Concepts of Operating System

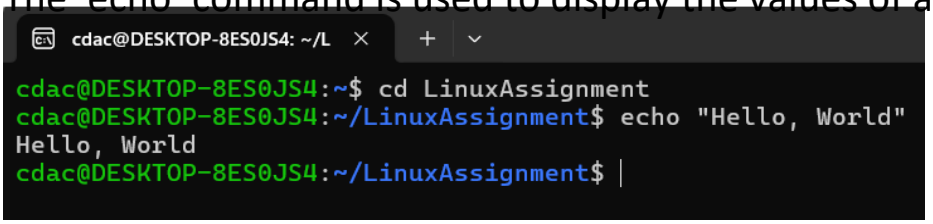
Assignment 2

Part A

What will the following commands do?

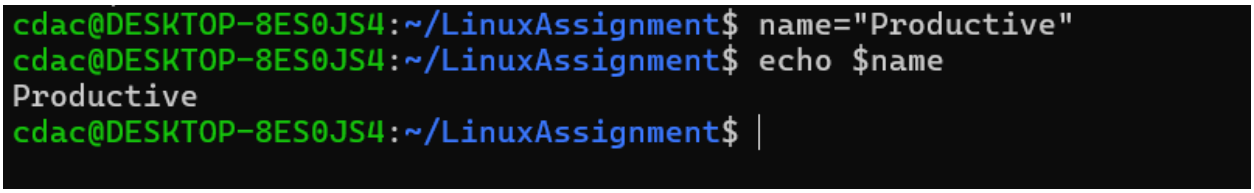
- echo "Hello, World!”

The ‘echo’ command is used to display the values of a variable.



- name="Productive

Name is the variable name and “Productive” is the string value assigned to the name variable.



- touch file.txt

Run the touch command to create a new empty file.txt in a specific directory.

```
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ touch file.txt
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ ls
abc.txt  docs.zip      file.txt   file2.txt  fruit.txt  numbers.txt
docs     duplicate.txt  file1.txt  file3.txt  input.txt  output.txt
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$
```

- ls -a

The ls -a command is used to list all files and directories in a directory including hidden ones.

ls ->Lists files and directories.

-a -> Shows hidden files.

```
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ ls -a
.   abc.txt  docs.zip      file.txt   file2.txt  fruit.txt  numbers.txt
..  docs     duplicate.txt  file1.txt  file3.txt  input.txt  output.txt
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$
```

- rm file.txt

The rm command deletes files from a directory.

```
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ rm file.txt
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ ls
abc.txt  docs  docs.zip  duplicate.txt  file1.txt  file2.txt  file3.txt  fruit.txt  input.txt  numbers.txt  output.txt
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$
```

- cp file1.txt file2.txt

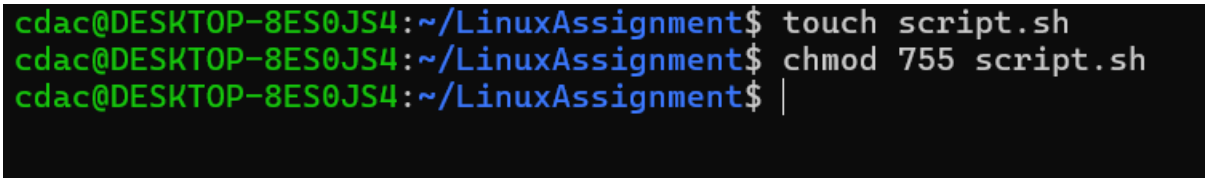
Use the **cp** command to copy files from your current directory to another folder.

- mv file.txt /path/to/directory/

The main usage of the **mv** command is to move a file or folder to another location.

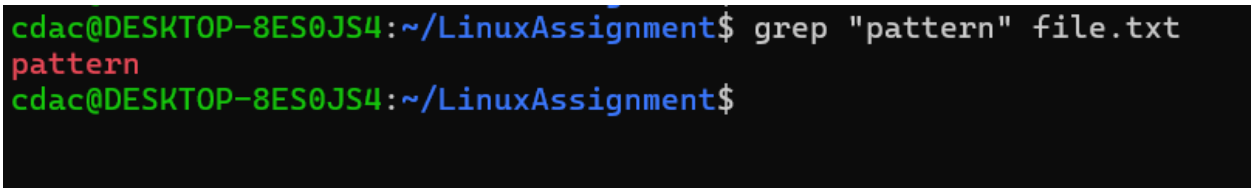
- chmod 755 script.sh

Chmod lets you change the permission of files or directories.



