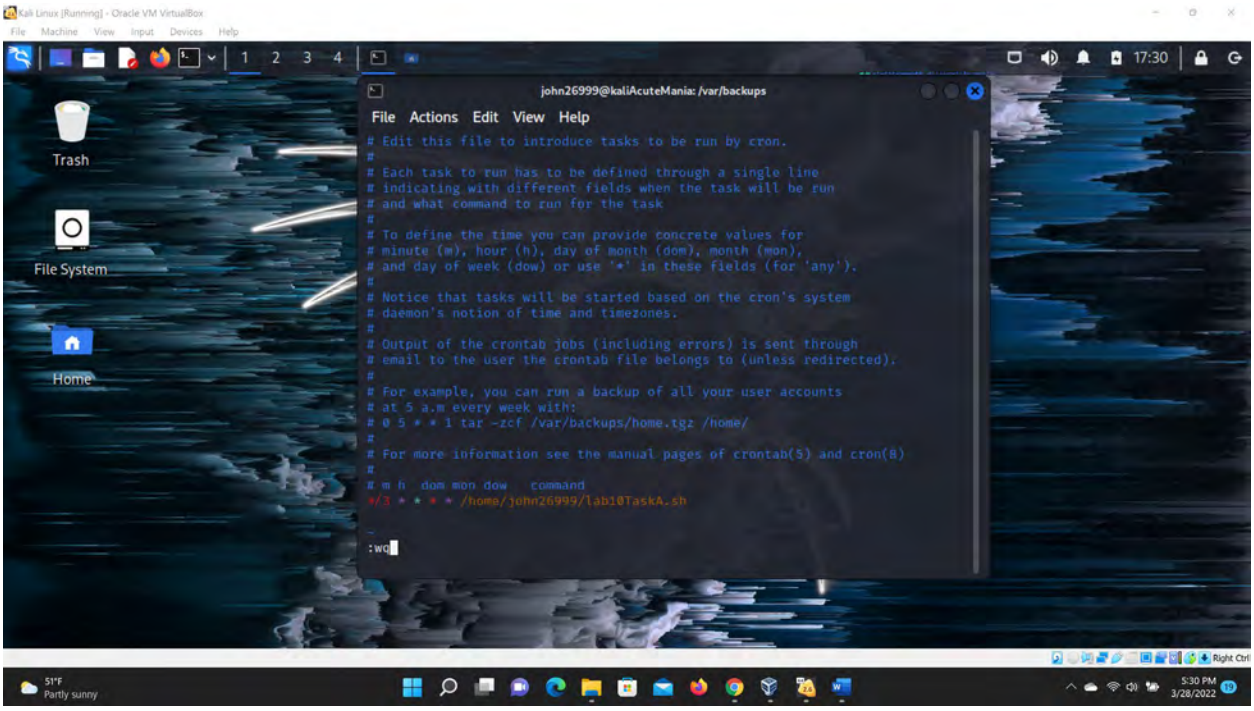


Figure 2 Screenshots of JWILS082 Computer screen for Step 1.

Above is the screen shot using the command “sudo useradd -m -k /etc/skel Alice” to create a new user with a home directory. “sudo” is the command that allows you to run programs with the security privileges of another user (otherwise known as a super user). “useradd” is the command that adds a new user profile to the system. “-m” is the command that creates the new user’s home directory. “-k” is the command the uses this skeleton directory. “/etc/shadow” is where the users home directotry is located. “Alice” is the new user name added. I also used the command “tail -5 /etc/passwd” to prove the new user account was created with a home directory.

