

CDAC MUMBAI
Concepts of Operating System
Assignment 2
Part A

Name: Mansi Hemant Patil_KH

What will the following commands do?

- **echo "Hello, World!"**

echo - display a line of text.

```
cdac@DESKTOP-I1SKC2V: ~/L × + v
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ echo "Hello, World!"
Hello, World!
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$
```

- **name="Productive"**

name is the variable & productive is the value assign to it
so by echo command we print the value assign in variable.

```
cdac@DESKTOP-I1SKC2V: ~/L × + v
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ name="Productive"
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ echo $name
Productive
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$
```

- **touch file.txt**

Create new empty file in the current directory.

```
cdac@DESKTOP-I1SKC2V: ~/L × + v
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ touch file.txt
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ ls
file.txt
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ |
```

- **ls -a**

list all files in the current directory, including hidden files.

```
cdac@DESKTOP-I1SKC2V: ~/L × + v
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ ls -a
.  ..  file.txt
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ |
```

- **rm file.txt**

Delete the file present in current directory

```
cdac@DESKTOP-I1SKC2V: ~/L × + v
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ ls
file.txt
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ rm file.txt
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ ls
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ |
```

- **cp file1.txt file2.txt**

Copy the content of file1.txt into file2.txt

```
cdac@DESKTOP-I1SKC2V: ~/L × + v
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ touch file1.txt
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cat file1.txt
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ nano file1.txt
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cat file1.txt
hii
hello
OS

cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cp file1.txt file2.txt
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ ls
file1.txt  file2.txt
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cat file2.txt
hii
hello
OS

cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ |
```

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

Used to rename or move a file, here it will moves the file (file.txt) into the specified directory (/path/to/directory/).

- **chmod 755 script.sh**

It is used to assign read, write, and execute permissions to owner, group and other users. The above command gives read, write and execute permissions to the owner and read and execute permissions to group and other users to script.sh file.

- **grep "pattern" file.txt**

Command is used to search for a specific pattern in a file and display the lines that contain that pattern.

```
cdac@DESKTOP-I1SKC2V: ~/L × + v
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cat file.txt
program
pattern program
triangle pattern
square
spaces

cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ grep "pattern" file.txt
pattern program
triangle pattern
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ |
```

- **kill PID**

It will terminate the process whose PID is mentioned. Here, command doesn't contain any process id, above command will result in an error.

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

mkdir- command create new directory (mydir) using &&
cd mydir- went in that directory and file.txt is create then print hello word using echo and append in file.txt and print the output using cat command.

```
cdac@DESKTOP-I1SKC2V: ~/L × + v
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt
Hello, World!
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2/mydir$ ls
file.txt
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2/mydir$ cat file.txt
Hello, World!
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2/mydir$ |
```

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

ls -l : Lists the contents of the current directory.

| : piping is done and grep command print search .txt pattern and display.

```
cdac@DESKTOP-I1SKC2V: ~/L × + v
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2/mydir$ ls -l | grep ".txt"
-rw-r--r-- 1 cdac cdac 14 Feb 28 12:32 file.txt
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2/mydir$ |
```

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

Combine output of file1&2 and then sort it and print unique details present in it.

```
cdac@DESKTOP-I1SKC2V: ~/L × + v
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cat file1.txt
hii
hello
OS

cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cat file2.txt
hii
hello
OS
program
linux
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cat file1.txt file2.txt | sort | uniq
OS
hello
hii
linux
program
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ |
```

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

List out the files and directory present and output is add as input to grep command and grep command display the 'd' letter present details.

```
cdac@DESKTOP-I1SKC2V: ~/L × + v
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ ls -l
total 20
drwxr-xr-x 2 cdac cdac 4096 Feb 28 12:25 directory
-rw-r--r-- 1 cdac cdac 56 Feb 28 12:30 file.txt
-rw-r--r-- 1 cdac cdac 14 Feb 28 12:22 file1.txt
-rw-r--r-- 1 cdac cdac 27 Feb 28 12:42 file2.txt
drwxr-xr-x 2 cdac cdac 4096 Feb 28 12:32 mydir
-rw-r--r-- 1 cdac cdac 0 Feb 28 12:27 script.sh
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ ls -l | grep "^d"
drwxr-xr-x 2 cdac cdac 4096 Feb 28 12:25 directory
drwxr-xr-x 2 cdac cdac 4096 Feb 28 12:32 mydir
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ |
```

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

It is used to recursively search for given pattern “pattern” in the directory /path/to/directory, provided that such directory exists in first place. The output will display the lines containing the “pattern” pattern in it.

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

cat command displays the content of file1.txt followed by file2.txt. sort command is used to perform sort on the result of cat command. uniq -d command is used to display only duplicate lines in the previous output.

- **chmod 644 file.txt**

It is used to assign read and write permissions to owner of the file file.txt and read permission to group users and other users.

