Shell Scripting

Day2

Nada Mohamed Ahmed Hassan Eleshmawy

Mansoura Open Source

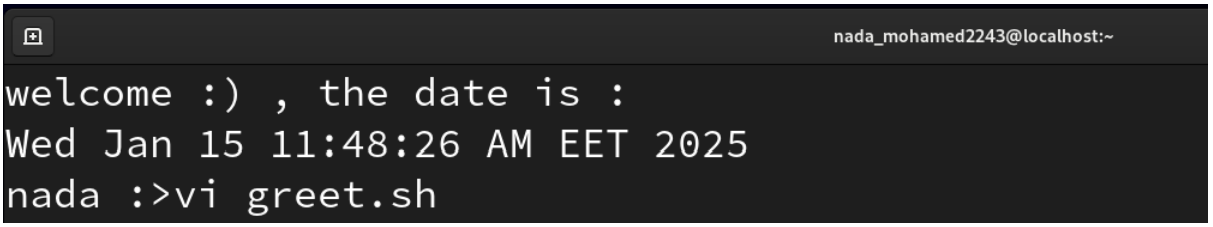
Date : 15/1/2025

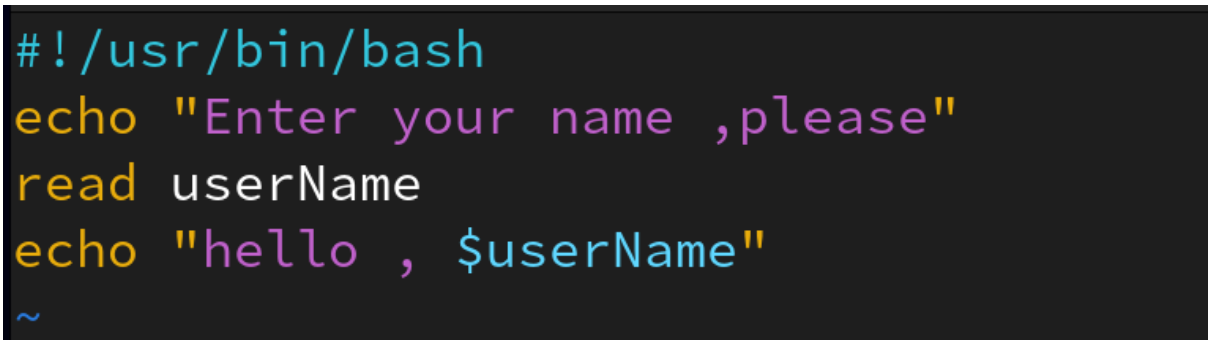
1. Create a script that asks for user name then send a greeting to him.

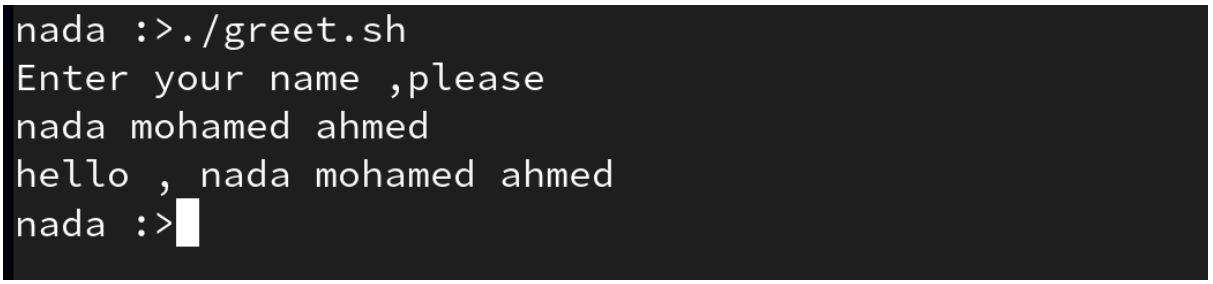
vi greet.sh

#!/usr/bin/bash
echo “Enter your name please : ”
read userName
echo “hello , $userName”

ls -l greet.sh
chmod +x
./greet.sh







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.

