

# Concepts of Operating System

## Assignment 2

### Part A

What will the following commands do?

- `echo "Hello, World!"`  
→ To print statement.

```
cdac@DESKTOP-CNIG80F:~/demo$ echo "Hello, World!"  
Hello, World!
```

- `name="Productive"`  
→ Declare & Assigning value.

```
cdac@DESKTOP-CNIG80F:~/demo$ name="Productive"  
cdac@DESKTOP-CNIG80F:~/demo$ echo $name  
Productive
```

- `touch file.txt`  
→ To create files.

```
cdac@DESKTOP-CNIG80F:~/demo$ touch file.txt  
cdac@DESKTOP-CNIG80F:~/demo$ ls
```

- `ls -a`  
→ List all files including hidden files.

```
cdac@DESKTOP-CNIG80F:~/demo$ ls -a  
..  .  file.txt  text1.txt  text2.txt  text3.txt
```

- `rm file.txt`  
→ Delete files.

```
cdac@DESKTOP-CNIG80F:~/demo$ rm file.txt  
cdac@DESKTOP-CNIG80F:~/demo$ ls  
text1.txt  text2.txt  text3.txt  
cdac@DESKTOP-CNIG80F:~/demo$
```

- `cp file1.txt file2.txt`  
→ To copy content

```
cdac@DESKTOP-CNIG80F:~/demo$ cat text1.txt  
1  
2  
1  
cdac@DESKTOP-CNIG80F:~/demo$ cat text3.txt  
cdac@DESKTOP-CNIG80F:~/demo$ cp text1.txt text3.txt  
cdac@DESKTOP-CNIG80F:~/demo$ cat text3.txt  
1  
2  
1  
cdac@DESKTOP-CNIG80F:~/demo$ _
```

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

→ To move a file from one folder to another.

```
cdac@DESKTOP-CNIG80F:~/demo$ ls
file.txt  path  text1.txt  text2.txt  text3.txt
cdac@DESKTOP-CNIG80F:~/demo$ mv file.txt ./path/to/directory/
cdac@DESKTOP-CNIG80F:~/demo$ cd path/to/directory/
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$ ls
file.txt
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$ _
```

- chmod 755 script.sh

→ Give the owner of a script full permissions and give everyone else read and execute permissions.

```
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$ ls -l
total 0
-rw-r--r-- 1 cdac cdac 0 Aug 30 18:57 file.txt
-rw-r--r-- 1 cdac cdac 0 Aug 30 18:58 script.sh
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$ chmod 755 script.sh
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$ ls -l
total 0
-rw-r--r-- 1 cdac cdac 0 Aug 30 18:57 file.txt
-rwxr-xr-x 1 cdac cdac 0 Aug 30 18:58 script.sh
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$
```

- grep "pattern" file.txt

→ Use a pattern to find lines from a file.

```
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$ grep "CDAC" file.txt
CDAC Mumbai
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$ _
```

- kill PID

→ Process ID of the process to be killed.

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

→ Create mydir directory

Then go inside mydir directory

Then create file file.txt

Then write echo "Hello, World!" statement inside file.txt

Then print output on console.

```
cdac@DESKTOP-CNIG80F:~/demo$ ls
path  text1.txt  text2.txt  text3.txt
cdac@DESKTOP-CNIG80F:~/demo$ mkdir mydir
cdac@DESKTOP-CNIG80F:~/demo$ cd mydir/
cdac@DESKTOP-CNIG80F:~/demo/mydir$ touch file.txt
cdac@DESKTOP-CNIG80F:~/demo/mydir$ echo "Hello, World!" > file.txt
cdac@DESKTOP-CNIG80F:~/demo/mydir$ cat file.txt
Hello, World!
cdac@DESKTOP-CNIG80F:~/demo/mydir$ _
```

- `ls -l | grep ".txt"`

→ Listing lots of file details that are .txt

```
cdac@DESKTOP-CNIG80F:~/demo$ ls
mydir  path  text1.txt  text2.txt  text3.txt
cdac@DESKTOP-CNIG80F:~/demo$ ls -l | grep ".txt"
-rw-r--r-- 1 cdac cdac  0 Aug 28 07:56 text1.txt
-rw-r--r-- 1 cdac cdac 20 Aug 28 07:57 text2.txt
-rwxr--r-- 1 cdac cdac  0 Aug 28 08:00 text3.txt
cdac@DESKTOP-CNIG80F:~/demo$ _
```