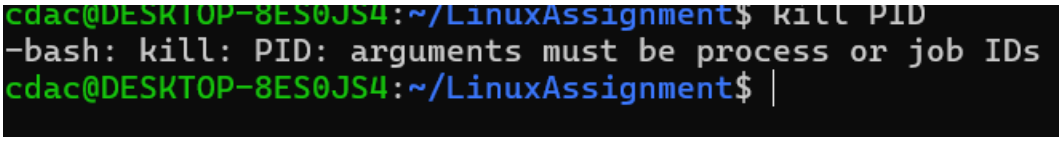
- grep "pattern" file.txt

Global regular expression print or **grep** lets you search specific lines from a file using keywords. It is useful for filtering large data like logs.



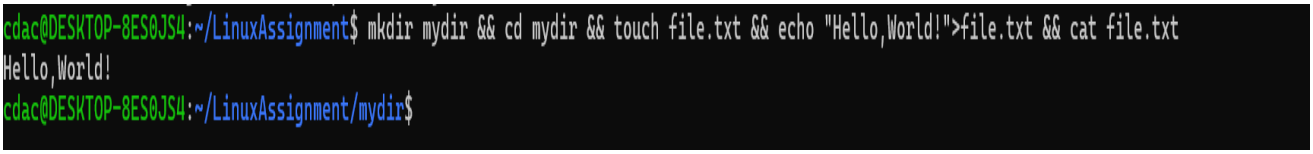
• kill PID

Use the kill command to terminate a process using its ID.



• mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt

The **mkdir** command lets you create one or multiple directories. The syntax looks like this:



• ls -l | grep ".txt"

This command is used to lists files in long format and l is used to passes the output of ls -l to grep command.

```
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment/mydir$ ls -l|grep ".txt"
-rw-r--r-- 1 cdac cdac 13 Mar  1 14:47 file.txt
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment/mydir$
```

- cat file1.txt file2.txt | sort | uniq

The **concatenate** or **cat** command has various usages. The most basic one is printing the content of a file. Here's the syntax:

```
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ cat file1.txt file2.txt | sort | uniq
Ram
geeta
krishna Namrata
shyam
sita
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$
```

- ls -l | grep "^d"

Lists iles and directories in long format,l passes the output of ls -l to grep and grep "^d" which indicates directories in the permission column.

```
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ ls -l | grep "^d"
drwxr-xr-x 2 cdac cdac 4096 Feb 27 20:09 abc.txt
drwxr-xr-x 2 cdac cdac 4096 Mar  1 14:47 mydir
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$
```

- grep -r "pattern" /path/to/directory/

Grep ->Searches for a specific pattern in files.

-r -> Seraches recursively through all files in the specified directory and its subdirectories.

“pattern” -> The next or regular expression to search for.

```
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ grep -r "pattern" file.txt
pattern
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$
```

- cat file1.txt file2.txt | sort | uniq –d

The **concatenate** or **cat** command has various usages. The most basic one is printing the content of a file.

```
cdac@DESKTOP-8ES0JS4: ~/L
cdac@DESKTOP-8ES0JS4:~$ cd LinuxAssignment
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ cat file1.txt file2.txt | sort |uniq
Ram
geeta
krishna Namrata
shyam
sita
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$
```

- chmod 644 file.txt

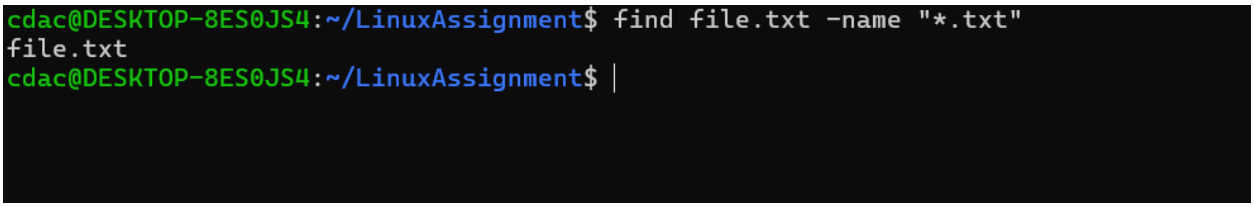
```
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ chmod 644 file.txt
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ chmod 644 file1.txt
```

- cp -r source_directory destination_directory

The **cp** command to copy files from your current directory to another folder.