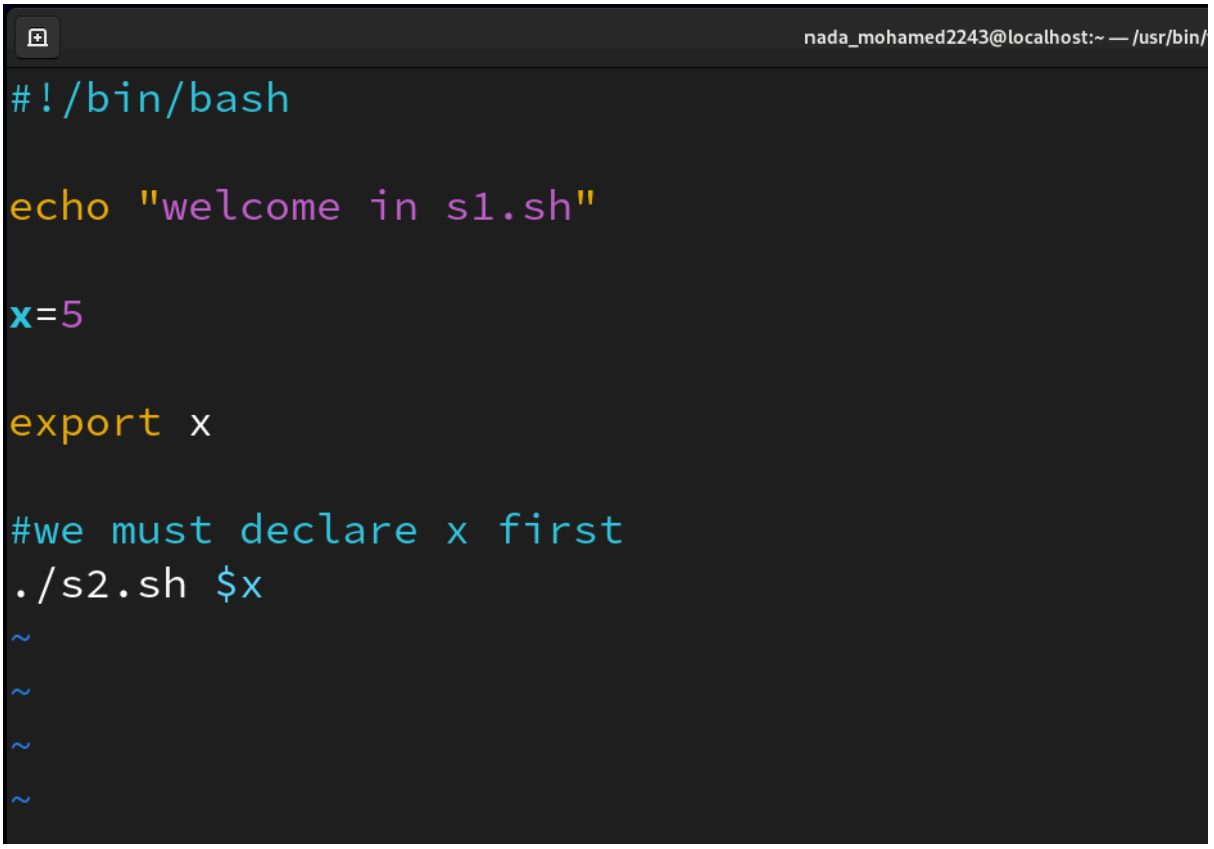
First method : as an argument

Second method : by export

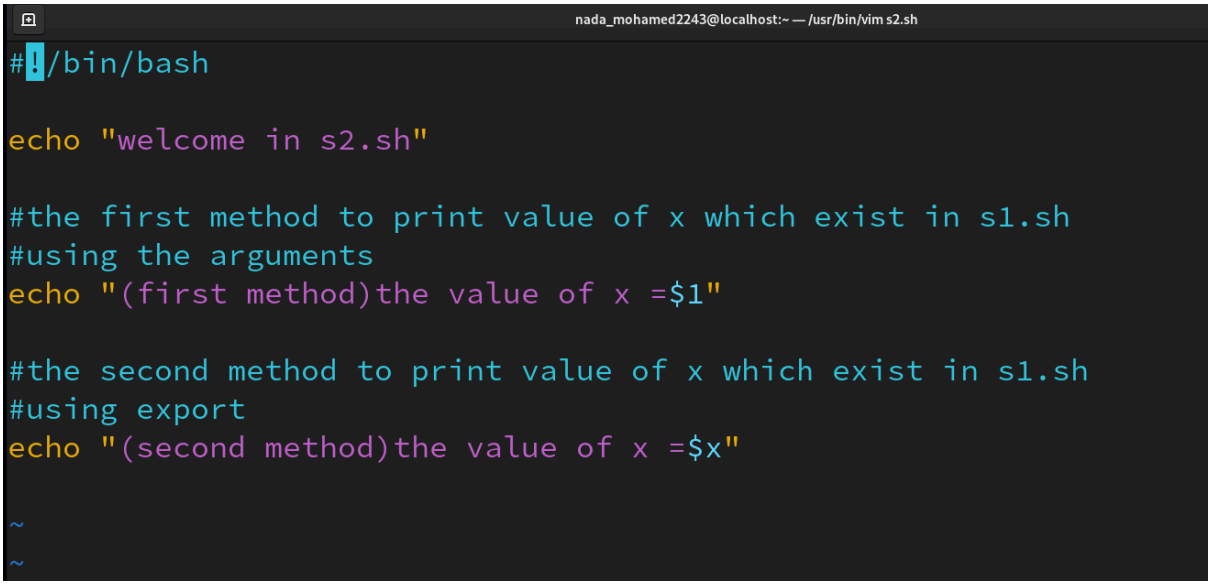
Note : export only done from parent to child