CYSE 270 Assignment #10 Automation Tasks and Shell Scripting





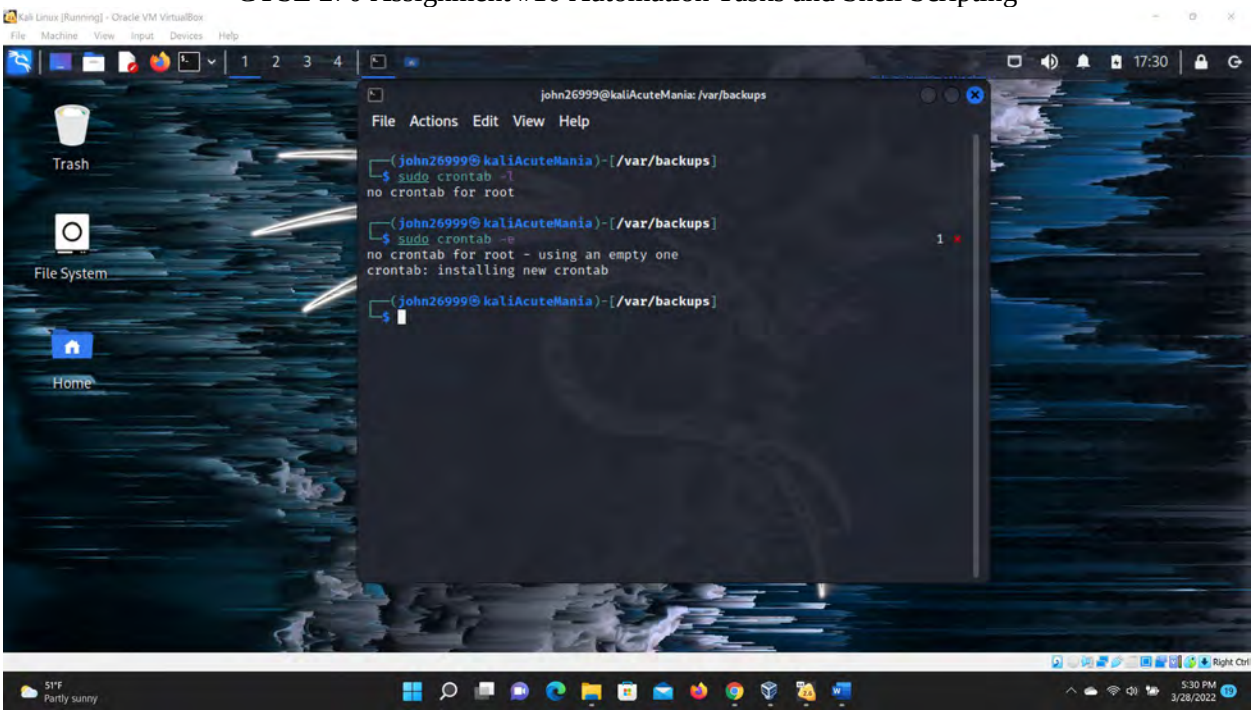


Figure 3 Screenshots of JWILS082 Computer screen for Step 1.

Above is the screen shot of me running crontab. The first shot is running the command “sudo crontab -l” which checks to see if one is already running. Then I run the command “sudo crontab -e” which opens up the VIM editor to put in the run script in crontab.

In the second is a screenshot of the command “*/3 * * * * /home/john26999/lab10TaskA.sh” that will this script every 3mins. Then I save it

In the third screenshot, you can see the crontab has been installed.

