

Linux Assignment 1 - DAC(sep-2022)

1. Practice the below-mentioned commands with all the possible options:
cd, cat, ls, mkdir, rmdir, pwd, mv, cp, rm.

```
nischal at anton in ~ cat, ls, mkdir, rmdir, pwd, mv, cp, rm
$ cd Demo/
nischal at anton in ~/Demo cd do when used without arguments?
$ ls
a.out dir helloworld.c myScript.sh q2 q2.c q3 q3.c q4 q4.c q5 q5.c test text
nischal at anton in ~/Demo
$ ls -al
total 116
drwxrwxr-x 3 nischal nischal 4096 Sep 25 00:49 .
drwxr-xr-x 56 nischal nischal 4096 Sep 25 00:49 ..
-rwxrwxr-x 1 nischal nischal 16120 Sep 24 22:38 a.out
drwxrwxr-x 2 nischal nischal 4096 Sep 23 16:35 dir
-rw-rw-r-- 1 nischal nischal 123 Sep 24 16:02 helloworld.c
-rwxr-w-r-- 1 nischal nischal 346 Sep 24 01:47 myScript.sh
-rwxrwxr-x 1 nischal nischal 16072 Sep 24 21:28 q2
-rw-rw-r-- 1 nischal nischal 290 Sep 24 21:28 q2.c
-rwxrwxr-x 1 nischal nischal 16160 Sep 24 22:45 q3
-rw-rw-r-- 1 nischal nischal 430 Sep 24 22:45 q3.c
-rwxrwxr-x 1 nischal nischal 16160 Sep 24 23:15 q4
-rw-rw-r-- 1 nischal nischal 341 Sep 24 23:14 q4.c
-rwxrwxr-x 1 nischal nischal 15992 Sep 25 00:49 q5
-rw-rw-r-- 1 nischal nischal 224 Sep 25 00:03 q5.c
-rwxr-w-r-- 1 nischal nischal 0 Sep 23 17:14 test
-rw-rw-r-- 1 nischal nischal 0 Sep 23 17:39 text
```

```
nischal at anton in ~/Demo
$ pwd
/home/nischal/Demo
```

```
nischal at anton in ~/Demo
$ mv test ./dir/
```

```
nischal at anton in ~/Demo
$ cd dir/
nischal at anton in ~/Demo/dir
$ rm file1 file2 test
nischal at anton in ~/Demo/dir
$ cd ..
nischal at anton in ~/Demo
$ rmdir dir
```

```

nischal at anton in ~/Demo $2 -r,w to owner and group, r to others
$ mkdir dir
c. file3 -r,w to owner, r to group, none to others
nischal at anton in ~/Demo $ file named 'test.txt' having following lines:
$ ls
a.out dir helloworld.c myScript.sh q2 q2.c q3 q3.c q4 q4.c q5 q5.c test.txt text
nischal at anton in ~/Demo $

```

```

11. See difference between man and whatis
nischal at anton in ~/Demo $ ls
a.out helloworld.c myScript.sh q2 q2.c q3 q3.c q4 q4.c q5 q5.c text
12. Create 3 files (file1, file2, file3) & assign permissions:
nischal at anton in ~/Demo $2 -r,w to owner and group, r to others
$ cp text test.txt
c. file3 -r,w to owner, r to group, none to others
nischal at anton in ~/Demo $ file named 'test.txt' having following lines:
$ ls
a.out helloworld.c myScript.sh q2 q2.c q3 q3.c q4 q4.c q5 q5.c test.txt text

```

2. Enter two commands echo* and ls. What do you think the echo did?
 -> echo printed all the files and directories inside the current directory because of the wildcard character which is "*" it selects everything.

3. what does cd do when used without arguments?
 -> it takes us to the home directory.