Create s1.sh

```
nada :>vi s1.sh
```



Create s2.sh



Modify files permission:

```
nada :>ls -l s1.sh
-rw-r--r--. 1 nada_mohamed2243 nada_mohamed2243 30 Jan 15 12:20 s1.sh
nada :>chmod +x s1.sh
nada :>ls -l s1.sh
-rwxr-xr-x. 1 nada_mohamed2243 nada_mohamed2243 30 Jan 15 12:20 s1.sh
```

```
nada :>ls -l s2.sh
-rw-r--r--. 1 nada_mohamed2243 nada_mohamed2243 279 Jan 15 12:29 s2.sh
nada :>chmod +x s2.sh
nada :>ls -l s2.sh
-rwxr-xr-x. 1 nada_mohamed2243 nada_mohamed2243 279 Jan 15 12:29 s2.sh
```

run s1.sh

```
nada :>./s1.sh
welcome in s1.sh
welcome in s2.sh
(first method)the value of x =5
(second method)the value of x =5
```

3. Create a script called mycp where:

a. It copies a file to another

b. It copies multiple files to a directory.

Note : we can get the last arg using ${!#} or ${@: -1}

Get all arg except the last one : ${@:1:$#-1}

```
nada :>vi mycp.sh
```

nada_mohamed2243@localhost:~ — /usr/bin/vim mycp.sh

```
#!/bin/bash
#check we have less than two arg
if [ $# -lt 2 ]; then
        echo "you must input the target"
#check we have exactly two arg (file - file) (file - dir)
elif [ $# -eq 2 ]; then
        if [ -f $2 ]; then
                echo "---------contant of file1 before copy -----------"
                cat $1
                echo "---------contact of file2 before copy-----------"
                cat $2
                cp "$1" "$2"
                echo "---------contact of file1 after copy-----------"
                cat $1
                echo "---------contact of file2 after copy-----------"
                cat $2
                echo "---------------------"
                echo "copy file $1 to file $2 successfully"
        else
                echo "---------contact of file1-----------"
                cat $1
                echo "---------contact of dir before copy-------------"
                ls $2
                cp "$1" "$2"
```

```
                cp "$1" "$2"
                echo "---------contact of dir after copy--------------"
                ls $2
                echo "--------------------"
                echo "copy file $1 to dir $2 successfully"
        fi
#check we have more than 2 files
else
        echo "${@: -1}"
        if [ -f ${@: -1} ]; then
                echo "please input the target directory"
        else
                cp "${@:1:$#-1}" "${@: -1}"
                echo "copy files ${@:1:$#-1} to dir ${@: -1} successfully"
        fi
fi
```

Input one arg

```
nada :>./mycp.sh test1.txt
you must input the target
```

Copy one file to another :

```
nada :>./mycp.sh test1.txt test2.txt
---------contant of file1 before copy -----------
#we are in test1.txt
test1
---------contact of file2 before copy-----------
#we are in test2.txt
test2
---------contact of file1 after copy-----------
#we are in test1.txt
test1
---------contact of file2 after copy-----------
#we are in test1.txt
test1
--------------------
copy file test1.txt to file test2.txt successfully
```

Copy one file into dir

```
nada :>./mycp.sh test1.txt testdir
---------contact of file1-----------
#we are in test1.txt
test1
---------contact of dir before copy-------------
---------contact of dir after copy--------------
test1.txt
-------------------
copy file test1.txt to dir testdir successfully
nada :>
```

Try to input 3 files

```
nada :>./mycp.sh test1.txt test2.txt test1.txt
test1.txt
please input the target directory
```

Input 2 files into dir

```
nada :>./mycp.sh test1.txt test2.txt testdir
testdir
copy files test1.txt test2.txt to dir testdir successfully
nada :>
```