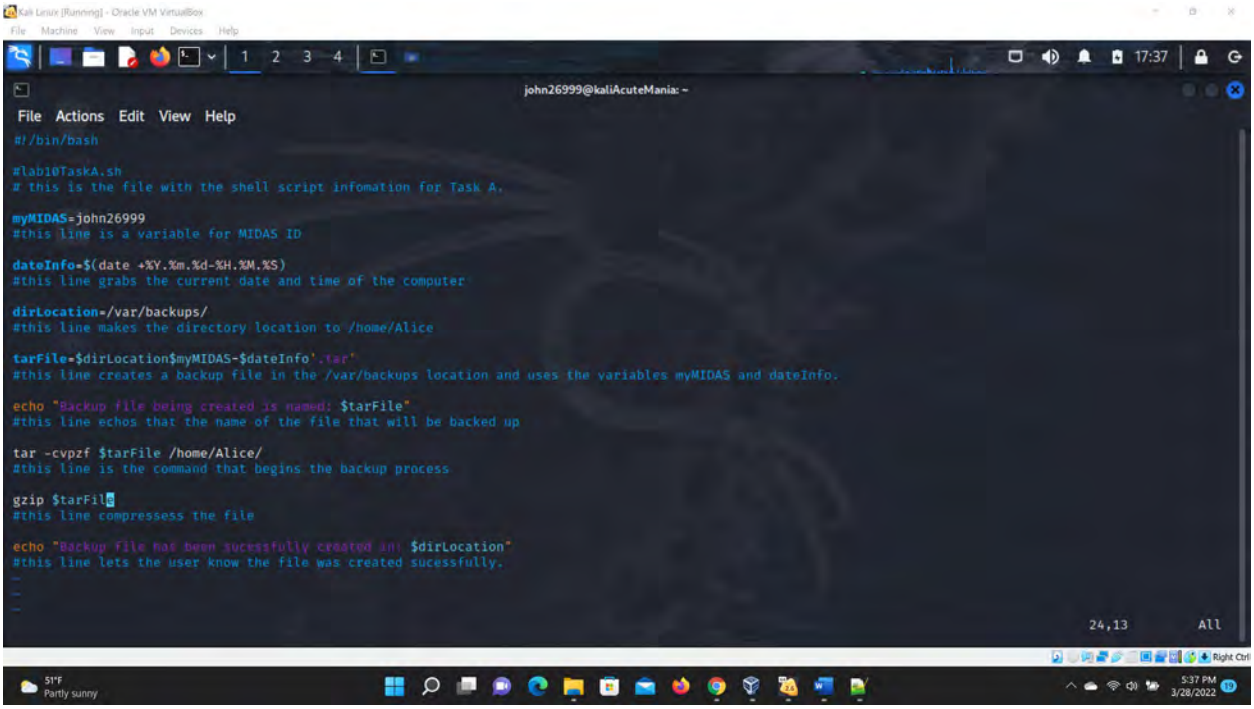


Figure 4 Screenshots of JWILS082 Computer screen for Step 1.

Above is the screen shot of the source code with comments.

Below is the source code for Task A with comments as directed in the case you cannot read it from the screen shots.

```
#!/bin/bash

#lab10TaskA.sh
# this is the file with the shell script infomation for Task A.

myMIDAS=john26999
#this line is a variable for MIDAS ID

dateInfo=$(date +%Y.%m.%d-%H.%M.%S)
#this line grabs the current date and time of the computer

dirLocation=/var/backups/
#this line makes the directory location to /home/Alice

tarFile=$dirLocation$myMIDAS-$dateInfo'.tar'
#this line creates a backup file in the /var/backups location and uses the
variables myMIDAS and dateInfo.

echo "Backup file being created is named: $tarFile"
#this line echos that the name of the file that will be backed up

tar -cvpzf $tarFile /home/Alice/
#this line is the command that begins the backup process
```

```
gzip $tarFile
#this line compressess the file

echo "Backup file has been sucessfully created in: $dirLocation"
#this line lets the user know the file was created sucessfully.
```

2) Keep the scheduled task running for 3 minutes, then **check the contents in the /var/backups** directory.

