

OST LAB

LAB 2 ASSIGNMENT SUBMISSION

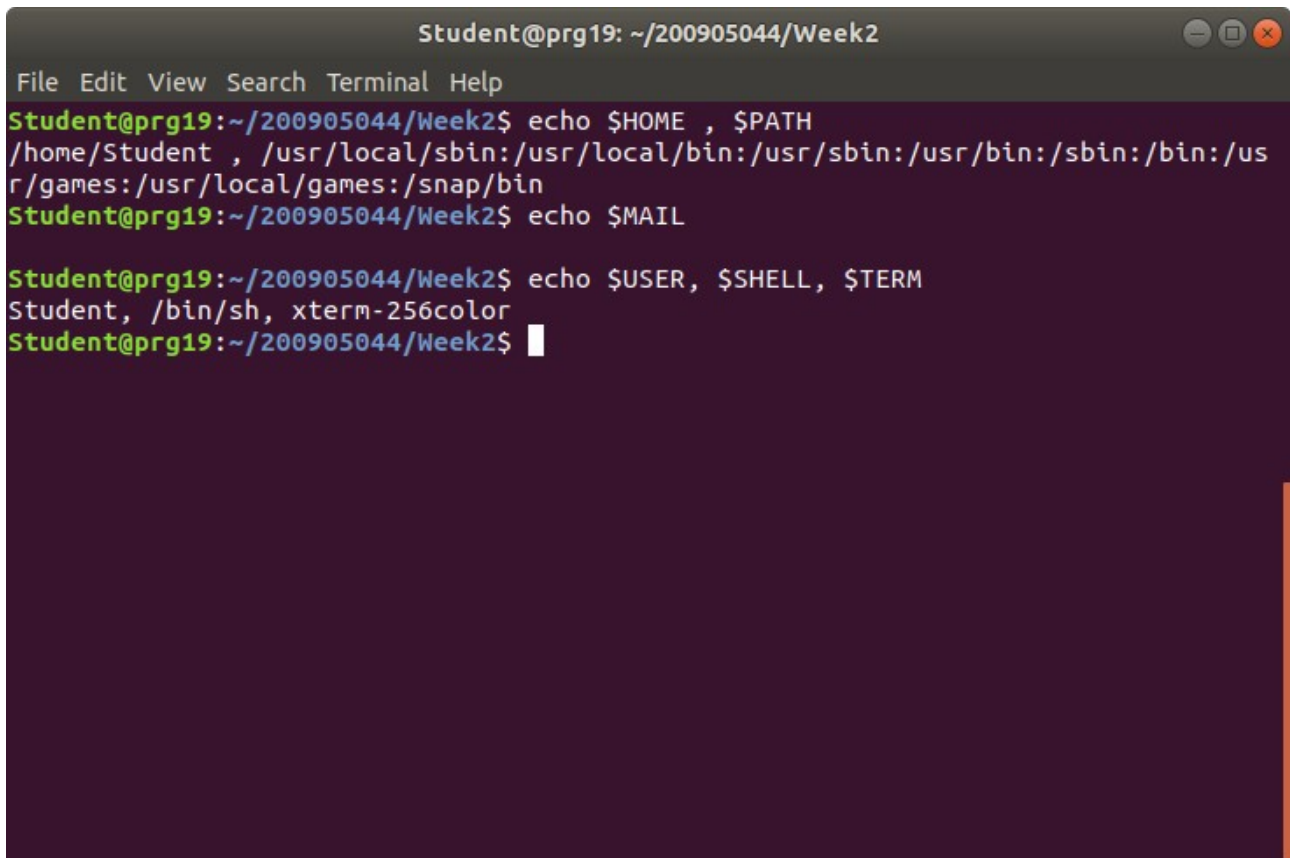
I)

1) Try the following shell commands

```
$ echo $HOME , $PATH
```

```
$ echo $MAIL
```

```
$ echo $USER, $SHELL, $TERM
```

