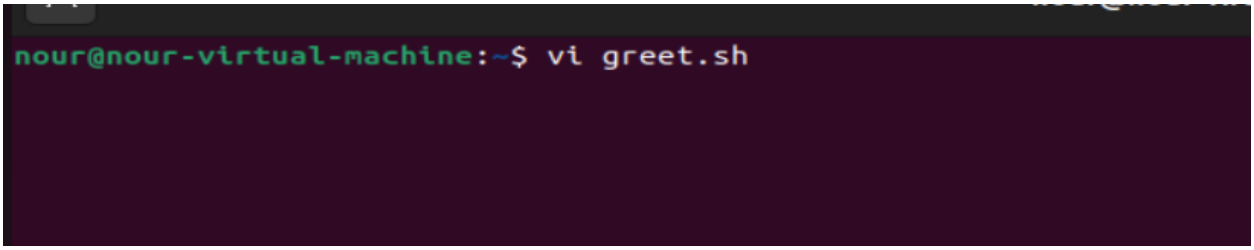
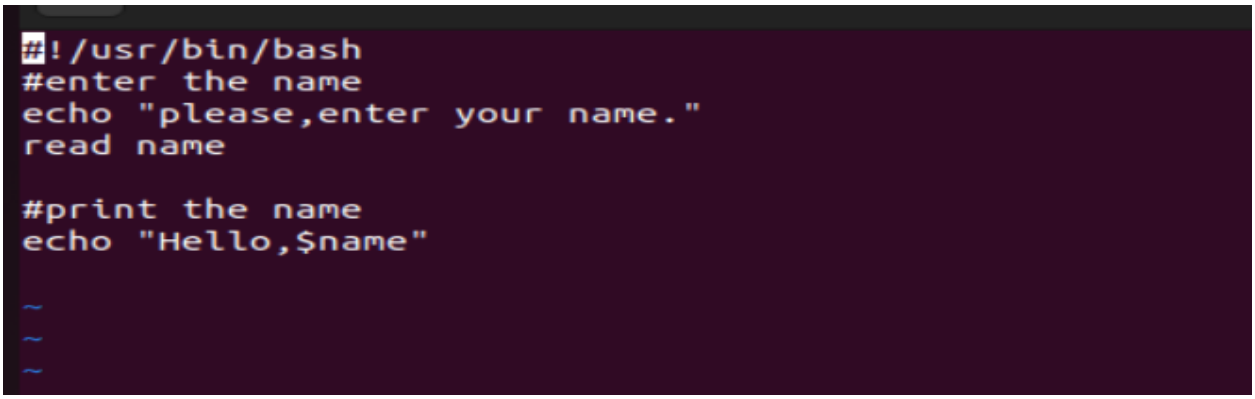
Bash Shell Script
Lab 2