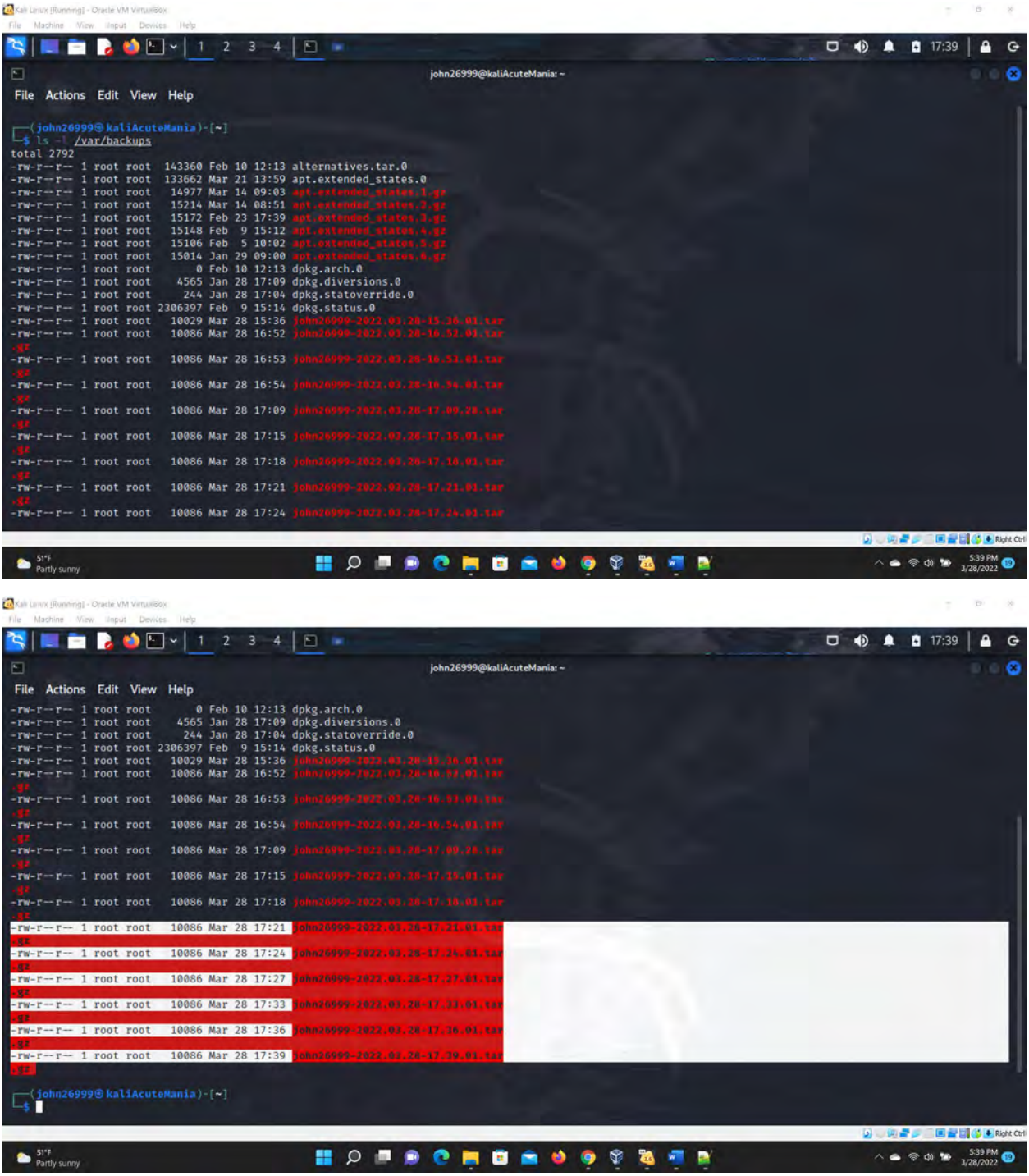


Figure 5 Screenshots of JWILS082 Computer screen for Step 2.

Above is the screen shot of the source code executing the commands every three minutes.

3) **Cancel** the crontab jobs.

