CYSE 270: Linux System for Cybersecurity

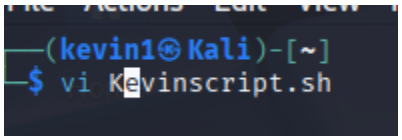
Lab 8 – Shell Scripting

(Total 100 Points)

Please refer to the slides for **week 8 - Shell scripting** and write shell scripts to complete the following tasks. **Submit the screenshot for the script and its output, both.**

NOTE: Please replace the name of the script with the name you used for the script. In the sample screenshot, I have used those names to create my script.

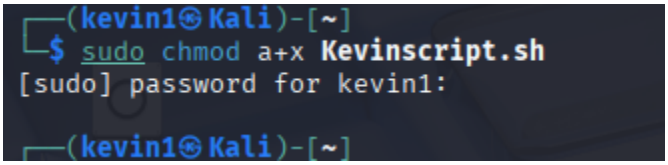
Step-1: Use vi or nano editor to write your script (Ex, **vi YourScriptName.sh**) for the following tasks.



The command uses the vi editor to write my script.

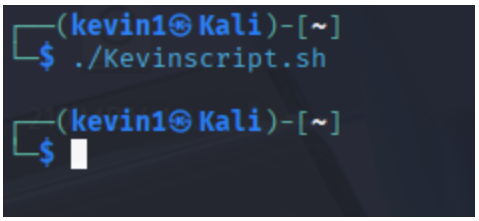
Step-2: After saving the script, **save and exit out of the editor** and make the script executable by adding execute permission (**chmod +x YourScriptName.sh**)

The command saves my script.



As root (using sudo), the "chmod a+x" would change the permissions of the file to allow my script to execute.

Step-3: Run your script using **./YourScriptName.sh**

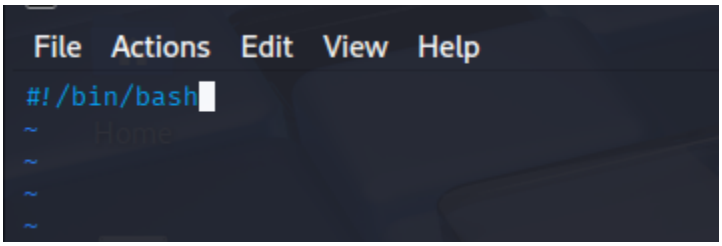


The "./" and the script name would run since it has execution permissions from the last steps.

Task A (Correct script (25 points) + result/output after executing the script (25 points)- Conditional Statement

Write a shell script using nano or vi editor (eg, vi scriptname.sh) like below, that performs the following task:

1. Add the **Shebang** (#!/bin/bash) as the first line in your script.



This would tell the system to execute the script with bash.

2. **Read** a number using **read** function

```
#!/bin/bash
read -p "Enter the number:" number
```

Using the read function to prompt user information, the "-p" displays the purple text, and the "number" is the variable that will store the user's input.

3. Using **if statement**, check if the input number is greater than 10, then print the message "**Input number is greater than 10**".

```
if [ "$number" -gt 10 ]; then
        echo "Input number is greater than 10"
```

Using the if command, the "$number" uses the user's input to check and see if it is greater than 10 (the command for this is "-gt"). If it is greater than 10, then the bottom message will say so.

4. If the number is not greater than 10, then print the message, "**Input number is not greater than 10"**.

```
else
        echo "Input number is not greater than 10"
fi
```

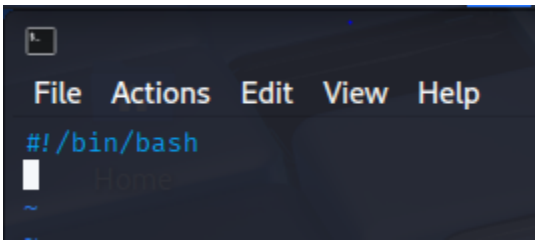
With the else command, if the number is not greater than 10, that message will display instead, and the fi ends the if command.

```
┌──(kevin1㉿Kali)-[~]
└─$ vi TaskA.sh

┌──(kevin1㉿Kali)-[~]
└─$ ./TaskA.sh
Enter the number:4
Input number is not greater than 10

┌──(kevin1㉿Kali)-[~]
└─$ ./TaskA.sh
Enter the number:20
Input number is greater than 10

┌──(kevin1㉿Kali)-[~]
└─$
```

Task B (Correct script (25 points) + result/output after executing the script (25 points) - Shell Script to Create a new file

Write a shell script using nano or vi editor (eg, nano scriptname.sh) like below, that performs the following task:

1. Add the **Shebang** (#!/bin/bash) as the first line in your script.



2. **Reads** the **name** of the file to check for a filename that exists.

```
read -p "Enter the name of file or directory:" filename
```

With the read command, the "-p" displays the text, and the "filename" is used as user input.