A screenshot of a terminal window titled "Student@prg19: ~/200905044/Week2". The terminal has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The background is dark purple. The text shows the following commands and their outputs:
1. Command: `Student@prg19:~/200905044/Week2$ echo $HOME , $PATH`
Output: `/home/Student , /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin`
2. Command: `Student@prg19:~/200905044/Week2$ echo $MAIL`
Output: (empty line)
3. Command: `Student@prg19:~/200905044/Week2$ echo $USER, $SHELL, $TERM`
Output: `Student, /bin/sh, xterm-256color`
4. Command: `Student@prg19:~/200905044/Week2$`
Output: (empty line with a cursor)

2) Try the following code snippet-

```
$ firstname=Rakesh
```

```
$ lastname=Sharma
```

```
$ echo $firstname $lastname
```

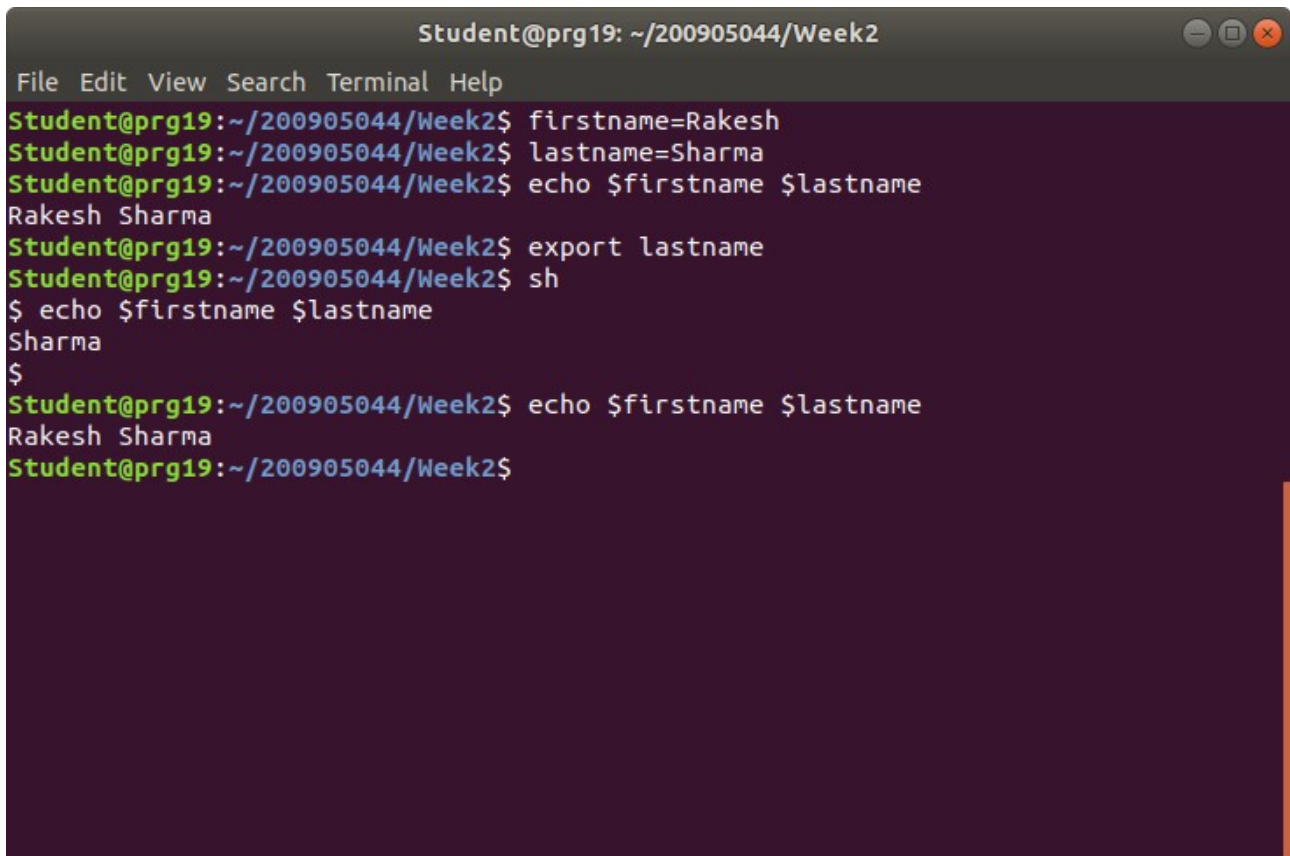
```
$ export lastname
```

```
$ sh
```

```
$ echo $firstname $lastname
```

