Table of Contents

Introduction

This is the documentation of a script file containing various types of fundamental UNIX commands have been utilized to perform the assigned tasks. It is about utilizing the basic commands to work with files and directories from terminal or shell. The interaction of read/write with files have been made in the coursework. Also, there is engagement with access permissions and other essential UNIX commands.

The objectives of this basic task are listed below:

- To learn how to work in the OS via terminal also without being dependent on graphical interface only.
- To be able to use the commands that we learned in regular classes.
- To be familiar with various UNIX based commands.

Transcript

```
Script started on 2021-03-19 15:21:44+05:45 [TERM="xterm-256color"
TTY="/dev/tty1" COLUMNS="120" LINES="30"]
```

1. Creating New Directories

#Task1

Command: Creating the NBA directory

```
/home/pexmu$mkdir -p NBA/{Lakers,Miami,"Chicago Bulls","Brooklyn
nets","Dallas Maverick"}
```

Command: Showing the NBA directory structure

```
/home/pexmu$tree NBA # showing structure of NBA
```

Response:

```
NBA

├── Brooklyn nets

├── Chicago Bulls

├── Dallas Maverick

├── Lakers

└── Miami



5 directories, 0 files
```

2. Removing Existing Files and directories

#Task2

Command: Changing to Miami directory using relative pathname

```
/home/pexmu$cd NBA/Miami
```

Command: Showing that we are in Miami directory

```
/home/pexmu/NBA/Miami$pwd
```

Response:

```
/home/pexmu/NBA/Miami
```

Command: Creating two files in Miami directory

```
/home/pexmu/NBA/Miami$cat > weather

It is very sunny right now.^Z

[1]+  Stopped                 cat > weather

/home/pexmu/NBA/Miami$cat > time

It is 3:25PM right now.^Z

[2]+  Stopped                 cat > time
```

Command: Showing that two files exist

```
/home/pexmu/NBA/Miami$ls #showing files created above exists in Miami

time  weather
```

#Task3

Command: Changing to the NBA directory

```
/home/pexmu/NBA/Miami$cd ../
```

Command: Removing previous two files of Miami directory with interactive mode

```
/home/pexmu/NBA$rm -i Miami/{weather,time}
```

Response:

```
rm: remove regular empty file 'Miami/weather'? y

rm: remove regular empty file 'Miami/time'? y
```

Command: Showing absence of the two files (time, weather)

```
/home/pexmu/NBA$ls -al Miami #showing absense of two files(weather,time)
```

Response:

```
total 0

drwxr-xr-x 1 pexmu pexmu 512 Mar 19 15:26 .

drwxr-xr-x 1 pexmu pexmu 512 Mar 19 15:23 ..
```

Command: Removing/deleting Miami directory

```
/home/pexmu/NBA$rmdir Miami #deleting Miami directory
```

Command: Showing absence of Miami directory

```
/home/pexmu/NBA$ls # showing Miami doesnot exist now
```

Response:

```
'Brooklyn nets'  'Chicago Bulls'  'Dallas Maverick'   Lakers
```

3. Usage of the echo commands

#Task4

Command: Printing the following strings in one echo command

```
/home/pexmu/NBA$echo -e "Hello! I am big fan of NBA finals.\n14<(2+2)"
```

Response:

```
Hello! I am big fan of NBA finals.

14<(2+2)
```

Command: Giving the pwd command

```
/home/pexmu/NBA$pwd
```

Response:

```
/home/pexmu/NBA
```

Command: Changing to Lakers directory

```
/home/pexmu/NBA$cd Lakers
```

Command: Showing that we are in Lakers directory

```
/home/pexmu/NBA/Lakers$pwd
```

Response:

```
/home/pexmu/NBA/Lakers
```

#Task5

Command: Give the group of following commands

```
/home/pexmu/NBA/Lakers$pwd; cd; pwd
```