4. See the difference between cd ~ and cd - and cd commands.
 -> cd ~ takes you to the home directory
 cd - takes to the last directory in your history

5. Use cat command to create append and display a file.

```

nischal at anton in ~
$ cat > newfile
hello world
nischal at anton in ~
$ cat >> newfile
another line
nischal at anton in ~
$ cat newfile
hello world
another line
nischal at anton in ~
$

```

6. make a directory bar1 with two empty files first.c and second.c
 - a. Make another directory bar2 now copy all files from bar1 to bar2.
 - b. Delete directory bar1 and all its files.

```
nischal at anton in ~
$ mkdir bar1

nischal at anton in ~
$ cd bar1/

nischal at anton in ~/bar1
$ touch first.c second.c

nischal at anton in ~/bar1
$ cd ..

nischal at anton in ~
$ mkdir bar2

nischal at anton in ~
$ cp ./bar1/first.c ./bar1/second.c ./bar2

nischal at anton in ~
$ rm -a bar1
rm: invalid option -- 'a'
Try 'rm --help' for more information.

nischal at anton in ~
$ rm -r bar1

nischal at anton in ~
$ cd bar2

nischal at anton in ~/bar2
$ ls
first.c  second.c

nischal at anton in ~/bar2
$
```

7. How will you copy dir1 to existing dir 2?

Using cp command:

```
cp dir1 ./dir2
```

8. delete directory dir1 and all the files present in this directory.

```
nischal at anton in ~
$ mkdir dir1

nischal at anton in ~
$ cd dir1

nischal at anton in ~/dir1
$ touch file file2 file3

nischal at anton in ~/dir1
$ cd ..

nischal at anton in ~
$ rm -r dir1
```

9.create a new file. Set the permissions of the file to have all permissions for yourself.

```
nischal at anton in ~
$ touch newfile

nischal at anton in ~
$ ls -l newfile
-rwxrwxrwx 1 nischal nischal 0 Sep 25 13:40 newfile

nischal at anton in ~
$ chmod u+rwx newfile

nischal at anton in ~
$ ls -l newfile
-rwxrwxrwx 1 nischal nischal 0 Sep 25 13:40 newfile
```

10.practice below commands:

Chmod, find, head, more, tail and chgrp

Chmod is used to modify the file permission

Find is used to search a file in a directory can be used for many other cases too

Head returns first 10 lines of a text file by default you can pass -n flag to modify the lines of output it produces.

Tail command is similar to head but it returns the last 10 lines by default it can also be modified using -n flag.

chgrp command lets you modify the group ownership of a file or directory similar to chown but it's for group instead of user.

11. See the difference between man and whatis.

Man is used to see manual of commands.

Whatis is one line manual it does not show detailed view like man does.

12.create 3 file (file1 file2 file3)& assign permissions.

```
nischal at anton in ~
$ touch file1 file2 file3
nischal at anton in ~
$ chmod ug+rw file1
nischal at anton in ~
$ ls -l file1
-rw-rw-r-- 1 nischal nischal 0 Sep 25 13:57 file1
nischal at anton in ~
$ chmod ug+rw file2
nischal at anton in ~
$ chmod o+r file2
nischal at anton in ~
$ ls -l file2
-rw-rw-r-- 1 nischal nischal 0 Sep 25 13:57 file2
nischal at anton in ~
$ chmod u+rw file3
nischal at anton in ~
$ chmod g+r file3
nischal at anton in ~
$ chmod o-rwx file3
nischal at anton in ~
$ ls -l file3
-rw-rw---- 1 nischal nischal 0 Sep 25 13:57 file3
```

13.create a file names test.txt having following lines:

I wish to wish the wish you wish to wish,

But if you wish the wish the witch wishes,

I won't wish the wish you wish to wish.

Now use grep command to print the lines matching words wish.

```
nischal at anton in ~  
$ cat newfile  
I wish to wish the wish you wish to wish,  
But if you wish the wish the witch wishes,  
I won't wish the wish you wish to wish.  
nischal at anton in ~  
$ grep -n "wish" newfile  
1:I wish to wish the wish you wish to wish,  
2:But if you wish the wish the witch wishes,  
3:I won't wish the wish you wish to wish.  
nischal at anton in ~  
$
```

14. Change the default system variable PS1 To a new value containing the current date along with username and hostname.

```
nischal at anton in ~  
$ PS1="[\d \u \h]\n"$  
→  
→ ^C  
nischal at anton in ~  
$ PS1='\d \u \h'  
Sun Sep 25 nischal anton
```

15.Explore all the commands with all the options taught in today's class

-> Successfully explored all the commands and their variations.