\$ ^D

\$ echo \$firstname \$lastname

A terminal window titled "Student@prg19: ~/200905044/Week2" with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
Student@prg19:~/200905044/Week2$ firstname=Rakesh
Student@prg19:~/200905044/Week2$ lastname=Sharma
Student@prg19:~/200905044/Week2$ echo $firstname $lastname
Rakesh Sharma
Student@prg19:~/200905044/Week2$ export lastname
Student@prg19:~/200905044/Week2$ sh
$ echo $firstname $lastname
Sharma
$
Student@prg19:~/200905044/Week2$ echo $firstname $lastname
Rakesh Sharma
Student@prg19:~/200905044/Week2$
```

3) Try the following code snippet-

\$ cat > script.sh

echo the name of this script is \$0

echo the first argument is \$1

echo a list of all arguments is \$*

echo this script places the date into a temporary file

echo called \$1.\$\$

date > \$1.\$\$

ls \$1.\$\$

rm \$1.\$\$

^D

\$ chmod +x script.sh

\$./script.sh Rahul Sachin Kumble

```
Student@prg19: ~/200905044/Week2
File Edit View Search Terminal Help
Student@prg19:~/200905044/Week2$ cat > script.sh
echo the name of this script is $0
echo the first argument is $1
echo a list of all arguments is $*
echo this script places the date into a temporary file
echo called $1.$$
date > $1.$$
ls $1.$$
rm $1.$$
Student@prg19:~/200905044/Week2$ chmod +x script.sh
Student@prg19:~/200905044/Week2$ ./script.sh Rahul Sachin Kumble
the name of this script is ./script.sh
the first argument is Rahul
a list of all arguments is Rahul Sachin Kumble
this script places the date into a temporary file
called Rahul.6378
Rahul.6378
Student@prg19:~/200905044/Week2$
```

II)

1) Try the following:

\$ (sleep 10; echo done) &

\$ ps

```
Student@prg19: ~/200905044/Week2
File Edit View Search Terminal Help
Student@prg19:~/200905044/Week2$ (sleep 10; echo done) &
[1] 6437
Student@prg19:~/200905044/Week2$ ps
  PID TTY          TIME CMD
 3544 pts/0        00:00:00 sh
 3547 pts/0        00:00:00 bash
 6437 pts/0        00:00:00 bash
 6438 pts/0        00:00:00 sleep
 6439 pts/0        00:00:00 ps
Student@prg19:~/200905044/Week2$ done
ps
  PID TTY          TIME CMD
 3544 pts/0        00:00:00 sh
 3547 pts/0        00:00:00 bash
 6470 pts/0        00:00:00 ps
[1]+  Done                  ( sleep 10; echo done )
Student@prg19:~/200905044/Week2$
```

2) Try the following:

\$ (sleep 10; echo done) &

\$ kill pid

```
Student@prg19: ~/200905044/Week2
File Edit View Search Terminal Help
Student@prg19:~/200905044/Week2$ (sleep 10; echo done) &
[1] 6515
Student@prg19:~/200905044/Week2$ done
ps
  PID TTY          TIME CMD
 3544 pts/0        00:00:00 sh
 3547 pts/0        00:00:00 bash
 6517 pts/0        00:00:00 ps
[1]+  Done                  ( sleep 10; echo done )
Student@prg19:~/200905044/Week2$ kill 3547
Student@prg19:~/200905044/Week2$ kill 6517
bash: kill: (6517) - No such process
Student@prg19:~/200905044/Week2$
```

3) Try the following:

\$ (sleep 10; echo done 1) &

\$ (sleep 10; echo done 2) &

\$ echo done 3; wait; echo done 4