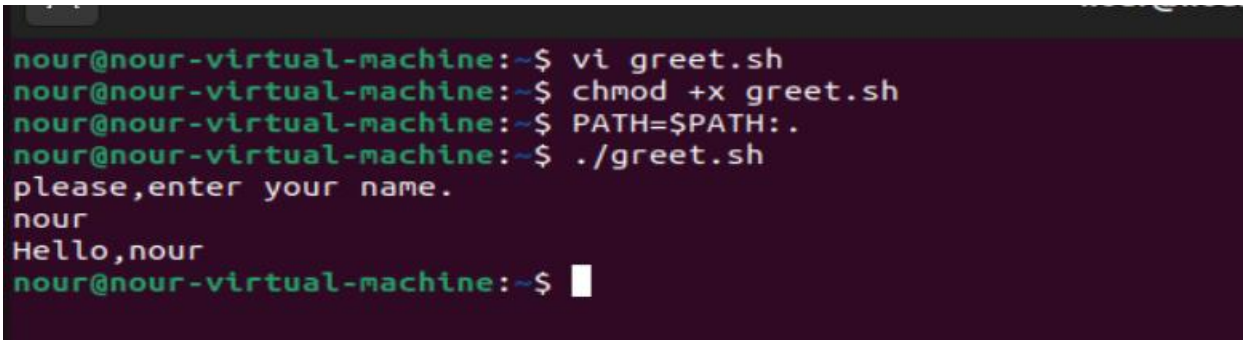
Name: Nourhan Radwan Gaber

ID: 26

1. Create a script that asks for user name then send a greeting to him.

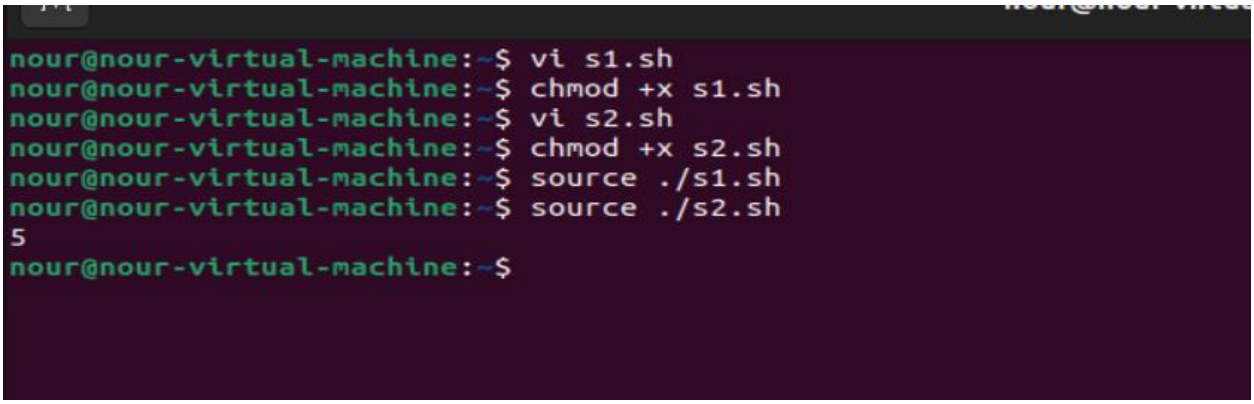






2. Create a script called s1 that calls another script s2 where:
    a. In s1 there is a variable called x, it's value 5
    b. Try to print the value of x in s2 by two different ways.

First: without call:



In this try, I make two scripts and run them in the same bash.

Second: I added “export command”:

The export command in bash is used to set an environment variable, so it makes that variable available to any child processes. This means that the variable is not only accessible in the current shell but also in any subprocesses or scripts called from that shell.

```
#!/usr/bin/bash

#first script

x=5

#calling s2


export x


./s2.sh
```

```
nour@nour-virtual-machine:~$ vi s1.sh
nour@nour-virtual-machine:~$ vi s2.sh
nour@nour-virtual-machine:~$ ./s1.sh
5
nour@nour-virtual-machine:~$
```

Third: “as an argument”:

I can use $x as an argument to s2.sh

```
#! /usr/bin/bash


#first script

x=5

#calling s2

./s2.sh $x
```

```
#! /usr/bin/bash

#second script


echo "the value is $1"

~
```

```
nour@nour-virtual-machine:~$ vi s1.sh
nour@nour-virtual-machine:~$ vi s2.sh
nour@nour-virtual-machine:~$ ./s1.sh
the value is 5
nour@nour-virtual-machine:~$
```

3. Create a script called mycp where:
    a. It copies a file to another
    b. It copies multiple files to a directory.

```
nour@nour-virtual-machine:~$ touch file3
nour@nour-virtual-machine:~$ touch file4
nour@nour-virtual-machine:~$ mkdir dir
nour@nour-virtual-machine:~$ ./mycp.sh
Usage: ./mycp.sh <source> <destination>
       ./mycp.sh <source1> <source2> ... <destination_directory>
nour@nour-virtual-machine:~$ ./mycp.sh file3 file4
Copied file3 to file4
Copy operation completed successfully.
nour@nour-virtual-machine:~$ ./mycp.sh file3 file4 dir
Copied file3 to dir
Copied file4 to dir
Copy operation completed successfully.
nour@nour-virtual-machine:~$
```

```
#!/bin/bash

# mycp - a simple script to copy files

# Check for correct number of arguments
if [ "$#" -lt 2 ]; then
    echo "Usage: $0 <source> <destination>"
    echo "       $0 <source1> <source2> ... <destination_directory>"
    exit 1
fi

# Get the last argument to determine if it's a file or directory
last_arg="${!#}"

# Check if the last argument is a directory
if [ -d "$last_arg" ]; then
    # Copy multiple files to a directory
    for ((i = 1; i < $#; i++)); do
        cp -r "${!i}" "$last_arg"
        echo "Copied ${!i} to $last_arg"
    done
elif [ "$#" -eq 2 ]; then
    # Copy a single file to another
    cp "$1" "$2"
    echo "Copied $1 to $2"
else
    # Invalid combination of arguments
    echo "Invalid combination of arguments. See usage:"
    echo "Usage: $0 <source> <destination>"
    echo "       $0 <source1> <source2> ... <destination_directory>"
    exit 1
fi

echo "Copy operation completed successfully."
```