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

→ Sort a file and displaying unique content in console.

```
cdac@DESKTOP-CNIG80F:~/demo$ ls
mydir  path  text1.txt  text2.txt  text3.txt
cdac@DESKTOP-CNIG80F:~/demo$ cat text1.txt text2.txt
1
2
1
3
1
cdac@DESKTOP-CNIG80F:~/demo$ cat text1.txt text2.txt | sort | uniq
1
1
2
3
cdac@DESKTOP-CNIG80F:~/demo$
```

- `ls -l | grep "^d"`

→

```
cdac@DESKTOP-CNIG80F:~/demo$ ls -l | grep "^d"
drwxr-xr-x 1 cdac cdac 512 Aug 30 19:02 mydir
drwxr-xr-x 1 cdac cdac 512 Aug 30 18:48 path
cdac@DESKTOP-CNIG80F:~/demo$
```

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

→ Find specific pattern recursively in all directories.

```
cdac@DESKTOP-CNIG80F:~/demo$ ls
mydir  path  text1.txt  text2.txt  text3.txt
cdac@DESKTOP-CNIG80F:~/demo$ grep -r "Hello" ./path/to/directory/
./path/to/directory/file.txt:Hello,
cdac@DESKTOP-CNIG80F:~/demo$ _
```

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

→ Sort both files and display content with its occurrence.

```
cdac@DESKTOP-CNIG80F:~/demo$ cat text1.txt text2.txt
1
2
1
1
3
cdac@DESKTOP-CNIG80F:~/demo$ cat text1.txt text2.txt | sort | uniq -d
1
cdac@DESKTOP-CNIG80F:~/demo$ _
```

- `chmod 644 file.txt`

→ Owner gets read and write, groups get read and other get read permission.

```
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$ ls
file.txt  script.sh
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$ ls -l
total 0
-rw-r--r-- 1 cdac cdac 19 Aug 30 19:00 file.txt
-rwxr-xr-x 1 cdac cdac  0 Aug 30 18:58 script.sh
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$ chmod 644 file.txt
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$ ls -l
total 0
-rw-r--r-- 1 cdac cdac 19 Aug 30 19:00 file.txt
-rwxr-xr-x 1 cdac cdac  0 Aug 30 18:58 script.sh
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$
```

- `cp -r source_directory destination_directory`

→ Copy content recursively data from one file to another.

```
cdac@DESKTOP-CNIG80F:~/demo/source_directory$ ls
file.txt
cdac@DESKTOP-CNIG80F:~/demo/source_directory$ cd ..
cdac@DESKTOP-CNIG80F:~/demo$ ls
mydir  path  source_directory  text1.txt  text2.txt  text3.txt
cdac@DESKTOP-CNIG80F:~/demo$ cp -r source_directory/ destintion_directory
cdac@DESKTOP-CNIG80F:~/demo$ cd destintion_directory/
cdac@DESKTOP-CNIG80F:~/demo/destintion_directory$ ls
file.txt
cdac@DESKTOP-CNIG80F:~/demo/destintion_directory$
```

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

→ Display a .txt file within given directory.

```
cdac@DESKTOP-CNIG80F:~/demo$ cd path/to/directory/
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$ ls
file.txt  script.sh
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$ cd ..
cdac@DESKTOP-CNIG80F:~/demo/path/to$ cd ..
cdac@DESKTOP-CNIG80F:~/demo/path$ cd ..
cdac@DESKTOP-CNIG80F:~/demo$ find ./path/to/directory/ -name "*.txt"
./path/to/directory/file.txt
cdac@DESKTOP-CNIG80F:~/demo$
```

- `chmod u+x file.txt`

→ Owner of this file gets execute permission.

```
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$ ls
file.txt  script.sh
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$ ls -l
total 0
-rw-r--r-- 1 cdac cdac 19 Aug 30 19:00 file.txt
-rwxr-xr-x 1 cdac cdac  0 Aug 30 18:58 script.sh
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$ chmod u+x file.txt
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$ ls -l
total 0
-rwxr--r-- 1 cdac cdac 19 Aug 30 19:00 file.txt
-rwxr-xr-x 1 cdac cdac  0 Aug 30 18:58 script.sh
cdac@DESKTOP-CNIG80F:~/demo/path/to/directory$
```