Response:

```
/home/pexmu/NBA/Lakers

/home/pexmu
```

Explanation:

```
/home/pexmu$echo "The above group command first showed the present
directory which was Lakers

> Then cd; took us to home dir, the next pwd verifies that we are in home
dir now"

The above group command first showed the present directory which was
Lakers

Then cd; took us to home dir, the next pwd verifies that we are in home
dir now
```

#Task6

Command: Changing to the Lakers directory

```
/home/pexmu$cd NBA/Lakers
```

Command: Give the group of following command

```
/home/pexmu/NBA/Lakers$pwd; cd ..; pwd; cd ..; pwd
```

Response:

```
/home/pexmu/NBA/Lakers

/home/pexmu/NBA

/home/pexmu
```

Explanation:

```
/home/pexmu$echo "in the above command we saw we are in Lakers dir

> then go one directory back by (cd ..;) command then saw we are in NBA
dir

> then we return to home directory then saw home direction path by pwd
command"

in the above command we saw we are in Lakers dir

then go one directory back by (cd ..;) command then saw we are in NBA dir

then we return to home directory then saw home direction path by pwd
command
```

4. Usage of the ls command

#Task7

Command and Response:

```
/home/pexmu$cd; pwd

/home/pexmu
```

Explanation:

```
/home/pexmu$echo "the 'cd;' took us to home dir

> then pwd verifes that we are in home dir by showing path"

the 'cd;' took us to home dir

then pwd verifes that we are in home dir by showing path
```

Command and Response:

```
/home/pexmu$ls

19031371cw2  NBA
```

Explanation:

```
/home/pexmu$echo "(ls) command lists the files and dir present in the
current dir

> excluding hidden items"

(ls) command lists the files and dir present in the current dir

excluding hidden items
```

Command and Response:

```
/home/pexmu$ls -a

.  ..  19031371cw2  .bash_history  .bash_logout  .bashrc  .local  NBA
.profile
```

Explanation:

```
/home/pexmu$echo "(ls -a) command shows all the file/dir names

> present in the current dir including the hidden items."

(ls -a) command shows all the file/dir names

present in the current dir including the hidden items.
```

Command and Response:

```
/home/pexmu$ls -al

total 28

drwxr-xr-x 1 pexmu pexmu  512 Mar 19 15:23 .

drwxr-xr-x 1 root  root   512 Mar 17 14:02 ..

-rw-r--r-- 1 pexmu pexmu 5120 Mar 19 16:23 19031371cw2

-rw------- 1 pexmu pexmu 1697 Mar 19 15:50 .bash_history

-rw-r--r-- 1 pexmu pexmu  220 Mar 17 14:02 .bash_logout

-rw-r--r-- 1 pexmu pexmu 3565 Mar 19 08:26 .bashrc

drwxr-xr-x 1 pexmu pexmu  512 Mar 18 19:29 .local

drwxr-xr-x 1 pexmu pexmu  512 Mar 19 15:28 NBA

-rw-r--r-- 1 pexmu pexmu  807 Mar 17 14:02 .profile
```

Explanation:

```
/home/pexmu$echo "(ls -al) command shows information like

> file/dir name, access permision,modified date/time including hidden
items."

(ls -al) command shows information like

file/dir name, access permision,modified date/time including hidden items.
```

#Task8

Command and Response:

```
/home/pexmu/NBA$cd; pwd; cd cw2; pwd

/home/pexmu

bash: cd: cw2: No such file or directory

/home/pexmu
```

Explanation:

```
/home/pexmu$# in above group of command, we went to home dir by cd;

/home/pexmu$# then pwd; shows the path that we are in home dir

/home/pexmu$# then cd cw2 tries to get inside cw2 dir, as cw2 is not
present: we get error message

/home/pexmu$#then pwd showed that we are in home dir
```

Command and Response:

```
/home/pexmu$ls -R

.:

19031371cw2  NBA


./NBA:

'Brooklyn nets'  'Chicago Bulls'  'Dallas Maverick'   Lakers


'./NBA/Brooklyn nets':


'./NBA/Chicago Bulls':


'./NBA/Dallas Maverick':


./NBA/Lakers:
```