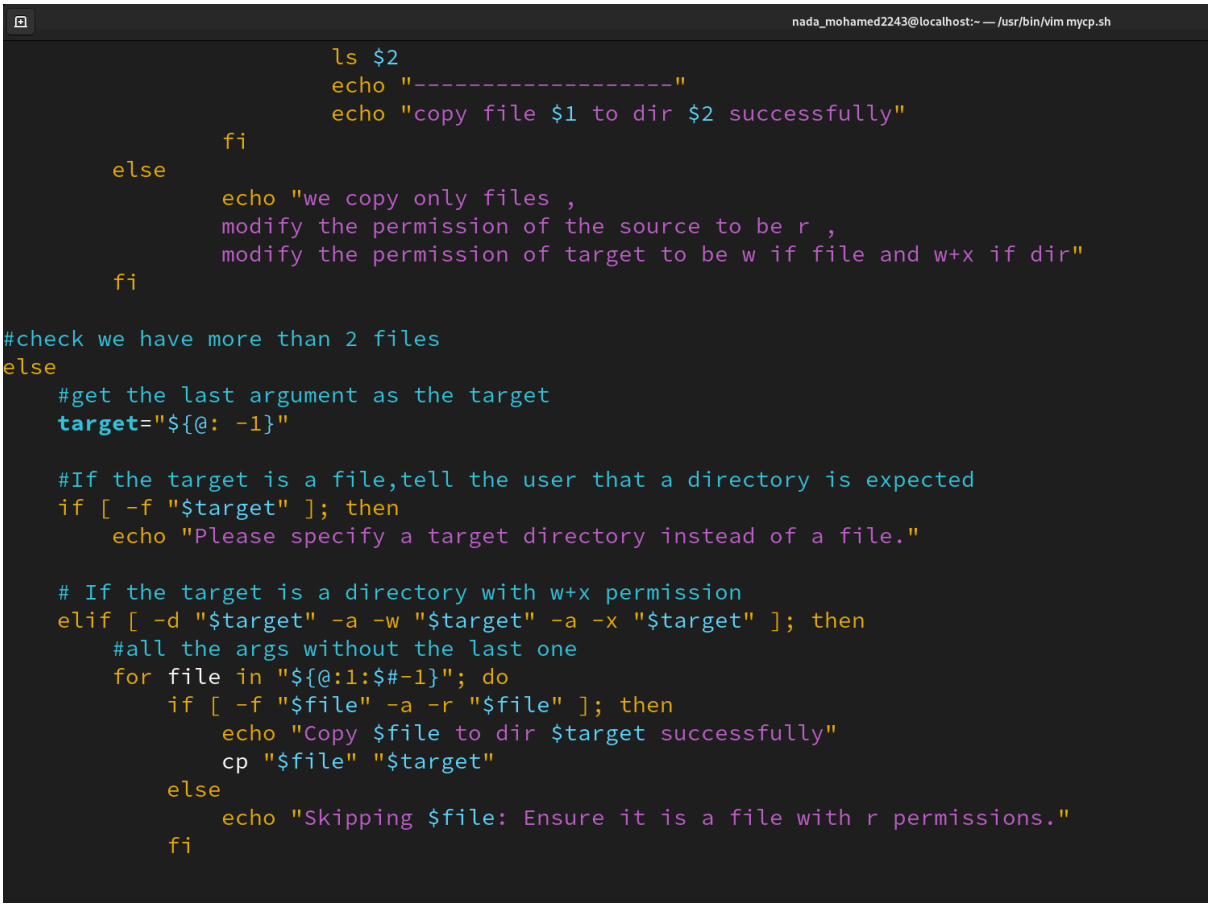
```
nada :>cd testdir
nada :>ls
test1.txt  test2.txt
nada :>
```