- echo \$PATH  
→ Print the content of the PATH variable

```
cdac@DESKTOP-CNIG80F:~/demo$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/mnt/c/Program Files/WindowsApp
s/CanonicalGroupLimited.Ubuntu22.04LTS_2204.3.63.0_x64_79rhkp1fndgsc:/mnt/c/Program Files/Common Files/Oracle/Java/java
path:/mnt/c/Windows/system32:/mnt/c/Windows:/mnt/c/Windows/System32/Wbem:/mnt/c/Windows/System32/WindowsPowerShell/v1.0/
:/mnt/c/Windows/System32/OpenSSH:/mnt/c/Program Files/Git/cmd:/mnt/c/Program Files/nodejs:/mnt/c/Program Files/MySQL/M
ySQL Shell 8.0/bin:/mnt/c/Users/DDR/AppData/Local/Microsoft/WindowsApps:/mnt/c/Users/DDR/AppData/Local/Programs/Microso
ft VS Code/bin:/mnt/c/Users/DDR/AppData/Roaming/npm:/snap/bin
cdac@DESKTOP-CNIG80F:~/demo$
```

## 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. → True

### Identify the Incorrect Commands:

1. **chmodx** is used to change file permissions. → chmod +x
2. **cpy** is used to copy files and directories. → cp
3. **mkfile** is used to create a new file. → nano, cat, touch
4. **catx** is used to concatenate files. → cat
5. **rn** is used to rename files. → mv

## Part C

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

```
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ cat script1.sh
echo "Hello, World!"

cdac@DESKTOP-CNIG80F:~/ShellProgramming$ bash script1.sh
Hello, World!
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ _
```

Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.

```
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ cat script2.sh
name="CADC Mumbai"
echo "Name : $name"
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ bash script2.sh
Name : CADC Mumbai
cdac@DESKTOP-CNIG80F:~/ShellProgramming$
```

Question 3: Write a shell script that takes a number as input from the user and prints it.

```
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ cat script3.sh
echo "Enter Your No. : "
read no
echo "No. Is : $no"
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ bash script3.sh
Enter Your No. :
Akshay
No. Is : Akshay
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ _
```

Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.

```
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ cat script4.sh
echo Enter No1 :
read no1
echo Enter No2 :
read no2

echo Addition Is : $((no1+no2))
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ bash script4.sh
Enter No1 :
10
Enter No2 :
20
Addition Is : 30
cdac@DESKTOP-CNIG80F:~/ShellProgramming$
```

Question 5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".

```
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ cat script5.sh
echo Enter No :
read no

if [ $((($no%2)) -eq 0 ]
then
    echo $no Is Even
else
    echo $no Is Odd
fi
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ bash script5.sh
Enter No :
4
4 Is Even
cdac@DESKTOP-CNIG80F:~/ShellProgramming$
```

Question 6: Write a shell script that uses a for loop to print numbers from 1 to 5.

```
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ cat script6
#!/bin/bash
n=0
for n in 1 2 3 4 5
do
    echo $n
done
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ bash script6
1
2
3
4
5
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ _
```

Question 7: Write a shell script that uses a while loop to print numbers from 1 to 5.

```
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ cat script7
#!/bin/bash
n=1
while [ $n -le 5 ]
do
    echo $n
    n=`expr $n + 1`
done
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ bash script7
1
2
3
4
5
cdac@DESKTOP-CNIG80F:~/ShellProgramming$
```



Question 8: 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-CNIG80F:~/ShellProgramming$ ls
file.txt script1.sh script10 script2.sh script3.sh script4.sh script5.sh script6 script7 script8 script9.sh
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ cat script8
#!/bin/bash

filename="./home/cadc/demo/text1.txt"

if [ -f "file.txt" ];
then
    echo Regular File Exists
else
    echo File Does Not Exist
fi

cdac@DESKTOP-CNIG80F:~/ShellProgramming$ bash script8
Regular File Exists
cdac@DESKTOP-CNIG80F:~/ShellProgramming$
```

Question 9: 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-CNIG80F:~/ShellProgramming$ cat script9.sh
#!/bin/bash

echo Enter No :
read no

if [ $no -gt 10 ]
then
    echo No is greater
else
    echo No is smaller
fi

cdac@DESKTOP-CNIG80F:~/ShellProgramming$ bash script9.sh
Enter No :
20
No is greater
cdac@DESKTOP-CNIG80F:~/ShellProgramming$
```