Explanation:

```
/home/pexmu$ #(ls -R) command lists the files/dir present in current dir
recursivly entering inside available directories
```

5. Usage of the cat command

```
/home/pexmu$cd NBA/'Chicago Bulls'
```

#Task9

Command: Creating three files using cat utility

```
/home/pexmu/NBA/Chicago Bulls$cat > testX
aaabb Aaaaa
AAAAA
bbbcc Bbbbb
BBBBB
ff-ff Ccccc
CCCCC
cccdd Ddddd
DDDDD^Z
[3]+  Stopped                 cat > testX
/home/pexmu/NBA/Chicago Bulls$cat > testY
aaabb Aaaaa
AAAAA
bbbcc Bbbbb
BBBBB
ff-ff Ccccc
CCCCC
cccdd Ddddd
DDDDD^Z
[4]+  Stopped                 cat > testY
/home/pexmu/NBA/Chicago Bulls$cat > testZ
```

```
aaabb Aaaaa

AAAAA

bbbcc Bbbbb

BBBBB

ff-ff Ccccc

CCCCC

cccdd Ddddd

DDDDD^Z

[5]+  Stopped                 cat > testZ
```

Command and Response: Showing three files exist

```
/home/pexmu/NBA/Chicago Bulls$ls

testX  testY  testZ
```

#Task10

Command: Displaying testX

```
/home/pexmu/NBA/Chicago Bulls$cat testX
```

Response:

```
aaabb Aaaaa

AAAAA

bbbcc Bbbbb

BBBBB

ff-ff Ccccc

CCCCC

cccdd Ddddd
```

Command: Displaying testY

```
/home/pexmu/NBA/Chicago Bulls$cat testY
```

Response:

```
aaabb Aaaaa

AAAAA

bbbcc Bbbbb

BBBBB

ff-ff Ccccc

CCCCC

cccdd Ddddd
```

Command: Displaying testZ

```
/home/pexmu/NBA/Chicago Bulls$cat testZ
```

Response:

```
aaabb Aaaaa

AAAAA

bbbcc Bbbbb

BBBBB

ff-ff Ccccc

CCCCC

cccdd Ddddd
```

#Task11

Command: Copying the files to Brooklyn nets directory

```
/home/pexmu/NBA/Chicago Bulls$cp testX testY testZ ../"Brooklyn nets" #cp command copies the asked items and (../"Brooklyn nets") is the path to paste our item
```

Command: Showing that the files are now available in Brooklyn nets directory

```
/home/pexmu/NBA/Chicago Bulls$ls ../"Brooklyn nets"

testX  testY  testZ
```

#Task12

Command: Concatenate the files using given command

```
/home/pexmu/NBA/Chicago Bulls$cat testX testY testZ #reading all three files once
```

Response:

```
aaabb Aaaaa

AAAAA

bbbcc Bbbbb

BBBBB

ff-ff Ccccc

CCCCC

cccdd Ddddd

aaabb Aaaaa

AAAAA

bbbcc Bbbbb

BBBBB

ff-ff Ccccc

CCCCC
```

```
cccdd Ddddd

aaabb Aaaaa

AAAAA

bbbcc Bbbbb

BBBBB

ff-ff Ccccc

CCCCC

cccdd Ddddd
```

Command: Concatenate the files using given command

```
/home/pexmu/NBA/Chicago Bulls$cat testX testY testZ > testResult

/home/pexmu/NBA/Chicago Bulls$echo "above cat command adds all the left
side files

> and overwrites to right side file which is testResult"

above cat command adds all the left side files

and overwrites to right side file which is testResult
```

Command: Displaying testResult using cat utility

```
/home/pexmu/NBA/Chicago Bulls$cat testResult #displaying testResult file
by cat
```

Response:

```
aaabb Aaaaa

AAAAA

bbbcc Bbbbb

BBBBB

ff-ff Ccccc

CCCCC

cccdd Ddddd

aaabb Aaaaa

AAAAA

bbbcc Bbbbb

BBBBB

ff-ff Ccccc

CCCCC

cccdd Ddddd

aaabb Aaaaa

AAAAA

bbbcc Bbbbb

BBBBB

ff-ff Ccccc

CCCCC

cccdd Ddddd
```