```
Student@prg19: ~/200905044/Week2
File Edit View Search Terminal Help
Student@prg19:~/200905044/Week2$ (sleep 10; echo done 1) &
[1] 6550
Student@prg19:~/200905044/Week2$ done 1
(sleep 10; echo done 2) &
[2] 6555
[1] Done ( sleep 10; echo done 1 )
Student@prg19:~/200905044/Week2$ done 2
echo done 3; wait; echo done 4
done 3
done 4
Student@prg19:~/200905044/Week2$
```

III) Write shell scripts for the following:

1) List all files under the given input directory, whose extension has only one character.

```
$ cat > w1q1.sh
```

```
echo Enter the directory name:
```

```
read d
```

```
cd $d && ls *.*
```

```
^D
```

```
$ chmod +x w1q1.sh
```

```
$ ./w1q1.sh
```



```
Student@prg19: ~/200905044/Week2
File Edit View Search Terminal Help
Student@prg19:~/200905044/Week2$ cat > w1q1.sh
echo Enter the directory name:
read d
cd $d && ls *.*
Student@prg19:~/200905044/Week2$ chmod +x w1q1.sh
Student@prg19:~/200905044/Week2$ mkdir newDir && cd newDir
Student@prg19:~/200905044/Week2/newDir$ touch a.txt b.v c.k d.css e.java
Student@prg19:~/200905044/Week2/newDir$ cd ..
Student@prg19:~/200905044/Week2$ ./w1q1.sh
Enter the directory name:
newDir
b.v c.k
Student@prg19:~/200905044/Week2$
```

2) Write a shell script that accepts 2 cmd line parameters. 1st parameter indicates the directory and the 2nd parameter indicates a regular expression. The script should display all files and directories in the directory specified in the 1st argument matching the format specified in the 2nd argument.

```
$ cat > w1q2.sh
```

```
cd $1 && ls $2
```

```
^D
```

```
$ chmod +x w1q2.sh
```

```
$ ./w1q2.sh newDir *.txt
```

```
Student@prg19: ~/200905044/Week2
File Edit View Search Terminal Help
Student@prg19:~/200905044/Week2$ cat > w1q2.sh
cd $1 && ls $2
Student@prg19:~/200905044/Week2$ chmod +x w1q2.sh
Student@prg19:~/200905044/Week2$ ./w1q2.sh newDir *.txt
a.txt hello.txt
Student@prg19:~/200905044/Week2$
```

3) Count the number of users logged on to the system. Display the output.

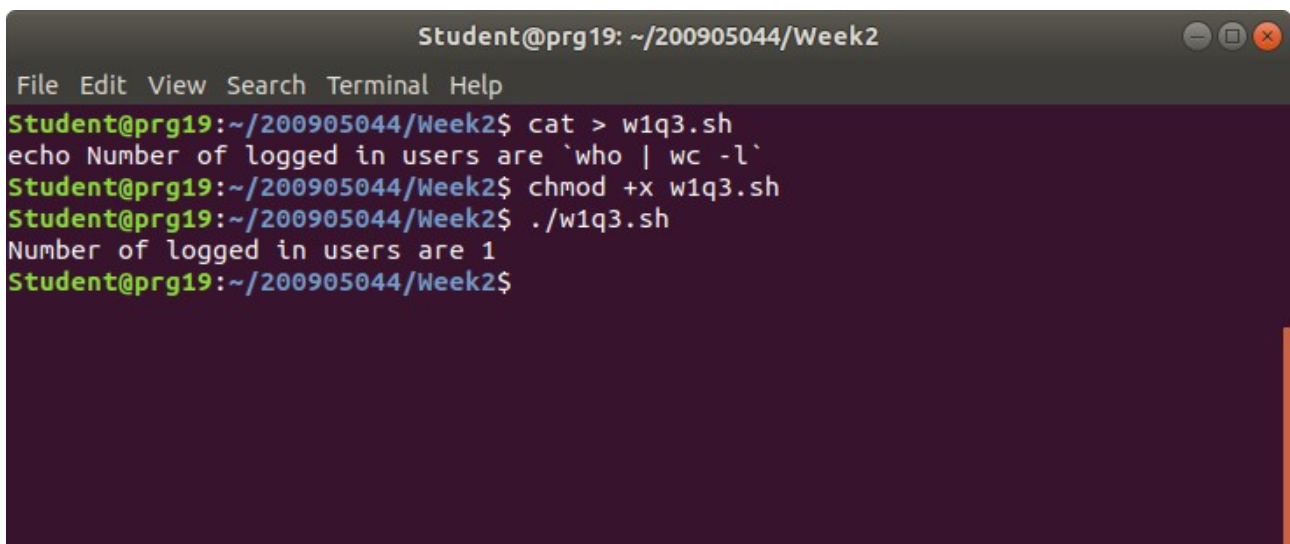
```
cat > w1q3.sh
```

