Linux Assignment – DAC (April 2024)

1. Practice the below mentioned commands with all the possible options:

cd, cat, ls, mkdir, rmdir, pwd, mv, cp

ls -> to list directives and files in the current working folder

cat -> to create files

- **cat> file1**
- **cat> file2**

cd -> to move to another directory

- **cd Desktop**

mkdir -> to create directory

- **mkdir os_lab**

rmdir -> to delete directory

- **rmdir os_lab**

pwd -> to check working directory path

mv -> to rename file or move it to another directory

-mv file1 Desktop

cp -> to copy content of one file to another

- **cp file1 newfile1**

2. Enter the two commands echo * and ls. What do you think echo did?

dac@ACTS07:~/Desktop> echo *

file1 file2 firefox.desktop google-chrome.desktop Home.desktop newfile1 trash.desktop

dac@ACTS07:~/Desktop> ls

file1 file2 firefox.desktop google-chrome.desktop Home.desktop newfile1 trash.desktop

3. What does cd do when used without arguments? **-> no output**

4. See the difference between cd ~ and cd - and cd commands

- **cd~ gives no ouput**
- **cd- shows the current directory path**

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

- cat file1

Hello

Good Morning

- cat file2

Hello

How are you?

- cat file1 >> file2 ---> to append one files content to another we use >>

- cat file2

O/P-> Hello

How are you?

Hello

Good Morning

6. Make a directory bar1 with two empty files first.c and second.c

- mkdir bar1

- cd bar1 --> move to bar1 directory

- touch first.c --> touch command is used to create empty file

- touch second.c

a. Make another directory bar2. Now copy all files from bar1 to bar2

- cd .. ---> go out of bar1 directory to create bar2

- mkdir bar2

- cd bar1 ---> navigate to bar1

b. Delete directory bar1 and all its files

- rmdir bar1

7. How will you copy a directory 'dir1' to an existing directory 'dir2' ?

NOTE- copy whole directory (not only files) from dir1 to dir2

- cp -R dir1 dir2 ---> -R is used to copy directory and its content

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

- rmdir dir1

(to check go to dir2)

dac@ACTS07:~/Desktop> cd dir2

dac@ACTS07:~/Desktop/dir2> ls

O/P-> dir1

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

- cat > newfile

- chmod 777 newfile

- ls –l

-rwxrwxrwx 1 dac DAC 0 Apr 1 21:06 newfile

10. Practice the below mentioned commands with all the possible options:

head, tail, & chgrp

head filename : read or display top 5 lines from a file

tail filename : read or display bottom 5 lines from a file

11. See difference between man and whatis

Man: allows users to access detailed information about various commands

whatis: Gives a very brief one-line summary of a command

12. Create 3 files (file1, file2, file3) & assign permissions:

- **cat > file1**
- **cat > file2**
- **cat > file3**

a. file1 - r,w to owner, group, and others.

- **chmod 666 file1**

b. file2 - r,w to owner and group, r to others

- **chmod 664 file2**

c. file3 - r,w to owner, r to group, none to others

- **chmod 640 file3**

(to check)

- **ls**

O/P-

-rw-rw-rw- 1 dac DAC 0 Apr 1 21:40 file1

-rw-rw-r-- 1 dac DAC 0 Apr 1 21:41 file2

-rw-r----- 1 dac DAC 0 Apr 1 21:41 file3

13. Create a file named '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.

- **cat test.txt**
  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 'wish' words.

cat test.txt | grep wish

Also print the lines as well as line number at the beginning of line containing 'wish' word in file

- **grep -n "wish" test.txt**

O/P->

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.