Assignment 2

1. Redirect both the output and error of a command to a file

```
$ echo "error simulation" 2> newfile.txt

nischal at anton in ~/Demo
$ cat newfile.txt
Command 'eccho' not found, did you mean:
  command 'echo' from deb coreutils (8.32-4.1ubuntu1)
Try: sudo apt install <deb name>

nischal at anton in ~/Demo
$ echo "Hello World!" 2> newfile.txt
Hello World!

nischal at anton in ~/Demo
$ cat newfile.txt

nischal at anton in ~/Demo
$ echo "Hello World!" > newfile.txt

nischal at anton in ~/Demo
$ cat newfile.txt
Hello World!

nischal at anton in ~/Demo
$
```

2. Include your current directory in the PATH environment variable using export command

```
nischal at anton in ~/Demo
$ echo $HOME
/home/nischal

nischal at anton in ~/Demo
$ export PATH="$HOME/Demo:$PATH"

nischal at anton in ~/Demo
$ echo $PATH
/home/nischal/Demo:/home/nischal/bin:/home/nischal/.local/bin:/home/nischal/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin

nischal at anton in ~/Demo
$
```


3. Create 2 files "file1" and "file2". Create a directory "dir". Copy the files to the directory and then delete the files. Do so using shell script.

```
myScript.sh (~ / Demo) - VIM      Demo
12 #! /bin/bash
11
10 touch file1 file2
9 echo "file1 and file1 created successfully!"
8
7 mkdir dir
6 echo "Directory dir created successfully!"
5
4 cp file1 file2 ./dir
3 echo "file1 and file2 copied to dir successfully"
2
1 rm file1 file2
13 echo "removed file1 and file2 successfully"
~
~
~
~
~
~
~
myScript.sh      13,42      All
```

```
myScript.sh (~ / Demo) - VIM      Demo
$ ./myScript.sh
file1 and file1 created successfully!
mkdir: cannot create directory 'dir': File exists
Directory dir created successfully!
file1 and file2 copied to dir successfully
removed file1 and file2 successfully

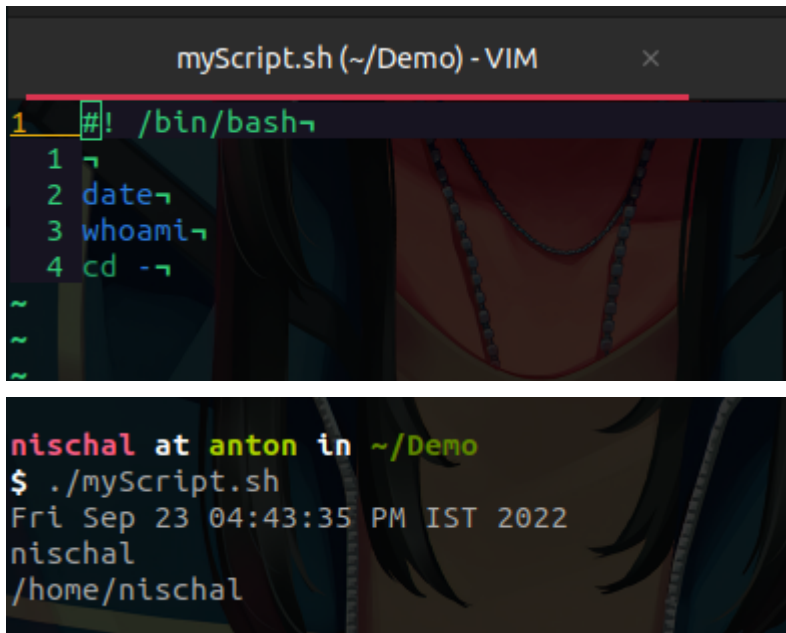
nischal at anton in ~/Demo
$ ls
dir helloworld.c myScript.sh

nischal at anton in ~/Demo
$ tree
.
├── dir
│   ├── file1
│   └── file2
├── helloworld.c
└── myScript.sh

1 directory, 4 files

nischal at anton in ~/Demo
```