```
echo Number of logged in users are `who | wc -l`
```

```
^D
```

```
$ chmod +x w1q3.sh
```

```
$ ./w1q3.sh
```

A terminal window titled "Student@prg19: ~/200905044/Week2" with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
Student@prg19:~/200905044/Week2$ cat > w1q3.sh
echo Number of logged in users are `who | wc -l`
Student@prg19:~/200905044/Week2$ chmod +x w1q3.sh
Student@prg19:~/200905044/Week2$ ./w1q3.sh
Number of logged in users are 1
Student@prg19:~/200905044/Week2$
```

4) Count only the number of files in the current directory

```
cat > w1q4.sh
```

```
echo Number of files only in this directory are `ls -al | grep ^- | wc -l`
```

```
^D
```

```
$ chmod +x w1q4.sh
```

```
$ ./w1q4.sh
```



```
Student@prg19: ~/200905044/Week2
File Edit View Search Terminal Help
Student@prg19:~/200905044/Week2$ cat > w1q4.sh
echo Number of files only in this directory are `ls -al | grep ^- | wc -l`
Student@prg19:~/200905044/Week2$ chmod +x w1q4.sh
Student@prg19:~/200905044/Week2$ ./w1q4.sh
Number of files only in this directory are 14
Student@prg19:~/200905044/Week2$
```

5) Write a shell script that takes 2 sorted numeric files as input and produces a single sorted numeric file without any duplicate contents.

```
cat > w1q5.sh
```

```
echo Enter name of 1st file:
```

```
read first
```

```
echo Enter name of 2nd file:
```

```
read second
```

```
sort -u $first $second > result.txt
```

```
cat result.txt
```

```
^D
```

```
$ chmod +x w1q5.sh
```

```
$ ./w1q5.sh a.txt b.txt
```

```
Student@prg19: ~/200905044/Week2
File Edit View Search Terminal Help

Student@prg19:~/200905044/Week2$ cat > w1q5.sh
echo Enter name of 1st file:
read first
echo Enter name of 2nd file:
read second
sort -u $first $second > result.txt
cat result.txt
Student@prg19:~/200905044/Week2$ chmod +x w1q5.sh
```

```
Student@prg19:~/200905044/Week2$ echo -e "1\n3\n5\n6\n" > a.txt
Student@prg19:~/200905044/Week2$ cat a.txt
1
3
5
6

Student@prg19:~/200905044/Week2$ echo -e "5\n6\n8\n10\n12\n" > b.txt
Student@prg19:~/200905044/Week2$ cat b.txt
5
6
8
10
12

Student@prg19:~/200905044/Week2$ ./w1q5.sh
Enter name of 1st file:
a.txt
Enter name of 2nd file:
b.txt

1
10
12
3
5
6
8
```