#Task13

Command: Give the following command

```
/home/pexmu/NBA/Chicago Bulls$cat test[XYZ]
```

Response:

```
aaabb Aaaaa

AAAAA

bbbcc Bbbbb

BBBBB

ff-ff Ccccc

CCCCC

cccdd Ddddd

aaabb Aaaaa

AAAAA

bbbcc Bbbbb

BBBBB

ff-ff Ccccc

CCCCC

cccdd Ddddd

aaabb Aaaaa

AAAAA

bbbcc Bbbbb

BBBBB

ff-ff Ccccc

CCCCC

cccdd Ddddd
```

Explanation:

```
/home/pexmu/NBA/Chicago Bulls$echo "the value inside square bracket is
optional

> in unix, cat test[XYZ] tries to cat testX,testY,testZ respectivly

> the square bracket tool can be useful where multiple items have same

> initial filename"

the value inside square bracket is optional

in unix, cat test[XYZ] tries to cat testX,testY,testZ respectivly

the square bracket tool can be useful where multiple items have same

initial filename
```

6. Usage of the chmod command

#Task14

Command: Displaying all access permissions for files in Chicago Bulls

```
/home/pexmu/NBA/Chicago Bulls$ls -l #displaying access permission of items present in current dir
```

Response:

```
total 0

-rw-r--r-- 1 pexmu pexmu 198 Mar 19 17:13 testResult

-rw-r--r-- 1 pexmu pexmu  66 Mar 19 16:44 testX

-rw-r--r-- 1 pexmu pexmu  66 Mar 19 16:45 testY

-rw-r--r-- 1 pexmu pexmu  66 Mar 19 16:46 testZ
```

Command: Removing all access permissions for the testX file

```
/home/pexmu/NBA/Chicago Bulls$chmod 000 testX #removing access permision for testX file
```

Command and Response: Displaying access permissions for the testX file

```
/home/pexmu/NBA/Chicago Bulls$ls -l testX

---------- 1 pexmu pexmu 66 Mar 19 16:44 testX
```

Command: Trying to read testX file

```
/home/pexmu/NBA/Chicago Bulls$cat testX #trying to read file after removing permision
```

Response:

```
cat: testX: Permission denied
```

Command and Response: Trying to write testX file

```
/home/pexmu/NBA/Chicago Bulls$cat > testX #trying to write testX with no permissions

bash: testX: Permission denied
```

Command: Adding read and write permissions for testX file

```
/home/pexmu/NBA/Chicago Bulls$chmod 600 testX #adding read/write permision
for User
```

Command and Response: Displaying access permissions for testX file

```
/home/pexmu/NBA/Chicago Bulls$ls -l testX

-rw------- 1 pexmu pexmu 66 Mar 19 16:44 testX
```

Command and Response: Trying to read testX file

```
/home/pexmu/NBA/Chicago Bulls$cat testX #reading after giving permission

aaabb Aaaaa

AAAAA

bbbcc Bbbbb

BBBBB

ff-ff Ccccc

CCCCC

cccdd Ddddd
```

Command and Response: Trying to write testX file

```
/home/pexmu/NBA/Chicago Bulls$nano testX #trying to write after giving
write permission

/home/pexmu/NBA/Chicago Bulls$#was able to write from the above command
```

#Task15

```
/home/pexmu/NBA/Chicago Bulls$cd ../
```

Command: Displaying access permissions for Chicago Bulls

```
/home/pexmu/NBA$ls -l 'Chicago Bulls' #displaying access permission of
Chicago Bulls dir
```

Response:

```
total 0

-rw-r--r-- 1 pexmu pexmu 198 Mar 19 17:13 testResult

-rw------- 1 pexmu pexmu  66 Mar 19 17:49 testX

-rw-r--r-- 1 pexmu pexmu  66 Mar 19 16:45 testY

-rw-r--r-- 1 pexmu pexmu  66 Mar 19 16:46 testZ
```