4. Write a script to see current date, time, username and current directory.

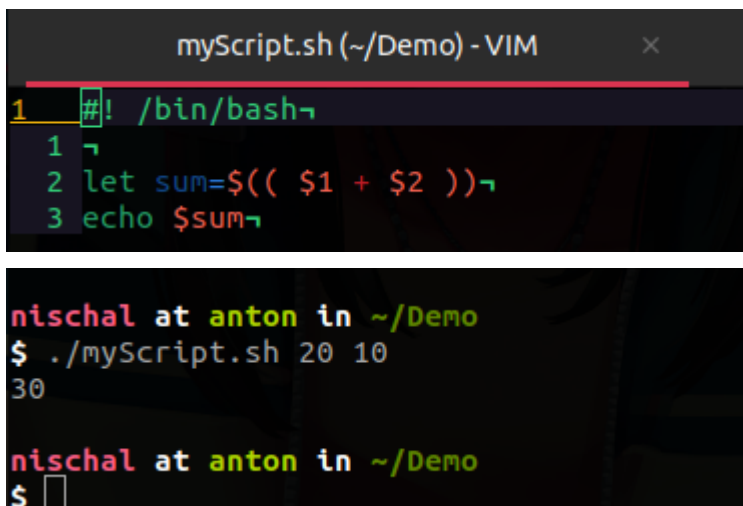


The image shows two screenshots from a terminal window. The top screenshot is a Vim editor window titled 'myScript.sh (~/.Demo) - VIM'. It displays a script with four lines: a shebang line '#!/bin/bash', followed by 'date', 'whoami', and 'cd -'. The bottom screenshot shows the terminal output after running './myScript.sh'. It displays the current date and time as 'Fri Sep 23 04:43:35 PM IST 2022', the username 'nischal', and the current directory '/home/nischal'.

```
myScript.sh (~/.Demo) - VIM
1 #!/bin/bash
2 date
3 whoami
4 cd -

nischal at anton in ~/.Demo
$ ./myScript.sh
Fri Sep 23 04:43:35 PM IST 2022
nischal
/home/nischal
```

5. Write shell script that will add two numbers, which are supplied as command line argument.



The image shows two screenshots from a terminal window. The top screenshot is a Vim editor window titled 'myScript.sh (~/.Demo) - VIM'. It displays a script with three lines: a shebang line '#!/bin/bash', followed by 'let sum=\$((\$1 + \$2))', and 'echo \$sum'. The bottom screenshot shows the terminal output after running './myScript.sh 20 10'. It displays the result '30'.

```
myScript.sh (~/.Demo) - VIM
1 #!/bin/bash
2 let sum=$(( $1 + $2 ))
3 echo $sum

nischal at anton in ~/.Demo
$ ./myScript.sh 20 10
30

nischal at anton in ~/.Demo
$
```

6. Write a script to determine whether given file exists or not, the file name is supplied as command line argument, also check for sufficient number of command line arguments.

```
myScript.sh (~ /Demo) - VIM
1  #!/bin/bash
2  if [[ $# > 1 || $# < 1 ]];
3  then
4  ▶ echo "Please enter atleast and atleast one file name"
5  ▶ exit
6  fi
7
8  FILE=$1
9
10 if [[ -f "$FILE" ]];
11 then
12 ▶ echo "$FILE exists."
13 else
14 ▶ echo "$FILE does not exists"
15 fi
16
~
~
~
```

```
nischal at anton in ~/Demo
$ ./myScript.sh test test
Please enter atleast and atleast one file name

nischal at anton in ~/Demo
$ ./myScript.sh test
test exists.

nischal at anton in ~/Demo
$ ./myScript.sh helloworld.c
helloworld.c exists.

nischal at anton in ~/Demo
$
```

7. Write a script to print nos. as 5, 4, 3, 2, 1 using while loop.

```
1  #!/bin/bash
2  NUM=5
3
4  while [ $NUM -ge 1 ]
5  do
6      echo $NUM
7      NUM=`expr $NUM - 1`
8  done
```

```
nischal at anton in ~/Demo
$ ./myScript.sh
5
4
3
2
1
nischal at anton in ~/Demo
$
```

8. Take a number as command line and using until loop print value from 1 to till number.

```
1 #!/bin/bash
2 NUM=1
3
4 while [ $NUM -le $1 ]
5 do
6     echo "$NUM"
7     NUM=`expr $NUM + 1`
8 done
~
~
~
~
~
~
~
~
~
~
myScript.sh
"myScript.sh" 9L, 85B
```

```
$ ./myScript.sh 20
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

nischal at anton in ~/Demo
$
```

9. Write a script, using case statement to perform basic math operation as follows

+ addition

-subtraction

x multiplication

/ division

NOTE- two numbers will be given as command line arguments.

```

1  #!/bin/bash
2  read -p "Enter 1->add 2->sub 3->mul 4->div :" NUM
3  echo "$1 $2"
4
5  case $NUM in
6
7  1)
8  echo "$1 + $2 = $((($1+$2))"
9  ;;
10 2)
11 echo "$1 - $2 = $((($1-$2))"
12 ;;
13 3)
14 echo "$1 * $2 = $((($1*$2))"
15 ;;
16 4)
17 echo "$1 / $2 = $((($1/$2))"
18 ;;
19 *)
20 echo "please enter a valid number between 1 and 4"
21 esac