3. Check whether the given input is a directory or regular file.

```
if [ -d "$filename" ]; then
        echo "Directory exists"
elif [ -f "$filename" ]; then
        echo "It is a regular file, and the file exists"
        echo "The contents of the file are"
        cat "$filename"
```

Using the "If and "elif", the following "-d" checks if the filename is a directory. If it is, then the text displays it. With "f", it checks if the filename is a file which will display the text and show the contents of the file (using "cat").

4. If the input is a directory and exists, then display the message "**Directory exists**".

```
┌──(kevin1㉿Kali)-[~]
└─$ ./TaskB.sh
Enter the name of file or directory:Documents
Directory exists

┌──(kevin1㉿Kali)-[~]
└─$
```

5. If the input is a regular file, then display the message "**It is a regular file, and the file exists**" and display the contents of the file.

```
┌──(kevin1㉿Kali)-[~]
└─$ ./TaskB.sh
Enter the name of file or directory:EC.txt
It is a regular file, and the file exists
The contents of the file are
5f4dcc3b5aa765d61d8327deb882cf99
63a9f0ea7bb98050796b649e85481845
```

6. **If the given input name in step-1 doesn't exist**, then create the new file with the given name in step-1.

```
┌──(kevin1㉿Kali)-[~]
└─$ ./TaskB.sh
Enter the name of file or directory:NoEC.txt
Does not exist, create file now...
Here is the file.

┌──(kevin1㉿Kali)-[~]
└─$ ls
copyright_cyse270  data  Desktop  Documents  Downloads  EC.txt  kevin1  Kevinscript.sh  kloar001  kloar001.hash  Music  NoEC.txt
```

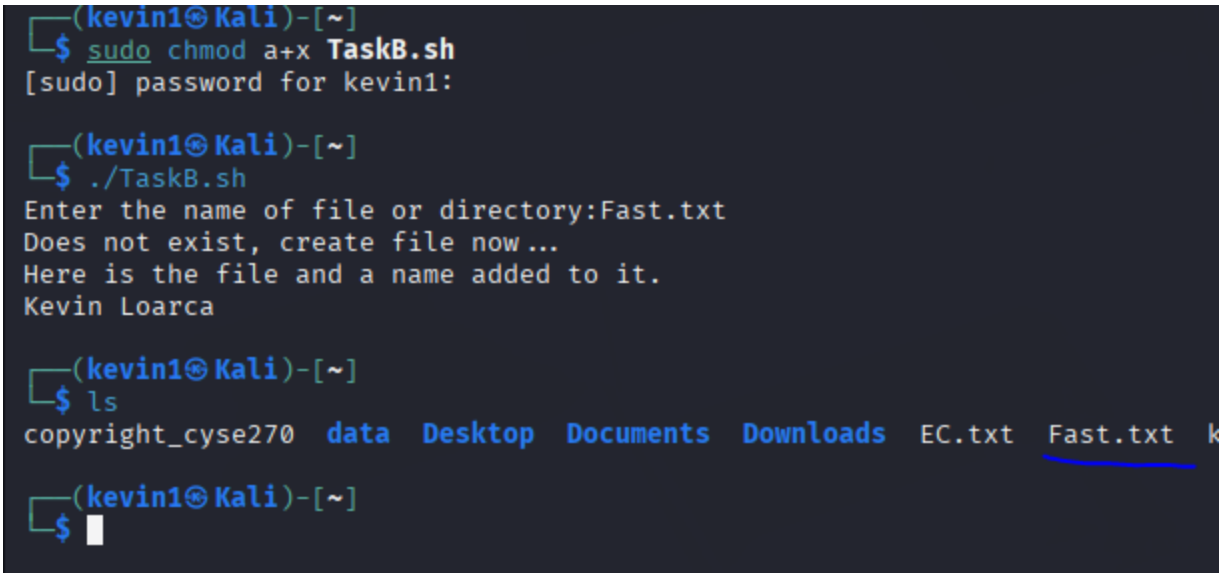
```
File  Actions  Edit  View  Help
#!/bin/bash
read -p "Enter the name of file or directory:" filename

if [ -d "$filename" ]; then
        echo "Directory exists"
elif [ -f "$filename" ]; then
        echo "It is a regular file, and the file exists"
        echo "The contents of the file are"
        cat "$filename"
else
        echo "Does not exist, create file now... "
        touch "$filename"
        echo "Here is the file."
        cat "$filename"
fi
~
~
~
```

(Extra credit: 10 points) Add your name to the file (using redirection operator '>') and display the contents for the newly created file.