- find /path/to/search -name "*.txt"

The **find** command searches for a file within a specific directory.



- chmod u+x file.txt

This command is used to grant execute permissions for file.txt to the user(Owner) of the file.

- echo $PATH

This command diplays the value of system environment variable that provide directories where executable programs are locate

Part B

Identify True or False:

1. ls is used to list files and directories in a directory. -True,
2. mv is used to move files and directories. -True,
3. cd is used to copy files and directories.-False,
4. pwd stands for "print working directory" and displays the current directory.
   True
5. grep is used to search for patterns in files.-True

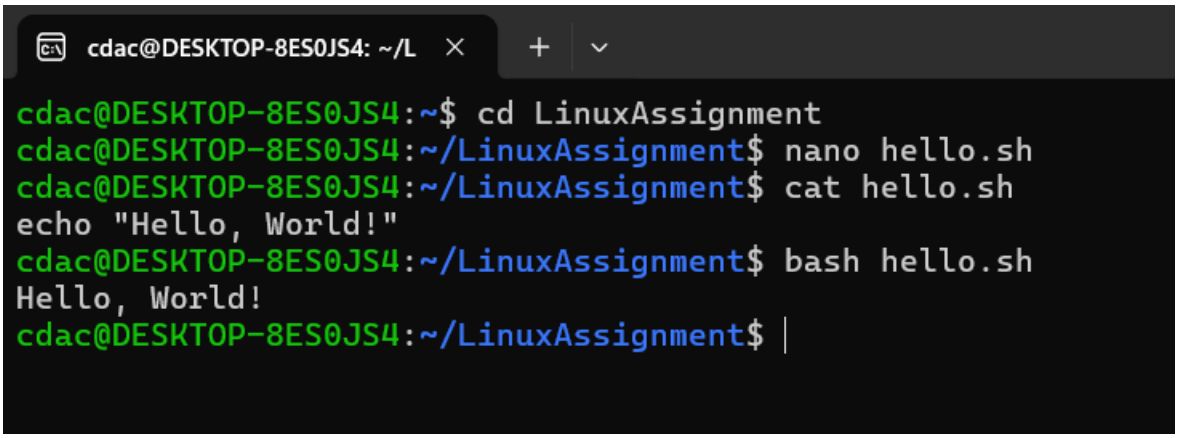
6. chmod 755 file.txt gives read, write, and execute permissions to the owner, and read and execute permissions to group and others.-True
7. mkdir -p directory1/directory2 creates nested directories, creating directory2 inside directory1 if directory1 does not exist.-True
8. rm -rf file.txt deletes a file forcefully without confirmation.-False

Identify the Incorrect Commands:

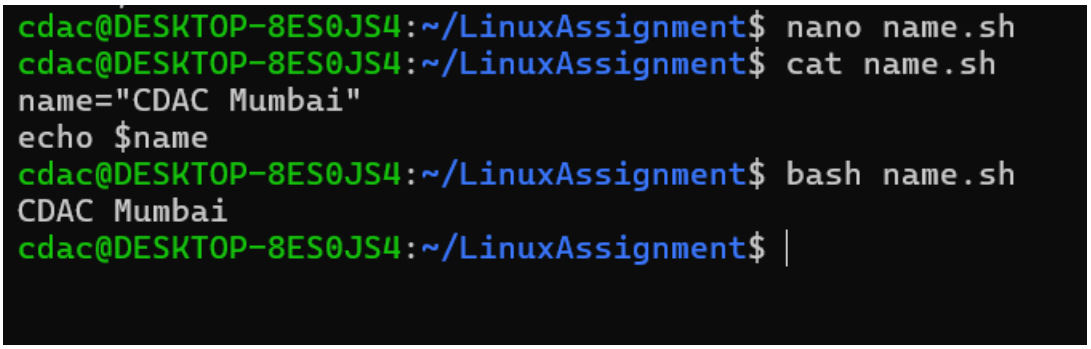
1. chmodx is used to change file permissions.
   Chmod command is used to change file permission.
2. cpy is used to copy files and directories.
   Cp command is used to files and directories.
3. mkfile is used to create a new file.
   Touch command is used to create new file. Mkdir command is used to create a new directory.
4. catx is used to concatenate files.
   cat command is used to concatenate files.
5. rn is used to rename files.
   mv command is ued to rename files when 2 files names are passed as arguments.