- **cp -r source_directory destination_directory**

It is used to copy the source_directory to destination directory. This is done by using -r option so that all files in source_directory are copied recursively.

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

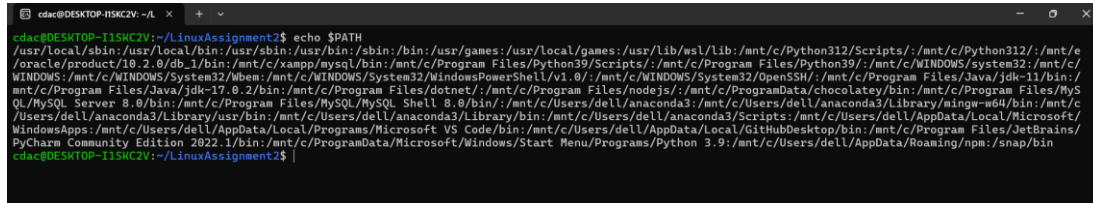
find command is used for searching the files and directories. Given command searches /path/to/search directory and its subdirectories for any file ending with .txt pattern.

- **chmod u+x file.txt**

It is used to give the user execute permissions for file.txt.

- **echo \$PATH**

It display the environment variable that contains a list of directories separated by colons (:).



```
cdac@DESKTOP-HSKC2V:~/LinuxAssignment2$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/lib/mwl/Lib:/mnt/c/Python312/Scripts:/mnt/c/Python312:/mnt/c/oracle/product/10.2.0/db_1/bin:/mnt/c/xampp/mysql/bin:/mnt/c/Program Files/Python39/Scripts:/mnt/c/Program Files/Python39:/mnt/c/WINDOWS/system32:/mnt/c/WINDOWS:/mnt/c/WINDOWS/System32/Wbem:/mnt/c/WINDOWS/System32/WindowsPowerShell/v1.0:/mnt/c/WINDOWS/System32/OpenSSH:/mnt/c/Program Files/Java/jdk-11/bin:/mnt/c/Program Files/Java/jdk-17.0.2/bin:/mnt/c/Program Files/dotnet:/mnt/c/Program Files/nodejs:/mnt/c/ProgramData/chocolatey/bin:/mnt/c/Program Files/MySQL/MySQL Server 8.0/bin:/mnt/c/Program Files/MySQL/MySQL Shell 8.0/bin:/mnt/c/Users/dell/anaconda3:/mnt/c/Users/dell/anaconda3/Library/mingw-w64/bin:/mnt/c/Users/dell/anaconda3/Library/usr/bin:/mnt/c/Users/dell/anaconda3/Library/bin:/mnt/c/Users/dell/anaconda3/Scripts:/mnt/c/Users/dell/AppData/Local/Microsoft/WindowsApps:/mnt/c/Users/dell/AppData/Local/Programs/Microsoft VS Code/bin:/mnt/c/Users/dell/AppData/Local/GitHubDesktop/bin:/mnt/c/Program Files/ JetBrains/PyCharm Community Edition 2022.1/bin:/mnt/c/ProgramData/Microsoft/Windows/Start Menu/Programs/Python 3.9:/mnt/c/Users/dell/AppData/Roaming/npm:/snap/bin
cdac@DESKTOP-HSKC2V:~/LinuxAssignment2$
```

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. It is used to change the directory.
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.**
Incorrect. The correct command is chmod
2. **cpy is used to copy files and directories.**
Incorrect. The correct command is cp
3. **mkfile is used to create a new file.**
Incorrect. The correct command for creating a new file is touch
4. **catx is used to concatenate files.**
Incorrect. The correct command for concatenating files is cat
5. **rn is used to rename files.**
Incorrect. The correct command for renaming files is mv

Part C

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

```
cdac@DESKTOP-I1SKC2V: ~/L × + v
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ nano sh1
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ bash sh1
Hello,World!
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cat sh1
echo "Hello,World!"

cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ |
```

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

```
cdac@DESKTOP-I1SKC2V: ~/L × + v
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ nano sh2
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ bash sh2
CDAC MUMBAI
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cat sh2
name="CDAC MUMBAI"
echo $name
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ |
```

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

```
cdac@DESKTOP-I1SKC2V: ~/L × + v
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ nano sh3
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ bash sh3
Enter the number
20
Number is 20
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cat sh3
echo Enter the number
read Number
echo Number is $Number

cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ |
```

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

```
cdac@DESKTOP-I1SKC2V: ~/L × + v
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ nano sh4
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ bash sh4
Enter a Number
5
Enter a Number
3
Sum of 5 and 3 is 8
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cat sh4
echo Enter a Number
read Num1
echo Enter a Number
read Num2
Sum=$((Num1 + Num2))
echo Sum of $Num1 and $Num2 is $Sum
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ |
```