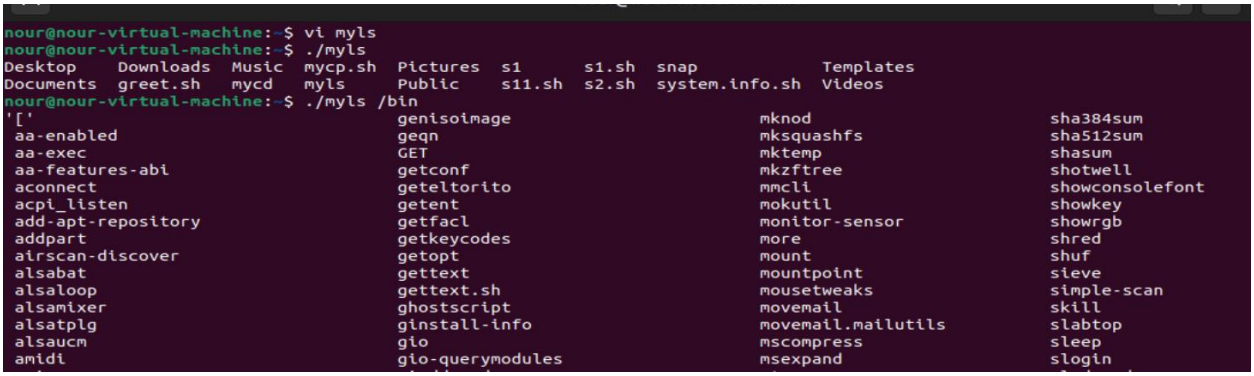
4. Create a script called mycd where:
    a. It changed directory to the user home directory, if it is called without arguments.
    b. Otherwise, it change directory to the given directory.

```
#! /usr/bin/bash


if [ "$#" -eq 0 ]
then
  cd "$HOME"


 else
  cd "$1"
fi
```

5. Create a script called myls where:
   a. It lists the current directory, if it is called without arguments.
   b. Otherwise, it lists the given directory.



6. Enhance the above script to support the following options individually:
    a. –l: list in long format
    b. –a: list all entries including the hiding files.
    c. –d: if an argument is a directory, list only its name
    d. –i: print inode number
    e. –R: recursively list subdirectories

```
#! /usr/bin/bash

 Check if no arguments are provided, list the current directory
if [ $# -eq 0 ]; then
  ls
  exit 0
fi

# Process options
if [ "$1" == "-l" ]; then
  # List in long format
  ls -l "$2"
elif [ "$1" == "-a" ]; then
  # List all entries, including hidden files
  ls -a "$2"
elif [ "$1" == "-d" ]; then
  # If an argument is a directory, list only its name
  if [ -d "$2" ]; then
    echo "$2"
  else
    echo "$2 is not a directory."
    exit 1
  fi
elif [ "$1" == "-i" ]; then
  # Print inode number
  ls -i "$2"
elif [ "$1" == "-R" ]; then
  # Recursively list subdirectories
  ls -R "$2"
else
  # If no options are provided, list the given directory
  ls "$1"
fi
```