Command: Removing all access permissions for Chicago Bulls

```
/home/pexmu/NBA$chmod 000 'Chicago Bulls'
```

Command: Displaying access permissions for Chicago Bulls

```
/home/pexmu/NBA$ls -l #checking access permission of Chicago Bulls
```

Response:

```
total 0

drwxr-xr-x 1 pexmu pexmu 512 Mar 19 17:07 'Brooklyn nets'

d--------- 1 pexmu pexmu 512 Mar 19 17:49 'Chicago Bulls'

drwxr-xr-x 1 pexmu pexmu 512 Mar 19 15:23 'Dallas Maverick'

drwxr-xr-x 1 pexmu pexmu 512 Mar 19 15:23  Lakers
```

Command: Trying to read a file from Chicago Bulls

```
/home/pexmu/NBA$cat 'Chicago Bulls'/testX #trying to read testX
```

Response:

```
cat: 'Chicago Bulls/testX': Permission denied
```

Command and Response: Trying to put a file into Chicago Bulls

```
/home/pexmu/NBA$nano 'Chicago Bulls'/sample.txt #trying to create new .txt
file

Use "fg" to return to nano.



[6]+  Stopped                 nano 'Chicago Bulls'/sample.txt

/home/pexmu/NBA$# not able to write due to no access permission in that
directory
```

Command: Trying to search in Chicago Bulls using ls utility

```
/home/pexmu/NBA$ls -l 'Chicago Bulls' #trying to search in this directory
```

Response:

```
ls: cannot open directory 'Chicago Bulls': Permission denied
```

Command: Adding read/write/execute permission to Owner for Chicago Bulls

```
/home/pexmu/NBA$chmod 700 'Chicago Bulls' #giving full permsion to User
```

Command: Displaying access permissions for Chicago Bulls

```
/home/pexmu/NBA$ls -l
```

Response:

```
total 0

drwxr-xr-x 1 pexmu pexmu 512 Mar 19 17:07 'Brooklyn nets'

drwx------ 1 pexmu pexmu 512 Mar 19 17:49 'Chicago Bulls'
```

```
drwxr-xr-x 1 pexmu pexmu 512 Mar 19 15:23 'Dallas Maverick'

drwxr-xr-x 1 pexmu pexmu 512 Mar 19 15:23  Lakers

/home/pexmu/NBA$#permision can be seen for the Chicago Bulls dir in above
```

Command: Trying to read a file giving permission

```
/home/pexmu/NBA$cat 'Chicago Bulls'/testX #trying to read after granting
permision
```

Response:

```
aaabb Aaaaa

AAAAA

bbbcc Bbbbb

BBBBB

ff-ff Ccccc

CCCCC

cccdd Ddddd
```

Command: Trying to put a file into Chicago Bulls

```
/home/pexmu/NBA$touch 'Chicago Bulls'/randomFile.txt #adding a file using
touch command
```

Command: Trying to search in Chicago Bulls

```
/home/pexmu/NBA$ls 'Chicago Bulls' #showing files inside Chicago Bulls
directory
```

Response:

```
randomFile.txt  testResult  testX  testY  testZ
```

7. Usage of the grep command

```
/home/pexmu/NBA$cd 'Brooklyn nets'
```

#Task16

Command: Give the following command

```
/home/pexmu/NBA/Brooklyn nets$grep bb testX
```

Response:

```
aaabb Aaaaa

bbbcc Bbbbb
```

Explanation:

```
/home/pexmu/NBA/Brooklyn nets$# above grep command shows those lines in
the file which contains bb
```

Command:

```
/home/pexmu/NBA/Brooklyn nets$grep -v bb testX
```

Response:

```
AAAAA

BBBBB

ff-ff Ccccc

CCCCC

cccdd Ddddd
```

Explanation:

```
/home/pexmu/NBA/Brooklyn nets$# those lines that donot contain (bb) in the
file are shown by above command
```