Part C

Question 1: Write a shell script that prints "Hello, World!" to the terminal.



• Q2. Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.



Q3. Write a shell script that takes a number as input from the user and prints it.

```
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ nano Q3.sh
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ cat Q3.sh
echo "Enter a number"
read a
echo Your numbaer is $a
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ bash Q3.sh
Enter a number
345
Your numbaer is 345
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$
```

Q4. Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.

```
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ nano Que4.sh
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ cat Que4.sh
echo "Enter a number"
read a
echo "Enter a number"
read b
sum='expr $a + $b'
echo sum of $a and $b is $sum

cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ bash Que4.sh
Enter a number
8
Enter a number
9
sum of 8 and 9 is expr $a + $b
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$
```

Q5. Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd"10:51 PM.

```
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ nano Que5.sh
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ cat Que5.sh
echo "Enter a number"
read a
if [ 'expr $a % 2' -eq 0 ]
then
        echo "$a is an even number"
else
        echo "$a is an odd number"
fi
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ bash Que5.sh
Enter a number
5
Que5.sh: line 3: [: expr $a % 2: integer expression expected
5 is an odd number
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$
```

Q6. Write a shell script that uses a for loop to print numbers from 1 to 5.

```
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ nano Que6.sh
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ cat Que6.sh
for i in 1 2 3 4 5
do
            echo $i
done
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ bash Que6.sh
1
2
3
4
5
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$
```

Q7. Write a shell script that uses a while loop to print numbers from 1 to 5.•

```
cdac@DESKTOP-8ES0JS4:~$ cd LinuxAssignment
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ nano Que7.sh
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ cat Que7.sh
a=1
while [ $a -lt 6 ]
do
        echo $a
a='expr $a + 1'
done
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ bash Que7.sh
1
Que7.sh: line 2: [: too many arguments
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ nano Que7.sh
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ cat Que7.sh
a=0
while [ $a -lt 6 ]
do
        echo $a
a='expr $a + 1'
done
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ bash Que7.sh
0
Que7.sh: line 2: [: too many arguments
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$
```

Q8. Write a shell script that checks if a file named "file.txt" exists in the current directory. If it does, print "File exists", otherwise, print "File does not exist".

```
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ nano Que8.sh
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ cat Que8.sh
if [ -e file.txt ]
then
        echo "File exists"
else
        echo "File doesn't exists"
fi
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ bash Que8.sh
File exists
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$
```

Q9. Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.

```
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ nano Que9.sh
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ cat Que9.sh
echo "Enter a number" ; read a
if [ $a -gt 10 ]
then
        echo "$a is greater than 10"
else
        if [ $a -eq 10 ]
        then
                echo "$a is equal to 10"
        else
                echo "$a is smaller than 10"
        fi
fi
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ bash Que9.sh
Enter a number
6
6 is smaller than 10
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ bash Que9.sh
Enter a number
11
11 is greater than 10
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$
```

Q11. Write a shell script that uses a while loop to read numbers from the user until the user enters a negative number. For each positive number entered, print its square. Use the break statement to exit the loop when a negative number is entered.

```
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ nano Que11.sh
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ cat Que11.sh
while [ true ]
do
        echo "Enter a number" ; read a
        if [ $a -lt 0 ]
        then
                break
        fi
done
echo "Program Terminated"

cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$ bash Que11.sh
Enter a number
4
Enter a number
3
Enter a number
2
Enter a number
5
Enter a number
-1
Program Terminated
cdac@DESKTOP-8ES0JS4:~/LinuxAssignment$
```