7. Create a script called mytest where:
    a. It check the type of the given argument (file/directory)
    b. It check the permissions of the given argument (read/write/execute)

```
nour@nour-virtual-machine:~$ ./test.sh mycd
mycd is a file.
Permissions for mycd:
Read permission granted
Write permission granted
Execute permission granted
nour@nour-virtual-machine:~$ ./test.sh /etc
/etc is a directory.
Permissions for /etc:
Read permission granted
No write permission
Execute permission granted
nour@nour-virtual-machine:~$
```

```
#! /usr/bin/bash
if [ -z "$1" ]; then
  echo "Usage: $0 <file_or_directory>"
  exit 1
fi

# Check if the provided path exists
if [ ! -e "$1" ]; then
  echo "Error: The specified path does not exist."
  exit 1
fi

# Check if the argument is a file or directory
if [ -f "$1" ]; then
  echo "$1 is a file."
elif [ -d "$1" ]; then
  echo "$1 is a directory."
else
  echo "$1 is neither a file nor a directory."
fi

# Check permissions
echo "Permissions for $1:"
[ -r "$1" ] && echo "Read permission granted" || echo "No read permission"
[ -w "$1" ] && echo "Write permission granted" || echo "No write permission"
[ -x "$1" ] && echo "Execute permission granted" || echo "No execute permission"
```

8. Create a script called myinfo where:
    a. It asks the user about his/her logname.
    b. It print full info about files and directories in his/her home directory
    c. Copy his/her files and directories as much as you can in /tmp directory.
    d. Gets his current processes status.

```
#! /usr/bin/bash
# Ask for logname
read -p "Enter your logname: " logname

# Print full info about files and directories in the home directory
echo "Full info about files and directories in the home directory:"
ls -l /home/"$logname"

# Copy files and directories to /tmp
echo "Copying files and directories to /tmp..."
cp -r /home/"$logname"/* /tmp/

# Get current processes status
echo "Current processes status:"
ps aux

~
```

```
nour@nour-virtual-machine:~$ vi myinfo
nour@nour-virtual-machine:~$ ./myinfo
Enter your logname: nour
Full info about files and directories in the home directory:
total 84
drwxr-xr-x 2 nour nour 4096 23:20 21 نوف Desktop
drwxr-xr-x 2 nour nour 4096 23:20 21 نوف Documents
drwxr-xr-x 2 nour nour 4096 23:20 21 نوف Downloads
-rwxrwxr-x 1 nour nour  111 13:26 11 نوس greet.sh
drwxr-xr-x 2 nour nour 4096 23:20 21 نوف Music
-rwxrwxr-x 1 nour nour   99 18:22 11 نوس mycd
-rw-rw-r-- 1 nour nour   22 16:40 11 نوس mycp.sh
-rwxrwxr-x 1 nour nour  425 20:10 11 نوس myinfo
-rwxrwxr-x 1 nour nour  341 19:56 11 نوس myls
-rwxrwxr-x 1 nour nour  641 19:07 11 نوس myls.sh
drwxr-xr-x 2 nour nour 4096 23:20 21 نوف Pictures
drwxr-xr-x 2 nour nour 4096 23:20 21 نوف Public
-rwxrwxr-x 1 nour nour   35 14:19 11 نوس s1
-rwxrwxr-x 1 nour nour   21 14:31 11 نوس s11.sh
-rwxrwxr-x 1 nour nour   74 15:17 11 نوس s1.sh
-rwxrwxr-x 1 nour nour   59 15:17 11 نوس s2.sh
drwx------ 4 nour nour 4096 19:45 11 نوس snap
-rwxrwxr-x 1 nour nour   82 18:25 23 نوف system.info.sh
drwxr-xr-x 2 nour nour 4096 23:20 21 نوف Templates
-rwxrwxr-x 1 nour nour  692 20:05 11 نوس test.sh
drwxr-xr-x 2 nour nour 4096 23:20 21 نوف Videos
Copying files and directories to /tmp...
Current processes status:
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.2  0.1 166612  2688 ?        Ss   18:10   0:15 /sbin/init auto noprompt splash
root           2  0.0  0.0      0     0 ?        S    18:10   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   18:10   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   18:10   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   18:10   0:00 [slub_flushwq]
root           6  0.0  0.0      0     0 ?        I<   18:10   0:00 [netns]
```