

ESE 2025 Week 6 Report - Bash

Instructor:

Student: Vy Nguyen

Introduction

Discussion

Question 1

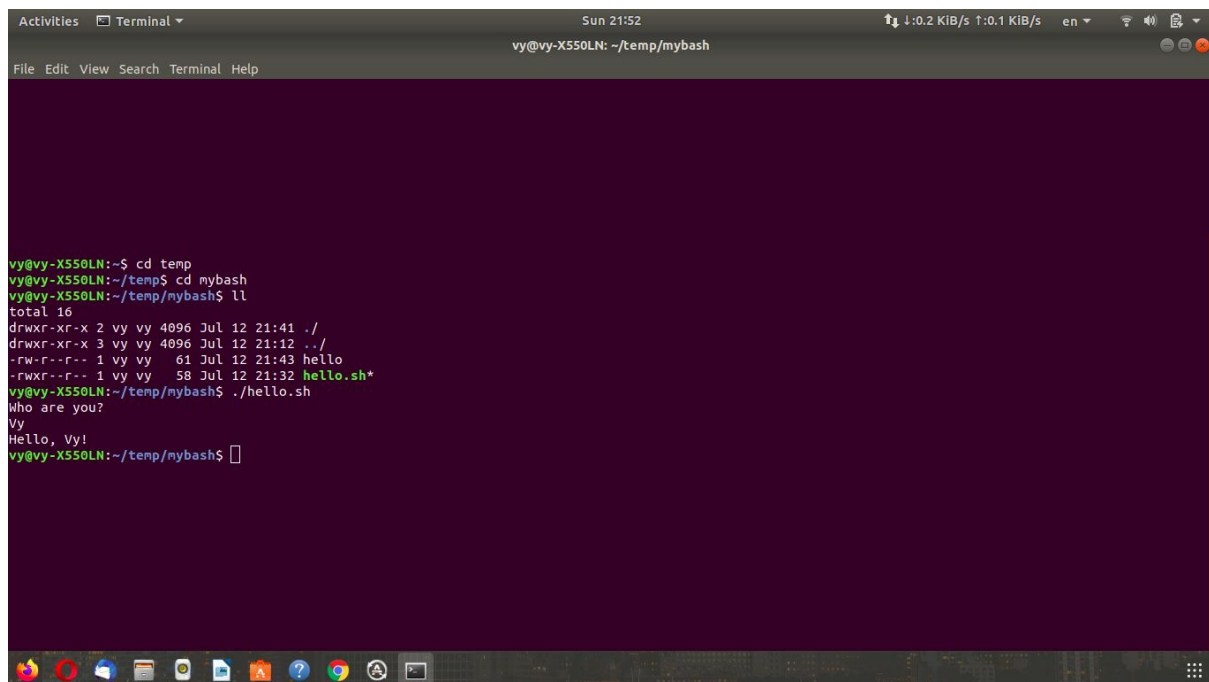
- a. Write a Bash script that accepts the user's name and prints out a "Hello <user name>!" greeting.

After creating a bash file, the script for it can be written like this

```
#!/bin/bash
echo Who are you?
read who
echo Hello, $who!
```

where "who" is a variable.

Result:

A terminal window titled "Terminal" with a dark purple background. The prompt is "vy@vy-X550LN: ~/temp/mybash". The user enters "cd temp", then "cd mybash", then "ll". The output shows a directory listing with a file named "hello.sh". The user then enters "./hello.sh". The script prompts "Who are you?", the user enters "Vy", and the script outputs "Hello, Vy!".

```
vy@vy-X550LN:~$ cd temp
vy@vy-X550LN:~/temp$ cd mybash
vy@vy-X550LN:~/temp/mybash$ ll
total 16
drwxr-xr-x 2 vy vy 4096 Jul 12 21:41 ./
drwxr-xr-x 3 vy vy 4096 Jul 12 21:12 ../
-rw-r--r-- 1 vy vy 61 Jul 12 21:43 hello
-rwxr--r-- 1 vy vy 58 Jul 12 21:32 hello.sh*
vy@vy-X550LN:~/temp/mybash$ ./hello.sh
Who are you?
Vy
Hello, Vy!
vy@vy-X550LN:~/temp/mybash$
```

- b. write a Bash script that searches the user's Documents/ folder for a file name "ese2025.txt"; if the file does not exist, create the file with string "Found First!". If the file exists, append the string "Found Again!" to the file. End your script by sending the contents of the ~/Document/ese2025.txt file to standard output.

Script:

```
#!/bin/bash

pwd
if [ -e ese2025.txt ]
then
```

```

    echo "Found Again" | tee -a ese2025.txt
else
    touch ese2025.txt
    echo "Found First" | tee -a ese2025.txt
fi
cat ese2025.txt

```

-e is used to check if the file “ese2025.txt” exists. Then write the content to ese2025.txt by using tee -a, -a means append.

cat ese2025.txt is used to display the content from the file.