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

- **export PS1="\d \u@\h $ "**

O/P -> Mon Apr 01 dac@ACTS07 $

Shell Programming Assignment – DAC (April 2024)

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

```
#!/bin/bash

eccho s > errfile 2>&1
```

```
stuti@stuti-1:~/assignment2$ ./Q1.sh
stuti@stuti-1:~/assignment2$ ls
dir       filetocheck  Q11.sh  Q13.sh  Q15.sh  Q2.sh  Q4.sh  Q6.sh  Q8.sh
errfile   Q10.sh       Q12.sh  Q14.sh  Q1.sh   Q3.sh  Q5.sh  Q7.sh  Q9.sh
stuti@stuti-1:~/assignment2$ cat errfile
./Q1.sh: line 2: eccho: command not found
```

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

```
#!/bin/bash
export PATH="$PATH:$(pwd)"
```

```
stuti@stuti-1:~/assignment2$ ./Q2.sh
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/game
s:/snap/bin:/snap/bin:/home/stuti/assignment2
```

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.

```
#!/bin/bash
touch file1
touch file2
mkdir -p dir
cp file1 dir/
cp file2 dir/
rm file1
rm file2
```

```
stuti@stuti-1:~/assignment2$ ./Q3.sh
stuti@stuti-1:~/assignment2$ ls
dir       filetocheck  Q11.sh  Q13.sh  Q15.sh  Q2.sh  Q4.sh  Q6.sh  Q8.sh
errfile   Q10.sh       Q12.sh  Q14.sh  Q1.sh   Q3.sh  Q5.sh  Q7.sh  Q9.sh
stuti@stuti-1:~/assignment2$ cd dir
stuti@stuti-1:~/assignment2/dir$ ls
file1  file2
```

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

```
#!/bin/bash
echo $(date)
echo $USER
echo #PWD
```

```
stuti@stuti-1:~/assignment2$ ./Q4.sh
Tue Apr 16 12:40:39 PM IST 2024
stuti
/home/stuti/assignment2
```

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

```
echo -n "Enter num: "
read num1
read -p "Enter num2: " num2
sum=$(($num1+$num2))
echo $sum
```

```
stuti@stuti-1:~/assignment2$ ./Q5.sh
Enter num: 56
Enter num2: 32
88
```

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.

```
#!/bin/bash
read -p "Enter file name to check in assignment2 dir: " file
if [ -f /home/stuti/assignment2/$file ];
then
```

```
    echo "Exists"
else
    echo "Does not Exist"
fi
```

```
stuti@stuti-1:~/assignment2$ ./Q6.sh
Enter file name to check in assignment2 dir: Q1.sh
Exists
stuti@stuti-1:~/assignment2$ ./Q6.sh
Enter file name to check in assignment2 dir: xyz.sh
Does not Exist
```

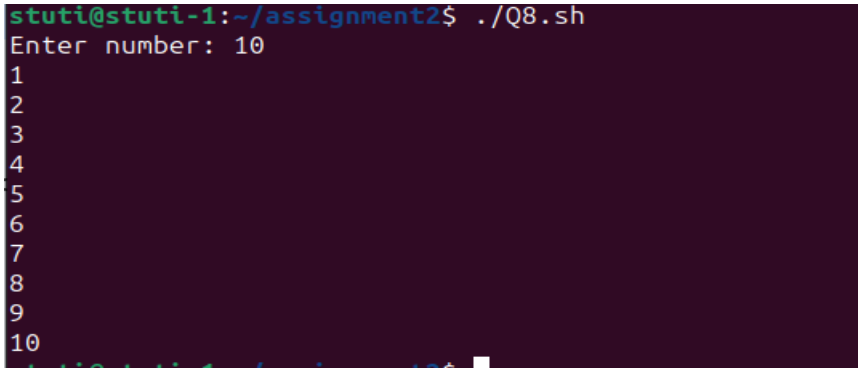
7. Write a script to print nos. as 5, 4, 3, 2, 1 using while loop.

```
#!/bin/bash
c=5
while [ $c -gt 0 ];
do
    echo "c is " $c
    ((c--))
done
```

```
stuti@stuti-1:~/assignment2$ ./Q7.sh
c is  5
c is  4
c is  3
c is  2
c is  1
```

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

```
#!/bin/bash
read -p "Enter number: " n
c=1
while [ $c -le $n ];
do
    echo $c
    ((c++))
done
```