```

~

~

~

myScript.sh 1,2

"myScript.sh" 22L, 311B

```

nischal at anton in ~/Demo
$ ./myScript.sh 9 9
Enter 1->add 2->sub 3->mul 4->div :4
9 9
9 / 9 = 1

nischal at anton in ~/Demo
$ 

```

10. Write a script to find out biggest number from three given numbers. Numbers are supplied as command line arguments. Print error if sufficient arguments are not supplied.

11. Write a program using while loop to print

0

1 0

2 1 0

3 2 1 0

4 3 2 1 0

5 4 3 2 1 0

6 5 4 3 2 1 0

7 6 5 4 3 2 1 0

8 7 6 5 4 3 2 1 0

9 8 7 6 5 4 3 2 1 0

```
nischal at anton in ~/Demo
$ ./myScript.sh
0
10
210
3210
43210
543210
6543210
76543210
876543210
9876543210

nischal at anton in ~/Demo
$
```

```
1 #!/bin/bash
2 NUM=0
3 while [ "$NUM" -lt 10 ]
4 do
5     NUM2=$NUM
6     while [ "$NUM2" -ge 0 ]
7     do
8         echo -n $NUM2
9         NUM2=$((NUM2-1))
10    done
11    echo
12    NUM=$((NUM+1))
13 done
```

```
~
~
~
~
~
~
~
myScript.sh
"myScript.sh" 14L, 156B
```

12. Write a program using for loop to print

```
*
*
* *
* * *
* * * *
* * * *
* * *
* *
*
*
```

```
1 #!/bin/bash
2
3 for NUMBER in 1 1 2 3 4 4 3 2 1
4 do
5     for(( NUM=0; NUM<$NUMBER; NUM++ ))
6     do
7         echo -n "*"
8     done
9     echo
10 done
```

```
nischal at anton in ~/Demo
$ ./myScript.sh
*
*
* *
* * *
* * * *
* * * *
* * *
* *
*
*

nischal at anton in ~/Demo
$
```

13. Write a script to print given number in reverse order, for eg. If number is 123, it must print as 321.


```
1 #!/bin/bash
2 NUM=$1
3 RES=0
4
5 while [ $NUM -gt 0 ]
6 do
7     RES=$((RES*10))
8     RES=$((RES + (($NUM%10)))
9     NUM=$((NUM/10))
10 done
11
12 echo $RES
13
14
~
~
~
~
~
~
~

nischal at anton in ~/Demo
$ ./myScript.sh 123
321
```

14. Write script to print the sum of all the digits of a given number. For eg. If the number is 123, sum of all the digits will be $1+2+3 = 6$.

```
1 #!/bin/bash↵
2 NUM=$1↵
3 RES=0↵
4 ↵
5 while [ $NUM -gt 0 ]↵
6 do↵
7   RES=$((RES + ($NUM%10)))↵
8   NUM=$((NUM/10))↵
9 done↵
10 ↵
11 echo $RES↵
12 ↵
13 ↵
~
~
~
~
~
~
```

```
nischal at anton in ~/Demo
$ ./myScript.sh 66
12

nischal at anton in ~/Demo
$
```

15. Create a file named file.txt and write a shell script to check if a file is readable, writable and executable.

```
1 #!/bin/bash
2
3 if [ -r "$1" ]
4 then
5     echo "File has read Permission";
6 else
7     echo "File does not has read Permission";
8 fi
9
10 if [ -x "$1" ]
11 then
12     echo "file has executable permission";
13 else
14     echo "file does not has executable permission";
15 fi
16
17 if [ -w "$1" ]
18 then
19     echo "file has write permission";
20 else
21     echo "file does not has write permission";
22 fi
23
~
~
~
myScript.sh
```

```
nischal at anton in ~/Demo
$ ./myScript.sh text
File has read Permission
file does not has executable permission
file has write permission

nischal at anton in ~/Demo
$ ./myScript.sh test
File has read Permission
file has executable permission
file has write permission

nischal at anton in ~/Demo
$
```