The result:

```

Fri 17:15
vy@vy-X550LN: ~/Documents/folder
File Edit View Search Terminal Help
vy@vy-X550LN:~/Documents/folder$ ./find.sh
/home/vy/Documents/folder
Found Again
Found Again
Found Again
Found Again
Found Again
Found Again
vy@vy-X550LN:~/Documents/folder$ cat ese2025.txt
Found Again
Found Again
Found Again
Found Again
Found Again
vy@vy-X550LN:~/Documents/folder$

```

Comment:

After running ./find.sh the result showing 6 lines of “Found Again” because in the bash file itself, we have it to print out “Found Again” (echo).

When checking the ese2025.txt with cat, manually, we see only 5 lines that are written in the text file.

- c. **Augment the functionality of the Bash script from b) by changing "Found Again!" to "Found for the <n>th time!" where n represents the number of times that the file has been found.**

Answer:

With each time access to the bash script file, it will print out a line without overwriting the content. Then, we use wc -l to count the line, that is how we solve this problem

bash script:

```
#!/bin/bash
```

```

if [ -e ese2025.txt ]
then
    echo -n "Found for the " >> ese2025.txt | wc -l
ese2025.txt | tee -a ese2025.txt
else
    touch ese2025.txt
    echo "Found First" | tee =a ese2025.txt
fi
cat ese2025.txt

```

```

vy@vy-X550LN:~/Documents/folder$ ./find.sh
11 ese2025.txt
Found First
Found Again 1 ese2025.txt
Found Again 2 ese2025.txt
Found Again 3 ese2025.txt
Found Again 4 ese2025.txt
Found Again 5 ese2025.txt
Found Again 6 ese2025.txt
Found for the 7 ese2025.txt
Found for the 8 ese2025.txt
Found for the 9 ese2025.txt
Found for the 10 ese2025.txt
Found for the 11 ese2025.txt
vy@vy-X550LN:~/Documents/folder$

```

- d. using the Linux sort command (get information by typing "man sort"), write a Bash script that sorts 7 names entered by the user as command-line arguments. Note that if the user attempts to enter more than 7 strings, your Bash script exits with an error code 42 and the message "incorrect usage: no more than 7 elements allowed"!

Answer:

In this script, I used for loop, where it started from $i = 1$ to $i = 8$. When it is less than 8, the name will be read from the terminal. When i equal to 8, error happens.

```

#!/bin/bash
>sort.txt
for ((i=1;i<8;i++))
do
    if [ $i -lt 8 ]
    then
        echo "Enter the names"
        read name
    fi
done

```

```

        echo $name >> sort.txt
    elif [ $i == 8 ]
    then
        echo "incorrect usage, no more than 7 elements
allowed"
    fi
done
echo "0-Name after sorted-0"
sort sort.txt

```

>sort sort.txt is used to clear all the previous content in the sort.txt file.

Comment: I should use the while condition for this problem instead.

Result:

```

Activities  Terminal
Fri 21:01
vy@vy-X550LN: ~/Documents/folder

File Edit View Search Terminal Help
BATMAN
0-Name after sorted-0
Albert
Alibaba
BATMAN
Elyar
IronMan
Jarvis
Vy
vy@vy-X550LN:~/Documents/folder$ nano sort.sh
vy@vy-X550LN:~/Documents/folder$ ./sort.sh
Enter the names
Elyar
Enter the names
Vy
Enter the names
Alibaba
Enter the names
Albert
Enter the names
Jarvis
Enter the names
Ironman
Enter the names
Batman
incorrect usage, no more than 7 elements allowed
0-Name after sorted-0
Albert
Alibaba
Batman
Elyar
Ironman
Jarvis
Vy
vy@vy-X550LN:~/Documents/folder$

```

Summary:

This report sums up some of the basic commands when working with Linux system.

Some improvement should be made for better result and performance.

Appendix:

a.

```

#!/bin/bash
echo Who are you?
read who
echo Hello, $who!

```

b.

```

#!/bin/bash

pwd
if [ -e ese2025.txt ]

```

```

then

    echo "Found Again" | tee -a ese2025.txt
else
    touch ese2025.txt
    echo "Found First" | tee -a ese2025.txt
fi
cat ese2025.txt
c
#!/bin/bash

if [ -e ese2025.txt ]
then
    echo -n "Found for the " >> ese2025.txt | wc -l ese2025.txt
    | tee -a ese2025.txt
else
    touch ese2025.txt
    echo "Found First" | tee =a ese2025.txt
fi
cat ese2025.txt
d.
#!/bin/bash
>sort.txt
for ((i=1;i<8;i++))
do
    if [ $i -lt 8 ]
    then
        echo "Enter the names"
        read name
        echo $name >> sort.txt
    elif [ $i == 8 ]
    then
        echo "incorrect usage, no more than 7 elements
allowed"
    fi
done
echo "0-Name after sorted-0"

```