Command:

```
/home/pexmu/NBA/Brooklyn nets$grep -n bb testX
```

Response:

```
1:aaabb Aaaaa

3:bbbcc Bbbbb
```

Explanation:

```
/home/pexmu/NBA/Brooklyn nets$# -n in grep shows those lines containing bb
along with the line number as shown above
```

Command:

```
/home/pexmu/NBA/Brooklyn nets$grep -l bb *
```

Response:

```
testX

testY

testZ
```

Explanation:

```
/home/pexmu/NBA/Brooklyn nets$# (-l bb *) in grep command is referring to
grep those files which contains at least one time (bb) in the file. (*) is
referring to all files here. Those files having at least a single (bb)
inside the file are shown by the above command.
```

Command:

```
/home/pexmu/NBA/Brooklyn nets$grep -i bb *
```

Response:

```
testX:aaabb Aaaaa

testX:bbbcc Bbbbb

testX:BBBBB

testY:aaabb Aaaaa

testY:bbbcc Bbbbb

testY:BBBBB

testZ:aaabb Aaaaa

testZ:bbbcc Bbbbb

testZ:BBBBB
```

Explanation:

```
/home/pexmu/NBA/Brooklyn nets$# (-i) is meant to perform case sensitive
search. The capital or small casing does not matter, The above command
picks those lines from all files containing (bb-ignoring casing)
```

Command:

```
/home/pexmu/NBA/Brooklyn nets$grep -i BB *
```

Response:

```
testX:aaabb Aaaaa

testX:bbbcc Bbbbb

testX:BBBBB

testY:aaabb Aaaaa

testY:bbbcc Bbbbb

testY:BBBBB

testZ:aaabb Aaaaa

testZ:bbbcc Bbbbb

testZ:BBBBB
```

Explanation:

```
/home/pexmu/NBA/Brooklyn nets$# this is similar to previous command, only
the bb is replaced by BB. This command will also give same output because
the casing of the letters are ignored by (-i)
```

Command:

```
/home/pexmu/NBA/Brooklyn nets$grep -c bb *
```

Response:

```
testX:2

testY:2

testZ:2
```

Explanation:

```
/home/pexmu/NBA/Brooklyn nets$# (-c) in grep command counts the number of
times that our pattern is repeated. In this case, we are counting how many
times (bb) is present in each file.
```

Command:

```
/home/pexmu/NBA/Brooklyn nets$grep '^A' *
```

Response:

```
testX:AAAAA

testY:AAAAA

testZ:AAAAA
```

Explanation:

```
/home/pexmu/NBA/Brooklyn nets$# the ^ symbol is used to grip lines that
starts with provided pattern. In our case, we are greping those lines that
start with (A)
```

Command:

```
/home/pexmu/NBA/Brooklyn nets$grep -n '^' testX
```

Response:

```
1:aaabb Aaaaa

2:AAAAA

3:bbbcc Bbbbb

4:BBBBB

5:ff-ff Ccccc

6:CCCCC

7:cccdd Ddddd
```

Explanation:

```
/home/pexmu/NBA/Brooklyn nets$# above command shows all the lines in the
file along with line number from testX file
```

8. Aliasing

#Task17

Command: Defining two aliases command

```
/home/pexmu$cat > our_alias

alias lsal="ls -al"

alias lsa="ls -a"

^Z

[7]+  Stopped                 cat > our_alias

/home/pexmu$# above two aliases are added in our_alias file
```

Command: Adding our aliases to environmental file

```
/home/pexmu$cat our_alias >> .bashrc #appending our alliases in system
permanently
```

Command: Showing that our aliases are present in .bashrc file

```
/home/pexmu$grep '^alias' .bashrc #showing that our own aliases are
stored.
```

Response:

```
alias lsal="ls -al"

alias lsa="ls -a"
```

Command: Using our alias(lsal) from home directory

```
/home/pexmu$ lsal #using this alias from home directory
```