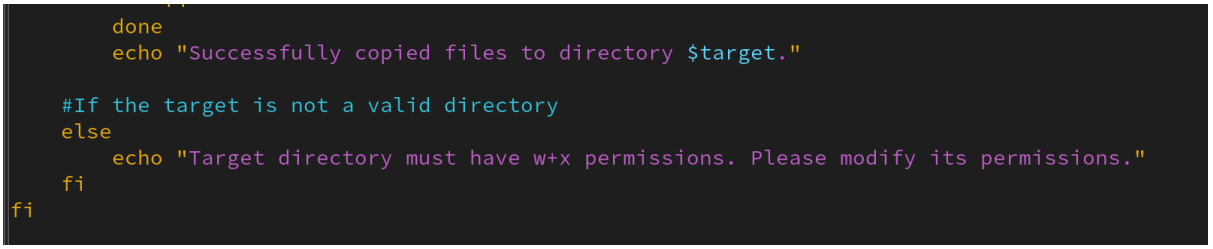
If we focus on permissions

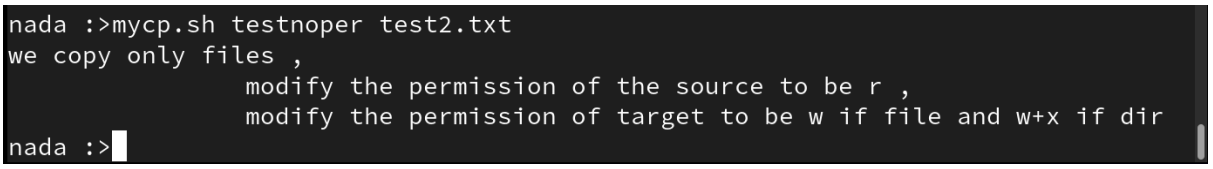
Copy only files , source files with permission r , target files with permission w , target dir with permission w+x

nada_mohamed2243@localhost:~ — /usr/bin/vim mycp.sh

```
#!/bin/bash
#check we have less than two arg
if [ $# -lt 2 ]; then
        echo "you must input at least file and the target"
#check we have exactly two arg (file - file) (file - dir)
# we copy only files
#the file permission should be r
#the target permissiom should be w (if dir w+x)
elif [ $# -eq 2 ]; then
        if [ -f $1 -a  -r $1 -a -w $2 ]; then
                if [ -f $2 ]; then
                        echo "---------contant of file1 before copy -----------"
                        cat $1
                        echo "---------contact of file2 before copy-----------"
                        cat $2
                        cp "$1" "$2"
                        echo "---------contact of file1 after copy-----------"
                        cat $1
                        echo "---------contact of file2 after copy-----------"
                        cat $2
                        echo "--------------------"
                        echo "copy file $1 to file $2 successfully"
                elif [ -d $2 -a -x $2 ]; then
                        echo "---------contact of file1-----------"
                        cat $1
                        echo "---------contact of dir before copy-------------"
                        ls $2
                        cp "$1" "$2"
                        echo "---------contact of dir after copy--------------"
```


nada_mohamed2243@localhost:~ — /usr/bin/vim mycp.sh
ls $2
echo "--------------------"
echo "copy file $1 to dir $2 successfully"
fi
else
echo "we copy only files ,
modify the permission of the source to be r ,
modify the permission of target to be w if file and w+x if dir"
fi
#check we have more than 2 files
else
#get the last argument as the target
target="${@: -1}"
#If the target is a file,tell the user that a directory is expected
if [-f "$target"]; then
echo "Please specify a target directory instead of a file."
If the target is a directory with w+x permission
elif [-d "$target" -a -w "$target" -a -x "$target"]; then
#all the args without the last one
for file in "${@:1:$#-1}"; do
if [-f "$file" -a -r "$file"]; then
echo "Copy $file to dir $target successfully"
cp "$file" "$target"
else
echo "Skipping $file: Ensure it is a file with r permissions."
fi


done
echo "Successfully copied files to directory $target."
#If the target is not a valid directory
else
echo "Target directory must have w+x permissions. Please modify its permissions."
fi
fi


nada :>mycp.sh testnoper test2.txt
we copy only files ,
modify the permission of the source to be r ,
modify the permission of target to be w if file and w+x if dir
nada :>

```
nada :>vi test2.txt
nada :>mycp.sh test1.txt test2.txt
---------contant of file1 before copy -----------
#we are in test1.txt
test1
---------contact of file2 before copy-----------
#we are in test2.txt
test2
---------contact of file1 after copy-----------
#we are in test1.txt
test1
---------contact of file2 after copy-----------
#we are in test1.txt
test1
--------------------
copy file test1.txt to file test2.txt successfully
nada :>
```

```
nada :>mycp.sh test2dir test3dir
we copy only files ,
                modify the permission of the source to be r+x ,
                modify the permission of target to be w+x
```

```
nada :>mycp.sh test1.txt test2.txt testdir
Copy test1.txt to dir testdir successfully
Copy test2.txt to dir testdir successfully
Successfully copied files to directory testdir.
nada :>ls testdir
test1.txt  test2.txt
nada :>
```

```
nada :>mycp.sh test1.txt test2.txt test
Target directory must have w+x permissions. Please modify its permissions.
nada :>mycp.sh test1.txt test2.txt testnoper
Please specify a target directory instead of a file.
nada :>ls -ld test
d---------. 2 nada_mohamed2243 nada_mohamed2243 6 Dec 25 13:53 test
nada :>
```

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.