7. Save and exit the editor and remember to make the script executable using the command **chmod +x scriptname.sh**)

```
touch "$filename"
echo "Kevin Loarca" > "$filename"
echo "Here is the file and a name added to it."
cat "$filename"
```

With the ">" operator, my name is added to the file.



(Your script should result into the output similar to this sample screenshot after executing as shown below)

Extra Credit (15 points)- Check Directory

Write a script like below that

1. Reads Two variables- your name and the name of the directory as input.

```
read -p "Enter your name:" name
read -p "Enter the name of directory:" directory
```

2. Your script should check for the validity of the given directory name, if the entered filename is a directory, then display its contents

```
if [ -d "$directory" ]; then
        echo "Hello $name, the contents of the directory '$directory' are:"
        ls "$directory"
else
```

3. If the directory doesn't exist, then print an error message "Sorry, the entered directory name is not a valid directory name."

```
else
        echo "Sorry, the entered directory '$directory' name is not a valid directory name."
fi
```

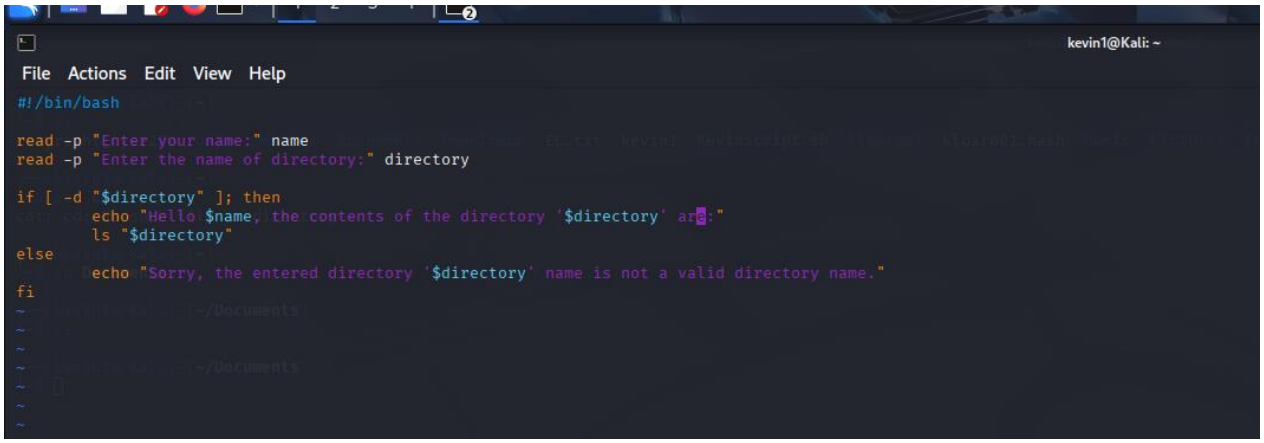
4. You need to execute your script and test the following directories to test with your script

- /etc/systemd
- /home
- A directory that does not exist

```
┌──(kevin1㉿Kali)-[~]
└─$ ./TaskEC.sh
Enter your name:Kevin
Enter the name of directory:/etc/systemd
Hello Kevin, the contents of the directory '/etc/systemd' is:
journald.conf  logind.conf  network  networkd.conf  pstore.conf
```

```
┌──(kevin1㉿Kali)-[~]
└─$ ./TaskEC.sh
Enter your name:Kevin
Enter the name of directory:/home
Hello Kevin, the contents of the directory '/home' are:
cyse_project  Emma  kevin1  kevin2  kevin3  kevin4  kevin5  kevin6
```

```
cyse_project  Emma  kevin1  kevin2  kevin3  kevin4  kevin5  kevin6  kevin7  kloa

┌──(kevin1㉿Kali)-[~]
└─$ ./TaskEC.sh
Enter your name:Kevin
Enter the name of directory:fake
Sorry, the entered directory 'fake' name is not a valid directory name.

┌──(kevin1㉿Kali)-[~]
└─$
```



Reflection

In this lab, I began practicing how to shell script. Key concepts I learned here was how to set the variables in the script. For example, in task B, whenever the user would enter a number, it would take it and see if it is greater than 10 (in the third photo). If the number was greater than 10, it would display the text below (using echo). There were some challenges, such as when I forgot to set the permission of the files to be executable.