WEEK 2

Lab Exercise 01

1: Try the following shell commands:

$echo $HOME, $PATH
$echo $MAIL
$echo $USER, $SHELL, $PATH

```
student@dslab:~/main$ echo $HOME, $PATH
/home/student, /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
student@dslab:~/main$ echo $MAIL

student@dslab:~/main$ echo $USER, $SHELL, $TERM
student, /bin/sh, xterm-256color
student@dslab:~/main$
```

2: Try the following snippet, which illustrates the difference between local and environment variable.

```
$ firstname=Rakesh          ……local variables
$ lastname=Sharma
$ echo $firstname $lastname
$ export lastname           …..make "lastname" an envi var
$ sh                         …..start a child shell
$ echo $firstname $lastname
$ ^D                         …..terminate child shell
$ echo $firstname $lastname
```

```
student@dslab:~/main$ firstname=Rakesh
student@dslab:~/main$ lastname=Sharma
student@dslab:~/main$ echo $firstname $lastname
Rakesh Sharma
student@dslab:~/main$ export lastname
student@dslab:~/main$ sh
$ echo $firstname $lastname
Sharma
$
student@dslab:~/main$ echo $firstname $lastname
Rakesh Sharma
student@dslab:~/main$
```

3: Try the following snippet, which illustrates the meaning of special local variable.

```
$ cat >script.sh
 echo the name of this script is $0
 echo the first argument is $1
 echo a list of all the arguments is $*
 echo this script places the date into a temporary file
 echo called $1.$$
 date > $1.$$          # redirect the output of date
 ls $1.$$              # list the file
 rm $1.$$              # remove the file
 ^D
 $ chmod +x script.sh
 $ ./script.sh Rahul Sachin Kumble
```

```
student@dslab:~/main$ cat>script.sh
echo the name of this script is $0
echo the first argument is $1
echo a list of all the arguments is $*
echo this script places the date into a temporary file
echo called $1.$$
date>$1.$$
ls $1.$$
rm $1.$$
student@dslab:~/main$ chmod +x scrip.sh
student@dslab:~/main$ ./script.sh Rahul Sachin Kumble
the name of this script is ./script.sh
the first argument is Rahul
a list of all the arguments is Rahul Sachin Kumble
this script places the date into a temporary file
called Rahul.8289
Rahul.8289
student@dslab:~/main$
```

Lab Exercise 02

1: Try the following, which illustrates the usage of ps:

$(sleep 10; echo done)&

$ps

```
student@dslab:~/main$ (sleep 10; echo done 1)&
[1] 8667
student@dslab:~/main$ (sleep 10; echo done 2)&
[2] 8669
student@dslab:~/main$ ps
  PID TTY          TIME CMD
 8655 pts/0    00:00:00 sh
 8656 pts/0    00:00:00 bash
 8667 pts/0    00:00:00 bash
 8668 pts/0    00:00:00 sleep
 8669 pts/0    00:00:00 bash
 8670 pts/0    00:00:00 sleep
 8671 pts/0    00:00:00 ps
student@dslab:~/main$ done 1
```

2: Try the following which will illustrate the usage of kill:

$(sleep 10; echo done)&

$kill pid

```
student@dslab:~$ (sleep 10; echo done)&
[1] 8769
student@dslab:~$ ps
  PID TTY          TIME CMD
 8715 pts/0    00:00:00 sh
 8731 pts/0    00:00:00 bash
 8769 pts/0    00:00:00 bash
 8770 pts/0    00:00:00 sleep
 8771 pts/0    00:00:00 ps
student@dslab:~$ kill 8769done

bash: kill: (8769) - No such process
[1]+  Done                    ( sleep 10; echo done )
student@dslab:~$ ps
  PID TTY          TIME CMD
 8715 pts/0    00:00:00 sh
 8731 pts/0    00:00:00 bash
 8772 pts/0    00:00:00 ps
student@dslab:~$
```

Lab Exercise 03

Write shell scripts to do the following:

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

```
cd $1
ls *.?
```

```
student@dslab:~/main$ cat>week2_1.sh
cd $1
ls *.?
student@dslab:~/main$ chmod +x week2_1.sh
student@dslab:~/main$ ./week2_1.sh  /home/student/Desktop/
test.c  test.o
```

2: Write a shell script that accepts two command line parameters. First parameter indicates the directory and the second parameter indicates a regular expression. The script should display all the files and directories in the directory specified in the first argument matching the format specified in the second argument.

```
cd $1
ls | grep $2
```

```
student@dslab:~/main$ cat>week2_2.sh
cd $1
ls | grep $2
student@dslab:~/main$ chmod +x week2_2.sh
student@dslab:~/main$ ./week2_2.sh  ~/main/ a
a.txt
```

3: Count the number of users logged on to the system. Display the output as Number of users logged onto the system.

```
echo "The number of users logged in is: `who | wc -l`"
```

```
student@dslab:~/main$ cat>week2_3.sh
echo "The number of users logged in is: `who | wc -l`"
student@dslab:~/main$ chmod +x week2_3.sh
student@dslab:~/main$ ./week2_3.sh
The number of users logged in is: 1
```

4: Count only the number of files in the current directory.

```
echo "The number of files in the current directory is: `ls -l | grep "^-" | wc -l`"
```

```
student@dslab:~/main$ cat>week2_4.sh
echo "The number of files in the current directory is: `ls -l | grep "^-" | wc
-l`"
student@dslab:~/main$ chmod +x week2_4.sh
student@dslab:~/main$ ./week2_4.sh
The number of files in the current directory is: 7
student@dslab:~/main$ ls
a.txt  hello.b  sub_2  week2_1.sh  week2_3.sh
A.txt  sub_1    test   week2_2.sh  week2_4.sh
```

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

```
cat $1 > merge
cat $2 >> merge
sort -n merge | uniq > sorted_merged
cat sorted_merged
```

```
student@dslab:~/main$ cat>nfile1.txt
1
2
3
4
student@dslab:~/main$ cat>nfile2.txt
1
2
3
student@dslab:~/main$ cat>week2_5.sh
cat $1 > merge
cat $2 >> merge
sort -n merge | uniq > sorted_merged
cat sorted_merged
student@dslab:~/main$ chmod +x week2_5.sh
student@dslab:~/main$ ./week2_5.sh  nfile1.txt nfile2.txt
1
2
3
4
```

6: Write a shell script that accepts two command line arguments. First argument indicates format of the file and the second argument indicates the destination directory. The script should copy all the files as specified in the first argument to the location indicated by the second argument. Also, try script where the destination directory name has space in it.

```
cp `ls *$1` $2
```

```
student@dslab:~/main$ cat>week2_6.sh
cp `ls *$1` $2
student@dslab:~/main$ chmod +x week2_6.sh
student@dslab:~/main$ ./week2_5.sh  .sh ./test
cat: .sh: No such file or directory
./week2_5.sh: line 5: warning: here-document at line 2 delimited by end-of-file (wanted `merge')
cat: ./test: Is a directory
student@dslab:~/main$ ./week2_6.sh  .sh ./test
student@dslab:~/main$ cd test
student@dslab:~/main/test$ ls
week2_1.sh  week2_2.sh  week2_3.sh  week2_4.sh  week2_5.sh  week2_6.sh
```