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

```
cdac@DESKTOP-I1SKC2V: ~/L × + v
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ nano sh5
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ bash sh5
Enter the Number
2
number is even
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cat sh5
echo Enter the Number
read Number
if [ $Number/2==0 ]
then
    echo number is even
else
    echo number is odd
fi
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ |
```


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

```
cdac@DESKTOP-I1SKC2V: ~/L × + ∨  
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ nano sh6  
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ bash sh6  
Enter the number:  
5  
for loop to print numbers from 1 to 5  
1  
2  
3  
4  
5  
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cat sh6  
echo Enter the number:  
read n  
echo for loop to print numbers from 1 to 5  
for((i=1;i<=n;i++))  
do  
    echo $i  
done  
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ |
```

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

```
cdac@DESKTOP-I1SKC2V: ~/L × + ∨  
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ nano sh7  
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ bash sh7  
Enter the number:  
1  
while loop to print numbers from 1 to 5  
1  
2  
3  
4  
5  
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cat sh7  
echo Enter the number:  
read i  
echo while loop to print numbers from 1 to 5  
while((i<=5))  
do  
    echo $i  
    i=$((i+1))  
done  
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ |
```

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-I1SKC2V: ~/L × + ∨  
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ nano sh8  
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ bash sh8  
file exist  
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cat sh8  
if [ -e "file.txt" ]  
then  
    echo file exist  
else  
    echo file does not exist  
fi  
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ |
```

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-I1SKC2V: ~/L × + ∨  
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ nano sh9  
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ bash sh9  
Enter the number  
5  
number is less than 10  
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cat sh9  
echo Enter the number  
read n  
if [ $n -gt 10 ]  
then  
    echo number is greater than 10  
else  
    echo number is less than 10  
fi  
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ |
```

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-I1SKC2V: ~/L × + v
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ nano sh10
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ bash sh10
1 2 3 4 5
2 4 6 8 10
3 6 9 12 15
4 8 12 16 20
5 10 15 20 25
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cat sh10
for i in {1..5}
do
    for j in {1..5}
    do
        echo -n "${i * j}) "
    done
    echo
done
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ |
```

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-I1SKC2V: ~/L × + v
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ nano sh11
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ bash sh11
Enter the number
5
Square of number is 25
Enter the number
-2
negative number exit loop
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ cat sh11
while true
do
    echo Enter the number
    read number
    if [ $number -lt 0 ]
    then
        echo negative number exit loop
        break
    fi
    square=$((number * number))
    echo Square of number is $square
done
cdac@DESKTOP-I1SKC2V:~/LinuxAssignment2$ |
```

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.

Q.1]

Process	Arrival Time	Burst Time	Completion Time	TAT	Waiting Time
P1	0	5	5	5	0
P2	1	3	8	7	4
P3	2	6	14	12	6

Gantt Chart

P1	P2	P3	
0	5	8	14

Average waiting Time = $(0+4+6)/3 = 10/3 = 3.3333$
 ≈ 3.33

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.

Q.2]

Process	Arrival Time	Burst Time	Waiting Time	Turnaround Time
P1	0	3	0	3
P2	1	5	7	12
P3	2	1	1	2
P4	3	4	1	5

Gantt Chart

P1	P3	P4	P2	
0	3	4	8	13

Average Turnaround Time = $\frac{3+12+2+5}{4} = \frac{22}{4} = 5.5$

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.

Q.3]

Process	Arrival Time	Burst Time	Priority	Waiting Time
P ₁	0	6	3	0
P ₂	1	4	1	5
P ₃	2	7	4	10
P ₄	3	2	2	7

Gantt Chart

P ₁	P ₂	P ₄	P ₃	
0	6	10	12	19

Average waiting Time = $\frac{0+5+10+7}{4} = \frac{22}{4} = \underline{\underline{5.5}}$

Gantt Chart (Preemptive)

P ₁	P ₂	P ₄	P ₁	P ₃	
0	1	5	7	12	19

waiting Time

P ₁	6
P ₂	0
P ₃	2
P ₄	10

Average waiting Time = $\frac{18}{4} = \underline{\underline{4.5}}$

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.

Q.4]

Process	Arrival Time	Burst Time	Waiting Time	Turnaround Time
P ₁	0	4	6	10
P ₂	1	5	8	13
P ₃	2	2	2	4
P ₄	3	3	7	10

Gantt Chart

P ₁	P ₂	P ₃	P ₄	P ₁	P ₂	P ₄	P ₂	
0	2	4	6	8	10	12	13	14

Average Turnaround Time

$$= \frac{10+13+4+10}{4} = \frac{37}{4} = \underline{\underline{9.25}}$$

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?



When the fork() system call is used, it creates a child process that is exact replica of current process.

Before forking, the parent process has a variable x value 5. After the fork, both the parent and child have x = 5.

Each process then increments x by 1, so both the parent and child have x = 6.

The final values of x in the parent and child processes after the fork() call is parent process x=6 and in child process x=6.