Create file

```
nada :>vi mycd.sh
```

```
#!/bin/bash
#called without argument
if [ $# -eq 0 ]; then
        echo "---the path before execution---"
        pwd
        echo "welcome to home directory"
        cd
        echo "---the path after execution---"
        pwd
else
        echo "---the path before execution---"
        pwd
        echo "welcome in $1 directory"
        cd $1
        echo "---the path after execution---"
        pwd
fi


~
```

Handle the permissions

```
nada :>ls -l mycd.sh
-rw-r--r--. 1 nada_mohamed2243 nada_mohamed2243 145 Jan 15 16:56 mycd.sh
nada :>chmod +x mycd.sh
nada :>ls -l mycd.sh
-rwxr-xr-x. 1 nada_mohamed2243 nada_mohamed2243 145 Jan 15 16:56 mycd.sh
```

```
nada :>. ./mycd.sh testdir
---the path before execution---
/home/nada_mohamed2243
welcome in testdir directory
---the path after execution---
/home/nada_mohamed2243/testdir
nada :>pwd
/home/nada_mohamed2243/testdir
```

```
nada :>. ../mycd.sh
---the path before execution---
/home/nada_mohamed2243/testdir
welcome to home directory
---the path after execution---
/home/nada_mohamed2243
nada :>pwd
/home/nada_mohamed2243
nada :>
```

Takecare :

If run a script, it runs in a subshell, meaning it doesn't affect the current shell session.

So when you run `./mycd.sh` , the directory change happens in the subshell, but once the script finishes, the changes are lost, and we are back in the original shell session.

```
nada :>./mycd.sh testdir
---the path before execution---
/home/nada_mohamed2243
welcome in testdir directory
---the path after execution---
/home/nada_mohamed2243/testdir
nada :>pwd
/home/nada_mohamed2243
```

So we don’t make ./mycd.sh but make . ./mycd.sh or source ./mycd.sh

Sourcing a script runs it in the current shell, allowing the `cd` command to change the directory of the current shell.

(The `cp` (copy) command works differently from `cd` because it doesn't change the current shell session's state. Instead, it simply performs an operation (copying files or directories) without affecting the environment or session.)

Instead of run script using relative path we can modify the environment variable PATH

```
nada :>PATH=$PATH:$HOME
```

```
nada :>. mycd.sh testdir
---the path before execution---
/home/nada_mohamed2243
welcome in testdir directory
---the path after execution---
/home/nada_mohamed2243/testdir
nada :>pwd
/home/nada_mohamed2243/testdir
nada :>. mycd.sh
---the path before execution---
/home/nada_mohamed2243/testdir
welcome to home directory
---the path after execution---
/home/nada_mohamed2243
```

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.

create script

```
nada :>vi myls.sh
```

nada_mohamed2243@localhost:~ — /usr/bin/vim myls.sh

```
#!/bin/bash
#without arg list current dir

if [ $# -eq 0 ]; then
        ls
else
        ls $1
fi

~
```

Modify permissions

```
nada :>ls -l myls.sh
-rw-r--r--. 1 nada_mohamed2243 nada_mohamed2243 85 Jan 15 23:31 myls.sh
nada :>chmod +x myls.sh
nada :>ls -l myls.sh
-rwxr-xr-x. 1 nada_mohamed2243 nada_mohamed2243 85 Jan 15 23:31 myls.sh
```

Run script

```
nada :>myls.sh
Desktop    filename2  Music    myteam     s2.sh        test2.txt
Documents  greet.sh   mycd.sh  oldpasswd  sortedUsers  testdir
Downloads  hi         mycp.sh  Pictures   Templates    text2.txt
file       ls_errors  mycv     Public     test         Videos
filename1  ls_output  myls.sh  s1.sh      test1.txt
nada :>myls.sh testdir
test1.txt  test2.txt
```

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**

Input only dirs , input only this 5 options

```
nada_mohamed2243@localhost:~ — /usr/bin/vim myls.sh
#!/bin/bash
#without arg list current dir

if [ $# -eq 0 ]; then
        ls
else
        if [ $1 == "-l" -o $1 == "-a" -o $1 == "-d" -o $1 == "-i" -o $1 == "-R"  ];
 then
                if [ $# -eq 1 ]; then
                        ls $1
                else
                        if [ -d $2 ]; then
                                ls $1 $2
                        else
                                echo "Enter the dir name correctly"
                        fi
                fi
        else
                echo "invalid option"

        fi
fi

"myls.sh" 22L, 331B                                           14,11-39      All
```

```
nada :>myls.sh -l test1.txt
Enter the dir name correctly
```

```
nada :>myls.sh -d testdir
testdir
```