Response:

```
total 48
drwxr-xr-x 1 pexmu pexmu   512 Mar 19 19:10 .
drwxr-xr-x 1 root  root    512 Mar 17 14:02 ..
-rw-r--r-- 1 pexmu pexmu 23040 Mar 19 19:13 19031371cw2
-rw------- 1 pexmu pexmu  1697 Mar 19 15:50 .bash_history
-rw-r--r-- 1 pexmu pexmu   220 Mar 17 14:02 .bash_logout
-rw-r--r-- 1 pexmu pexmu  3603 Mar 19 19:13 .bashrc
drwxr-xr-x 1 pexmu pexmu   512 Mar 18 19:29 .local
drwxr-xr-x 1 pexmu pexmu   512 Mar 19 15:28 NBA
-rw-r--r-- 1 pexmu pexmu    38 Mar 19 19:11 our_alias
-rw-r--r-- 1 pexmu pexmu   807 Mar 17 14:02 .profile
-rw-r--r-- 1 pexmu pexmu  1024 Mar 19 17:46 .ram.swp
```

Command: Using our alias(lsa) from home directory

```
/home/pexmu$ lsa #using lsa from home
```

Response:

```
.    19031371cw2     .bash_logout   .local   our_alias   .ram.swp

..   .bash_history   .bashrc        NBA      .profile
```

#Task18

Command: Opening .bashrc file and removing our previous two aliases

```
/home/pexmu$ nano .bashrc #opening the .bashrc file and removing our
aliases
```

Command: Unaliasing our two previous aliases

```
/home/pexmu$ unalias lsal

/home/pexmu$ unalias lsa
```

Command and Response: Showing that system does not store these aliases

```
/home/pexmu$ lsal #now not working

bash: lsal: command not found

/home/pexmu$ lsa #now not working

bash: lsa: command not found
```

#Task19

Command: Defining the previous aliases again to system

```
/home/pexmu$ cat our_alias >> .bashrc #adding previous script again
```

Command: Stopping the session by exiting and login in to system again

```
/home/pexmu$ exit

exit

There are stopped jobs.

Home/pexmu$ exit

exit



Script done on 2021-03-19 20:02:57+05:45 [COMMAND_EXIT_CODE="1"]
```

Command: Showing that our alias still works on new session

```
/home/pexmu$ lsa #using lsa from home in new session
```

Response:

```
.   19031371cw2    .bash_logout   .local   our_alias   .ram.swp

..  .bash_history  .bashrc        NBA      .profile
```

#Task20

Command: Creating ‘noAllf’ command that counts number of files including hidden ones in working directory

```
/home/pexmu$ cat > task20 #lets make command for task20 here

alias noAllf="ls -al | grep '^-' | wc -l"

^Z

[1]+  Stopped                 cat > task20
```

Explanation:

Task20 is about listing number of files including hidden one’s in present directory. For that we made a command where we list all items by (ls -al). Then by using pipe ‘|’ we passed the result to another command which is (grep ‘^-’), here we selected those items whose starting is ‘-’. The starting of directory is ‘d’ while starting of file is ‘-’. So, after selecting files we send it to another command using pipe ‘|’ again. Here (wc -l) means to count the present items. In this way, we solved this task.

Command: Putting the noAllf alias into environment variable

```
/home/pexmu$ cat task20 >> .bashrc #appending our task20 alias into
environment variable
```

#Task21

Command: Creating ‘noAsubsir’ alias that counts recursively all sub-dirs in working directory

```
/home/pexmu$ cat > task21 #lets make command for task21 here

alias noAsubsir="ls -RAl | grep '^d' | wc  -l"

^Z

[2]+  Stopped                 cat > task21
```

Explanation:

Task21 is about counting all the sub-directories that encounter in the current directory recursively. For this we made a group of commands where we list all the items including directories and the contents inside each available sub-directory recursively by (ls -RAl). Now we need only directories item. To get that, we passed previous result to another command by pipe. The command (grep ‘^d’) takes all the items which start with ‘d’. As directories have ‘d’, we are able to get those. Now as per task, to count those directories item we take the previous result to another command by pipe. Now (wc -l) command counts the available directory items. In this way, we solved this task.