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.

```
#!/bin/bash
read -p "Enter number 1: " x
read -p "Enter number 2: " y
echo -n "Choose any number 1-addition 2-subtraction 3-
multiplication 4-division: "
read n
case $n in
    1) echo "Sum: " $(($x+$x));;
    2) echo "Sub: " $(($x-$y));;
    3) echo "Mul: " $(($x*$y));;
    4) echo "Div: " $(($x/$y));;
    *) echo "Invalid";;
esac
```

```
stuti@stuti-1:~/assignment2$ ./Q9.sh
Enter number 1: 30
Enter number 2: 5
Choose any number 1-addition 2-subtraction 3-multiplication 4-division: 4
Div:  6
stuti@stuti-1:~/assignment2$ ./Q9.sh
Enter number 1: 30
Enter number 2: 5
Choose any number 1-addition 2-subtraction 3-multiplication 4-division: 3
Mul:  150
stuti@stuti-1:~/assignment2$ ./Q9.sh
Enter number 1: 30
Enter number 2: 5
Choose any number 1-addition 2-subtraction 3-multiplication 4-division: 2
Sub:  25
stuti@stuti-1:~/assignment2$ ./Q9.sh
Enter number 1: 30
Enter number 2: 5
Choose any number 1-addition 2-subtraction 3-multiplication 4-division: 1
Sum:  60
```

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.

```
#!/bin/bash
read -p "Enter number 1: " num1
read -p "Enter number 2: " num2
read -p "Enter number 3: " num3

if [ -z "$num1" ]; then
    echo "Value of num1 is not assigned"
elif [ -z "$num2" ]; then
    echo "Value of num2 is not assigned"
elif [ -z "$num3" ]; then
        echo "Value of num3 is not assigned"
fi
if [ ! -z $num1 ] && [ ! -z $num2 ] && [ ! -z $num3 ];
then
    if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]; then
        echo $num1 " is greatest"
    elif [ $num2 -gt $num3 ]; then
        echo $num2 " is greatest"
    else
        echo $num3 " is greatest"
```

```
    fi
fi
```

```
stuti@stuti-1:~/assignment2$ ./Q10.sh
Enter number 1: 45
Enter number 2: 34
Enter number 3: 6
45  is greatest
stuti@stuti-1:~/assignment2$ ./Q10.sh
Enter number 1: 23
Enter number 2:
Enter number 3: 4
Value of num2 is not assigned
```

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

```
#!/bin/bash
c=0
while [ $c -lt 10 ];
do
    i=$c
    while [ $i -ge 0 ];
    do
        echo -n $i
        ((i--))
    done
```

```
    echo -e
    ((c++))
done
```

```
stuti@stuti-1:~/assignment2$ ./Q11.sh
0
10
210
3210
43210
543210
6543210
76543210
876543210
9876543210
```

12. Write a program using for loop to print

*

*

* *

* * *

* * * *

* * * *

* * *

* *

*

*

```
#!/bin/bash
echo "*"
c=1
while [ $c -le 4 ];
do
    n=$c
    while [ $n -gt 0 ];
    do
```

```
        echo -n "*"
        ((n--))
    done
    echo -e
    ((c++))
done
s=4
while [ $s -ge 1 ];
do
    i=$s
    while [ $i -gt 0 ];
    do
        echo -n "*"
        ((i--))
    done
    echo -e
    ((s--))
done
echo "*"
```

```
stuti@stuti-1:~/assignment2$ ./Q12.sh
*
*
**
***
****
****
***
**
*
*
```

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

```
#!/bin/bash
read -p "Enter number: " n
rev=" "
while [ $n -gt 0 ]
do
```

```
    rem=$((n%10))
    rev=$rev$rem
    n=$((n/10))
done
echo $rev
```

```
stuti@stuti-1:~/assignment2$ ./Q13.sh
Enter number: 12345
54321
```

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.

```
#!/bin/bash
read -p "Enter number: " n
sum=0
while [ $n -gt 0 ];
do
    rem=$(($n%10))
    sum=$(($sum+$rem))
    n=$(($n/10))
done
echo $sum
```

```
stuti@stuti-1:~/assignment2$ ./Q14.sh
Enter number: 765
18
```

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

```
#!/bin/bash
file_path="/home/dac/filetocheck"
if [ -w "$file_path" ] && [ -r "$file_path" ] && [ -x
"$file_path" ]; then
    echo "file is with rwx permission"
else
```

```
    echo "file is not with rwx permission"
fi
```

```
stuti@stuti-1:~/assignment2$ ./Q14.sh
Enter number: 765
18
stuti@stuti-1:~/assignment2$ ./Q15.sh
file is not with rwx permission
```