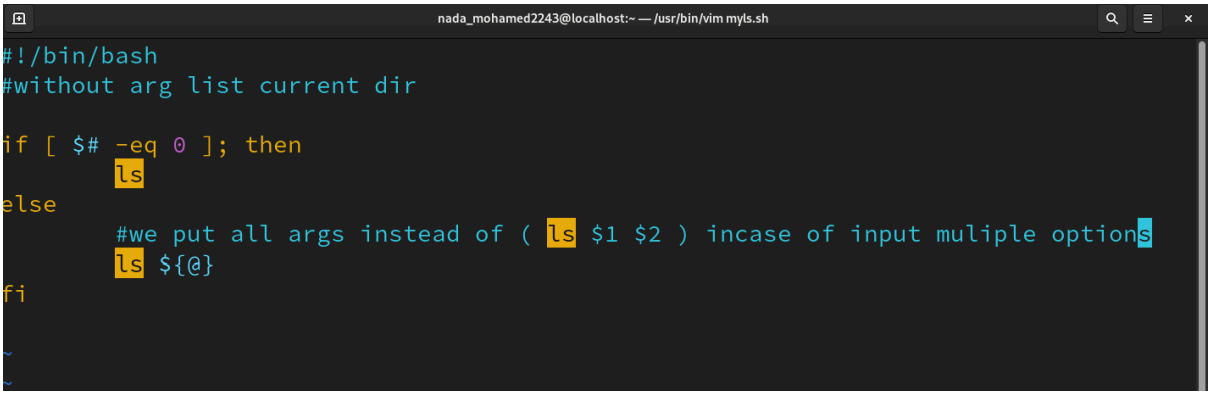
```
nada :>myls.sh -p testdir
invalid option
```

```
nada :>myls.sh -i
52068613 Desktop     1144251 ls_errors   1144237 oldpasswd     1144706 test1.txt
52068615 Documents   1144248 ls_output  17554575 Pictures      1144707 test2.txt
 1143735 Downloads   1143736 Music      35944621 Public        1248495 testdir
 1144236 file        1144709 mycd.sh     1248492 s1.sh         1248494 text2.txt
 1248450 filename1   1144705 mycp.sh     1248491 s2.sh        35944623 Videos
 1248474 filename2   1143718 mycv        1143734 sortedUsers
 1144240 greet.sh    1144714 myls.sh    17554574 Templates
 1248493 hi         17554618 myteam      1144233 test
```

Bonus:
enhance the above script to support the following Synopsis:

myls -option1 –option2
myls –option2 –option1
myls –option1option2
myls –option2option1

Generally any option , any type of input




```
nada :>myls.sh -a -l
total 21812
drwx------. 17 nada_mohamed2243 nada_mohamed2243     4096 Jan 16 00:18 .
drwxr-xr-x.  5 root             root                   56 Dec 29 15:07 ..
-rw-------.  1 nada_mohamed2243 nada_mohamed2243     1356 Dec 29 14:58 .bash_histor
y
-rw-r--r--.  1 nada_mohamed2243 nada_mohamed2243       18 Feb 15  2024 .bash_logout
-rw-r--r--.  1 nada_mohamed2243 nada_mohamed2243      141 Feb 15  2024 .bash_profil
e
```

```
nada :>myls.sh -d -l testdir
drwxr-xr-x. 2 nada_mohamed2243 nada_mohamed2243 40 Jan 15 23:44 testdir
nada :>myls.sh -d testdir
testdir
```

```
nada_mohamed2243@localhost:~ — /usr/bin/vim myls.sh
#!/bin/bash
#without arg list current dir
#echo $@
if [ $# -eq 0 ]; then
        ls
else
        options=""
        dir=""
        #associative array(maps ls options to their descriptions):
        declare -A optionDescriptions=(
        ["l"]="list in long format"
        ["a"]="show all files, including hidden ones"
        ["d"]="list directories themselves, not their contents"
        ["i"]="print the index number of each file"
        ["R"]="list directories recursively")

        #Loop Through Arguments
        for arg in "$@"; do
        # Check if the argument starts with a dash
        if [[ $arg == -* ]]; then
                # Loop through the argument itself ex: -la (-, l, a)
                #start from 1 to not include -
                for (( i=1; i<${#arg}; i++ )); do
                #Extracts a substring from arg at position i and length 1.
                #ex : la --> l then a
                        option="${arg:$i:1}"
                        if [[ ${optionDescriptions[$option]} ]]; then
                                options="$options -$option"
                                echo "-$option: ${optionDescriptions[$option]}"
-- INSERT --
```

```
                        else
                                echo "Invalid option: -$option"
                        fi
                done
        elif [ -d "$arg" ]; then
                dir="$arg"
        else
                echo "option ${arg} should start with '-' but if you try to input dir name it's not exist"
        fi
done
        ls $options $dir

fi
-- INSERT --
```