Command: Putting noAsubsir alias into environment variable

```
/home/pexmu$ cat task21 >> .bashrc #adding task21 alias also to
environment variable
```

#Task22

Command: Creating ‘noAcs’ alias that counts files starting with provided alphabets and also checks inside internal sub-directory recursively.

```
/home/pexmu$ cat > task22 #lets make command for task 22 here

alias noAcs="ls -la | grep "^-" | ls -R | grep -E "^g|^t|^w"  | wc -l"

^Z

[3]+  Stopped                 cat > task22
```

Explanation:

Task22 is about displaying the number of files whose name starts with ‘g’ or ‘t’ or ‘w’ in the current directory and inside sub-directory recursively. For this, we made a group of commands where (ls -la) lists down all the files/directories in the present directory including hidden items. Then the result is passed to another following command with pipe ‘|’. The (grep ‘^-’) selects only files because file starts with ‘-’. The next (ls -R) provides details of each internal sub-directories also. The (grep -E ‘^g|^t|^w’) selects those files whose name starts with ‘g’ or ‘t’ or ‘w’. Then the result is forwarded to next command by piping where the counting is done. In this way, we solved this task.

Command: Putting noAcs alias into environment variable

```
/home/pexmu$ cat task22 >> .bashrc #adding task22 alias also to env
variable
```

9. Usage of own commands

#Task23

Command: Using noAllf alias

```
/home/pexmu$ noAllf
```

Response:

```
15
```

#Task24

Command: Using noAsubsir alias

```
/home/pexmu$ noAsubsir
```

Response:

```
8
```

#Task25

Command: Using noAcs alias

```
/home/pexmu$ noAcs
```

Response:

```
9
```

10. Command history

#Task26

Command: Listing last six commands executed

```
/home/pexmu$ history 6
```

Response:

```
  607  #task20,21,22 explanation is in report as it covers long
explanation

  608  . .bashrc #loading file that contain our aliases

  609  noAllf #running alias as per task23

  610  .noAsubsir #running alias as per task24

  611  noAcs #running alias as per task25

  612  history 6
```

#Task27

Command: Re-executing the command given eight commands ago

```
/home/pexmu$ !-8  #reexecuting the past 8th command
```

Response:

```
ls

19031371cw2  apple  NBA  our_alias  task20  task21  task22  try
```

#Task28

Command: Re-executing the last command which name begins with ‘m’

```
/home/pexmu$ !m #reexecuting  last command that begin with m
```

Response:

```
mkdir apple #reexecuting  last command that begin with m

mkdir: cannot create directory ‘apple’: File exists
```

Conclusion

The tasks (i.e., 1-28) are completed successfully and are recorded in the script file. In addition, some rubbish is also cleaned before finalizing the report file. Reaching this part of the task immediately boosted my confidence on solving general works that can be achieved from UNIX commands. The report of this task is about documenting the works done in UNIX shell.

This assignment came to be very helpful for broadening my skills on UNIX based operating system. The limited habit of getting dependent on graphical interface only shifted to be more aware about the operating system, types of shell and similar command-based problem-solving aspects. While carrying out the task, there was need to search about UNIX documentation. On that phase, the resulted web articles were very effective to explore other unheard features and unique properties of shell command line options.

With the observant guidance and assist from our respected tutors, smooth completion of this assignment was possible. The revision of the subject matters from previous lectures and tutorial classes were also very useful.

References

geeksforgeeks, 2019. *expr command in Linux with examples.* [Online]
Available at: https://www.geeksforgeeks.org/expr-command-in-linux-with-examples/

Massachusetts Institute of Technology, 1994. *Bash Features - Bash Specific Features.* [Online]
Available at: http://web.mit.edu/gnu/doc/html/features_4.html

stanford.edu, 1997. *Basic UNIX commands.* [Online]
Available at: http://mally.stanford.edu/~sr/computing/basic-unix.html