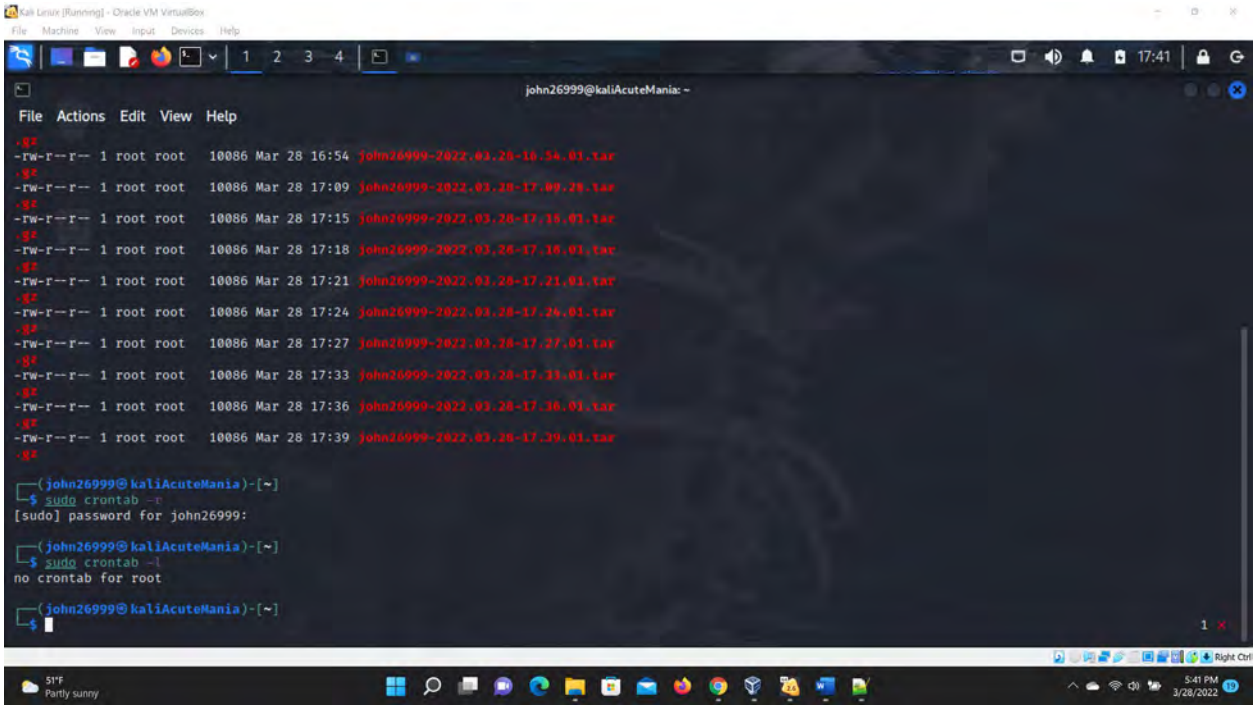


Figure 6 Screenshots of JWILS082 Computer screen for Step 3.

Above is the screen shot of the source code executing the commands every three minutes.

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/

John Wilson
3/28/22

CYSE 270 Assignment #10 Automation Tasks and Shell Scripting

(Extra Credit)- Take your MIDAS/name (10 points)

Write a script like below that reads **MIDAS/name** and displays the message (for example, the screenshot here), if the following requirements are satisfied:

1) Only lower-case letter [a-z] are allowed
2) MIDAS/name must be between 4 to 8 digits

Test your script with the following examples:

1) Your MIDAS /name with one upper case
2) A string less than 4 digits
3) A string longer than 8 digits
4) Your MIDAS/name in lower case, between 4-8 characters in length