Question 10: Write a shell script that uses nested for loops to print a multiplication table for numbers from 1 to 5. The output should be formatted nicely, with each row representing a number and each column representing the multiplication result for that number.

```

cdac@DESKTOP-CNIG80F:~/ShellProgramming$ cat script10
#!/bin/bash

for((i=1;i<=5;i++))
do
    for((j=1;j<=10;j++))
    do
        echo $i*$j = $((i*j))
    done
done
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ bash script10
1*1 = 1
1*2 = 2
1*3 = 3
1*4 = 4
1*5 = 5
1*6 = 6
1*7 = 7
1*8 = 8
1*9 = 9
1*10 = 10
2*1 = 2
2*2 = 4
2*3 = 6
2*4 = 8

```

Question 11: 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-CNIG80F:~/ShellProgramming$ cat script11
#!/bin/bash

while true
do
    echo Enter no. :
    read no

    if [ $no -lt 0 ]
    then
        echo Negative No. Entered
        break
    else
        temp=$((no*no))
        echo The square of $no is $temp
    fi
done
cdac@DESKTOP-CNIG80F:~/ShellProgramming$ bash script11
Enter no. :
4
The square of 4 is 16
Enter no. :
-1
Negative No. Entered
cdac@DESKTOP-CNIG80F:~/ShellProgramming$

```

## Part E

1. Consider the following processes with arrival times and burst times:

Process	Arrival Time	Burst Time
P1	0	5
P2	1	3
P3	2	6

Calculate the average waiting time using First-Come, First-Served (FCFS) scheduling.

2. Consider the following processes with arrival times and burst times:

Process	Arrival Time	Burst Time
P1	0	3
P2	1	5
P3	2	1
P4	3	4

Calculate the average turnaround time using Shortest Job First (SJF) scheduling.

3. Consider the following processes with arrival times, burst times, and priorities (lower number indicates higher priority):

Process	Arrival Time	Burst Time	Priority
P1	0	6	3
P2	1	4	1
P3	2	7	4
P4	3	2	2

Calculate the average waiting time using Priority Scheduling.

4. Consider the following processes with arrival times and burst times, and the time quantum for Round Robin scheduling is 2 units:

Process	Arrival Time	Burst Time
---------	--------------	------------

|-----|-----|-----|

| P1 | 0 | 4 |

| P2 | 1 | 5 |

| P3 | 2 | 2 |

| P4 | 3 | 3 |

Calculate the average turnaround time using Round Robin scheduling.

5. Consider a program that uses the fork() system call to create a child process. Initially, the parent process has a variable x with a value of 5. After forking, both the parent and child processes increment the value of x by 1. What will be the final values of x in the parent and child processes after the fork() call?

Expt. No. / Name :

Page No. / Name :

1.

Process	Arrival Time	Burst Time	Wait	Completion Time
P <sub>1</sub>	0	5	0	5
P <sub>2</sub>	1	3	4	8
P <sub>3</sub>	2	6	6	14

```

    | P1 | | P2 | | P3 | |
    0   5   8   14
  
```

$$\therefore \text{Average Waiting Time} = \frac{0+4+6}{3} = 3.33$$

2.

Process	Arrival Time	Burst Time	CT	Wait	TAT
P <sub>1</sub>	0	3	3	0	3
P <sub>2</sub>	1	5	13	1	12
P <sub>3</sub>	2	1	4	1	2
P <sub>4</sub>	3	4	8	7	5

```

    | P1 | | P3 | | P4 | | P2 | |
    0   3   4   8   13
  
```

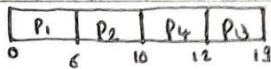
$$\therefore \text{Average TAT} = \frac{3+12+2+5}{4} = 5.5$$

3.

Process	AT	BT	P	CT	W	TAT
P <sub>1</sub>	0	6	3	6	0	6
P <sub>2</sub>	1	4	1	10	5	49
P <sub>3</sub>	2	7	4	13	10	17
P <sub>4</sub>	3	2	2	12	7	9

youva

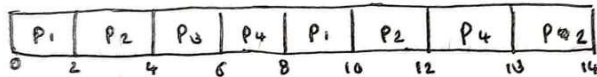
Teacher's Signature



$$\therefore \text{Average Waiting Time} = \frac{0 + 5 + 10 + 7}{4} = 5.5$$

4.

Process	AT	BT	WT	CT	TAT
P <sub>1</sub>	0	4	6	10	10
P <sub>2</sub>	1	5	8	14	13
P <sub>3</sub>	2	2	2	6	4
P <sub>4</sub>	3	3	10	13	10



$$\therefore \text{Average TAT} = \frac{10 + 13 + 4 + 10}{4} = 9.25$$

5.

$$x = 5$$

⇒ 6 (Parent)  
6 (Child)