6) Write a shell script that accepts 2 cmd line args. 1st argument indicates format of file and 2nd indicates the destination directory. The script should copy all files as specified in the 1st argument to location indicated by the 2nd argument. Also try the script where the destination directory name has space in it.

```
cat > w1q6.sh
```

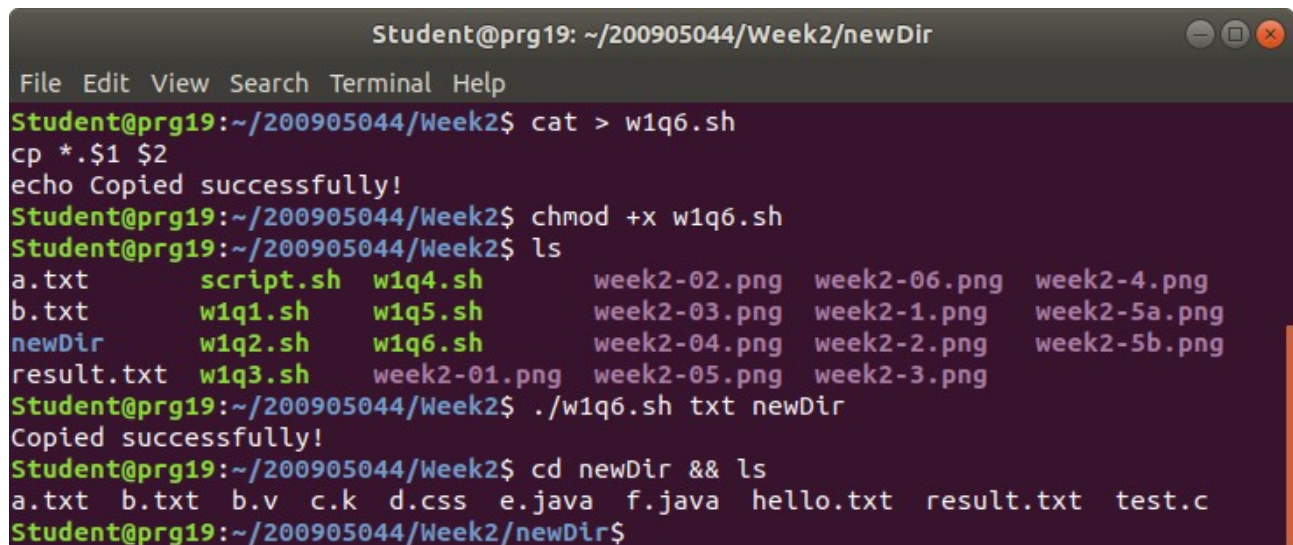
```
cp *.$1 $2
```

echo Copied successfully!

^D

\$ chmod +x w1q6.sh

\$./w1q6.sh txt newDir

A terminal window titled 'Student@prg19: ~/200905044/Week2/newDir' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
Student@prg19:~/200905044/Week2$ cat > w1q6.sh
cp *.$1 $2
echo Copied successfully!
Student@prg19:~/200905044/Week2$ chmod +x w1q6.sh
Student@prg19:~/200905044/Week2$ ls
a.txt      script.sh  w1q4.sh      week2-02.png week2-06.png week2-4.png
b.txt      w1q1.sh   w1q5.sh      week2-03.png week2-1.png   week2-5a.png
newDir     w1q2.sh   w1q6.sh      week2-04.png week2-2.png   week2-5b.png
result.txt w1q3.sh   week2-01.png week2-05.png week2-3.png
Student@prg19:~/200905044/Week2$ ./w1q6.sh txt newDir
Copied successfully!
Student@prg19:~/200905044/Week2$ cd newDir && ls
a.txt b.txt b.v c.k d.css e.java f.java hello.txt result.txt test.c
Student@prg19:~/200905044/Week2/newDir$
```

The command doesn't work if there is a space in Directory name as it treats it as another argument!

THANK YOU!