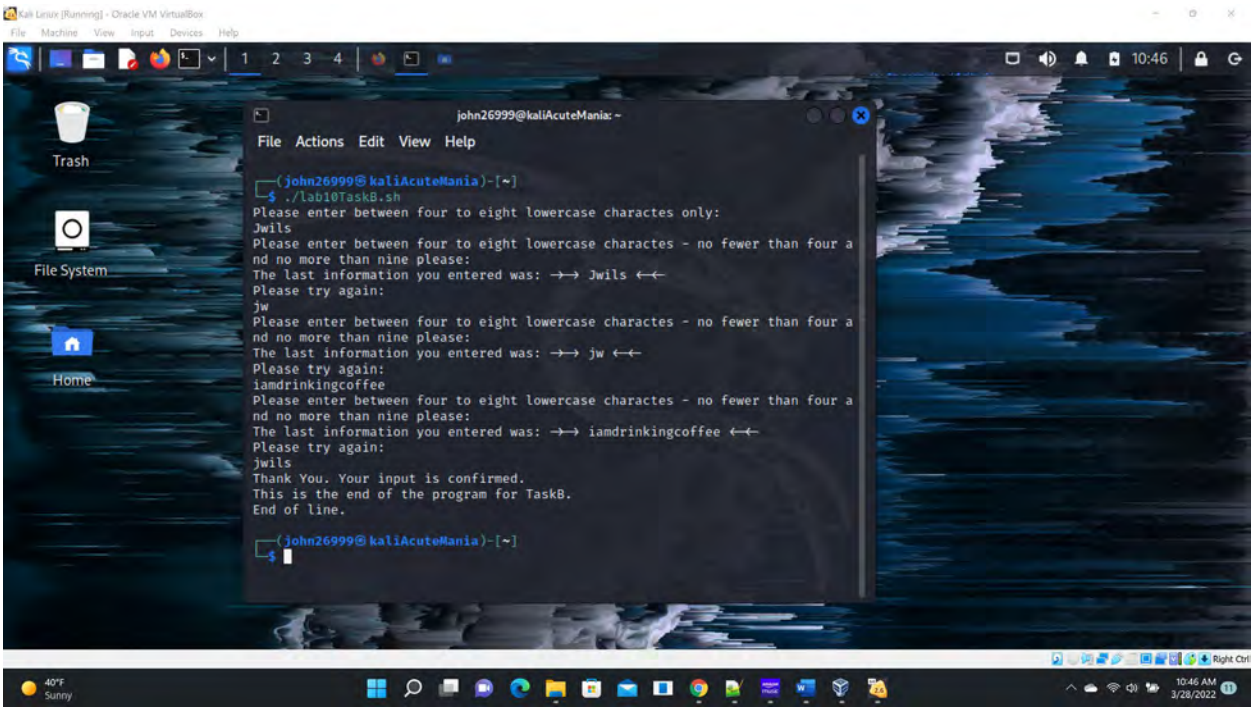
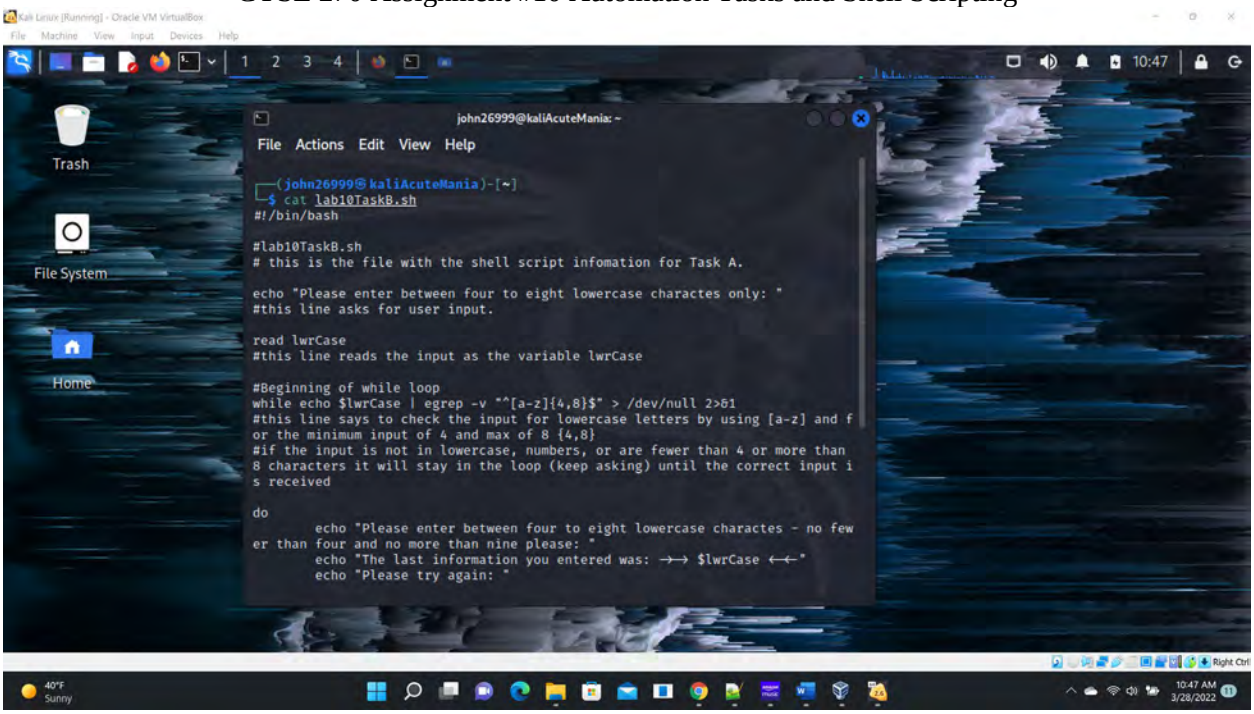


Figure 1 Screenshots of JWILS082 Computer screen for Task B (extra credit).

Above is the screenshots illustrating the scripts functionality as per steps 1 – 5. Below is the actual source code to further verify the results.