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)

```
nada :>vi mytest.sh
```

nada_mohamed2243@localhost:~ — /usr/bin/vim

```
#!/bin/bash

if [ $# -lt 1 ]; then
        echo "please, input file or dir"
else
        if [ -f $1 ]; then
                echo "$1 is file"
        elif [ -d $1 ]; then
                echo "$1 is dir"
        else
                echo "$1 is not file or dir"
        fi


        if [ -r $1 ]; then
                echo "your input $1 is readable"
        else
                echo "your input $1 is not readable"
        fi


        if [ -w $1 ]; then
                echo "your input $1 is writable"
        else
                echo "your input $1 is not writable"
        fi

        if [ -x $1 ]; then
                echo "your input $1 is executable"
```

```
        if [ -x $1 ]; then
                echo "your input $1 is executable"
        else
                echo "your input $1 is not executable"
        fi
fi
```

```
nada :>mytest.sh
please, input file or dir
nada :>mytest.sh test1.txt
test1.txt is file
your input test1.txt is readable
your input test1.txt is writable
your input test1.txt is not executable
nada :>ls test1.txt
test1.txt
nada :>ls -l test1.txt
-rw-r--r--. 1 nada_mohamed2243 nada_mohamed2243 27 Jan 15 16:22 test1.txt
nada :>mytest.sh testdir
testdir is dir
your input testdir is readable
your input testdir is writable
your input testdir is executable
```

8. Create a script called myinfo where:

a. It asks the user about his/her logname.

Using read

b. It print full info about files and directories in his/her home directory

ls -a -l nada_mohamed2243

c. Copy his/her files and directories as much as you can in /tmp directory.

If i want to copy dir the permission i needed is : read + execute

If i want to copy file the permission i needed is : read

And target dir : write and execute

cp -r /home/$logname/* /tmp/

d. Gets his current processes status.

ps -u nada_mohamed2243

`-u` option for the `ps` command is used to display processes for a specific user.

```
nada :>vi myinfo.sh
```

nada_mohamed2243@localhost:~ — /usr/bin/vim myinfo.sh

```
#!/bin/bash

echo "Enter your logname,please : "
read logname

echo "-----------------------------------------"
echo "Full info about files and directories :"
ls -l -a /home/$logname

echo "--------------------------------"
echo "Copying files and directories"
cp -r /home/$logname/* /tmp/
echo "copy process with permission r+x is done sucessfully"

echo "---------------------"
echo "the permission of target dir (must be w+x)"
ls -ld /tmp
echo "the contant of target dir"
ls -l /tmp

echo "--------------------------------"
echo "Current processes status for $logname:"
ps -u $logname
```

1,11

```
nada :>myinfo.sh
Enter your logname,please :
nada_mohamed2243
-----------------------------------------
Full info about files and directories :
total 21816
drwx------. 17 nada_mohamed2243 nada_mohamed2243     4096 Jan 16 10:05 .
drwxr-xr-x.  5 root             root                   56 Dec 29 15:07 ..
-rw-------.  1 nada_mohamed2243 nada_mohamed2243     2808 Jan 16 01:17 .bash_his
tory
```

```
--------------------------------
Copying files and directories
cp: cannot open '/home/nada_mohamed2243/file' for reading: Permission denied
cp: cannot access '/home/nada_mohamed2243/test': Permission denied
copy process with permission r+x is done sucessfully
---------------------
the permission of target dir (must be w+x)
drwxrwxrwt. 43 root root 4096 Jan 16 10:00 /tmp
the contant of target dir
total 21756
drwx------.  2 nada_mohamed2243 nada_mohamed2243        6 Jan 16 09:30 ahmed
drwxr-xr-x.  2 nada_mohamed2243 nada_mohamed2243        6 Jan 16 02:36 Desktop
drwxr-xr-x.  2 nada_mohamed2243 nada_mohamed2243        6 Jan 16 02:36 Documents
drwxr-xr-x.  2 nada_mohamed2243 nada_mohamed2243        6 Jan 16 02:36 Downloads
```

```
------------------------------
Current processes status for nada_mohamed2243:
    PID TTY          TIME CMD
   2279 ?        00:00:00 systemd
   2281 ?        00:00:00 (sd-pam)
   2297 ?        00:00:00 gnome-keyring-d
   2301 tty2     00:00:00 gdm-wayland-ses
   2305 ?        00:00:00 dbus-broker-lau
   2306 ?        00:00:00 dbus-broker
```