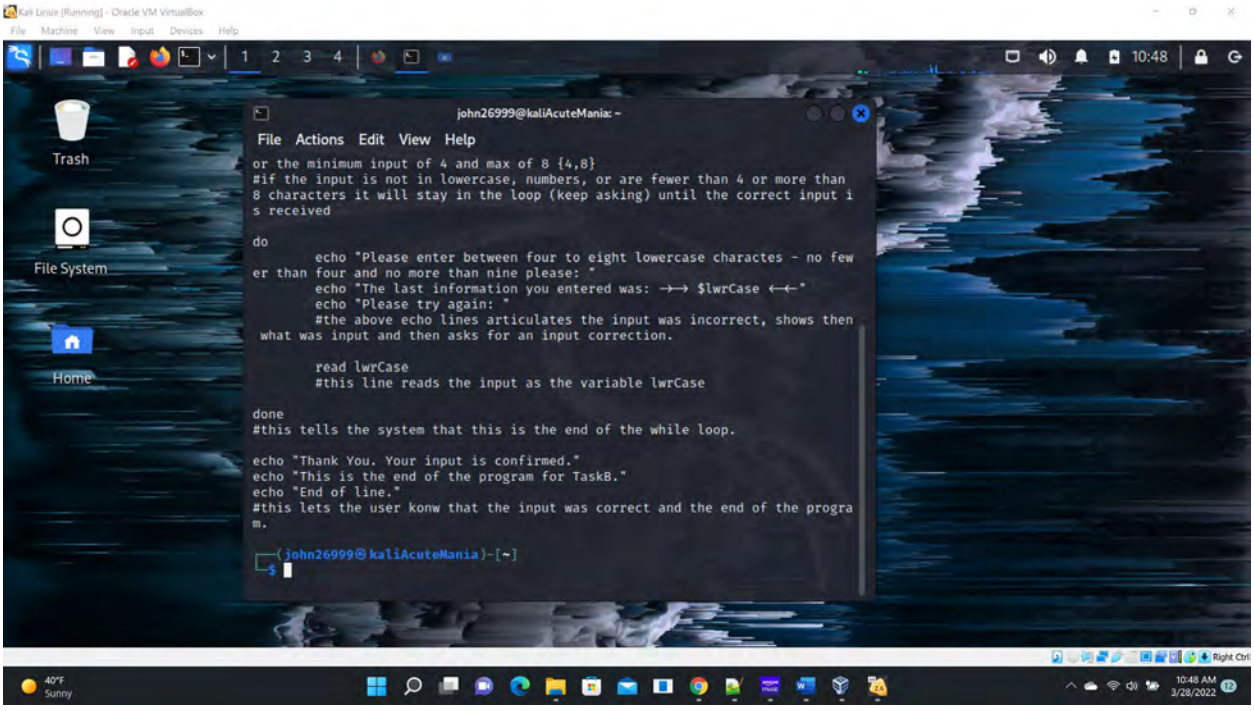


Figure 2 Screenshots of JWILS082 Computer screen for Task B (extra credit).

Above is the screenshots of the script code executing the extracredit assignment.

Below is the source code for TaskB (extra credit) with comments as directed in the case you cannot read it from the screen shots.

CYSE 270 Assignment #10 Automation Tasks and Shell Scripting

```
#!/bin/bash

#lab10TaskB.sh
# this is the file with the shell script infomation for Task A.

echo "Please enter between four to eight lowercase characters only: "
#this line asks for user input.

read lwrCase
#this line reads the input as the variable lwrCase

#Beginning of while loop
while echo $lwrCase | egrep -v "^[a-z]{4,8}$" >dev/null 2>&1
#this line says to check the input for lowercase letters by using [a-z] and
for the minimum input of 4 and max of 8 {4,8}
#if the input is not in lowercase, numbers, or are fewer than 4 or more than
8 characters it will not exit the loop (keep asking) until the correct input
is received

do
    echo "Please enter between four to eight lowercase characters - no fewer
than four and no more than nine please: "
    echo "The last information you entered was $lwrCase"
    echo "Please try again: "
    #the above echo lines articulates the input was incorrect, shows then
what was input and then asks for an input correction.

    read lwrCase
    #this line reads the input as the variable lwrCase

done
#this tells the system that this is the end of the while loop.

echo "Thank You. Your input is confirmed."
echo "This is the end of the program for TaskB."
echo "End of line."
#this lets the user konw that the input was correct and the end of the
program.
```