Concepts of Operating System

Assignment 2

Part A:

What will the following commands do?

• echo "Hello, World!" 🡺 echo is a shell command used to **display text/strings or variables** on the screen.

• name="Productive" 🡺It **creates a variable** called name and assigns it the value **Productive**.

• touch file.txt 🡺 If **file.txt does not exist** → it creates an empty file named file.txt.

If **file.txt already exists** → it updates the file’s **last modified timestamp** to the current time (without changing its content).

• ls -a 🡺 It lists **all files and directories** in the current directory, **including hidden ones**.

• rm file.txt🡺 It **deletes (removes)** the file named file.txt from the current directory.

• cp file1.txt file2.txt 🡺 It **copies** the contents of file1.txt into a new file named file2.txt.

If file2.txt does **not exist** → it creates it with the same content as file1.txt.

If file2.txt **already exists** → its contents are **overwritten** with those of file1.txt

• mv file.txt /path/to/directory/ 🡺 It **moves** the file file.txt into the specified directory (/path/to/directory/).

• chmod 755 script.sh 🡺 It **changes the permissions** of the file script.sh.

• grep "pattern" file.txt 🡺 It **searches** for the given text (pattern) inside file.txt and prints all lines that **contain** that pattern.

• kill PID 🡺It sends a **signal** to the process with the given **Process ID (PID)**.  
By default, it sends the **SIGTERM (15)** signal, which **politely asks the process to terminate**.

• mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt 🡺The && ensures that **each command runs only if the previous one succeeds**.  
So if mkdir mydir fails (say, the directory already exists and you don’t have permission), none of the following commands will run.

• ls -l | grep ".txt" 🡺 ls -l → lists files in **long format** (permissions, size, date, name).

| → pipes the output of ls -l into the next command.

grep ".txt" → filters the output, showing only lines with .txt (i.e., text files).

• cat file1.txt file2.txt | sort | uniq 🡺 **cat file1.txt file2.txt** → prints the contents of both files one after another

**sort** → arranges all the lines alphabetically (or numerically if they are numbers).

**uniq** → removes duplicate **consecutive** lines (that’s why sorting first is important)

• ls -l | grep "^d" 🡺  ls -l → lists files in **long format** (permissions, owner, size, date, name).

| → pipes the output to grep.

grep "^d" → filters lines that **start with d**.

• grep -r "pattern" /path/to/directory/ 🡺 **Searches recursively** in all files under /path/to/directory/ for lines containing "pattern"

Prints file name + matching line.

• cat file1.txt file2.txt | sort | uniq –d 🡺 cat → combine contents of file1.txt and file2.txt.

sort → sort lines.

uniq -d → show only **duplicate lines** (that appear in both files).

• chmod 644 file.txt 🡺 Changes permissions of file.txt

• cp -r source\_directory destination\_directory 🡺 **Copies a directory** (source\_directory) and its contents recursively into destination\_directory.

• find /path/to/search -name "\*.txt" 🡺 Searches recursively in /path/to/search for files with names ending in .txt.

• chmod u+x file.txt 🡺 Adds **execute (x) permission** for the **user (owner)** only.

Makes file.txt executable (e.g., if it’s a script).

• echo $PATH 🡺 Prints the value of the **PATH environment variable**.

PATH tells the shell where to look for executables when you type a command.

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. 🡺INCORRECT

2. cpy is used to copy files and directories. 🡺INCORRECT

3. mkfile is used to create a new file. 🡺INCORRECT

4. catx is used to concatenate files. 🡺INCORRECT

5. rn is used to rename files. 🡺INCORRECT

Part C Question

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

🡺cdac@DESKTOP-U0PFJ6O:~$ vi ass.sh

cdac@DESKTOP-U0PFJ6O:~$ chmod +x ass.sh

cdac@DESKTOP-U0PFJ6O:~$ ./ass.sh

Hello, World!

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

🡺cdac@DESKTOP-U0PFJ6O:~$ cat ass.sh

#!/bin/bash

name="CDAC Mumbai"

echo $name

cdac@DESKTOP-U0PFJ6O:~$ ./ass.sh

CDAC Mumbai

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

🡺cdac@DESKTOP-U0PFJ6O:~$ cat ass.sh

#!/bin/bash

echo "Enter a number="

read num

echo $num

cdac@DESKTOP-U0PFJ6O:~$ ./ass.sh

Enter a number=

10

10

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

🡺cdac@DESKTOP-U0PFJ6O:~$ cat ass.sh

#!/bin/bash

num1=5

num2=3

sum=$((num1+num2))

echo "The sum of $num1 and $num2 is:$sum"

cdac@DESKTOP-U0PFJ6O:~$ ./ass.sh

The sum of 5 and 3 is:8

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

🡺cdac@DESKTOP-U0PFJ6O:~$ cat ass.sh

#!/bin/bash

echo "Enter a number="

read num

if (($num%2==0)); then

echo "Even"

else

echo "Odd"

fi

cdac@DESKTOP-U0PFJ6O:~$ ./ass.sh

Enter a number=

10

Even

cdac@DESKTOP-U0PFJ6O:~$ ./ass.sh

Enter a number=

9

Odd

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

🡺cdac@DESKTOP-U0PFJ6O:~$ ./ass.sh

1

2

3

4

5

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

🡺cdac@DESKTOP-U0PFJ6O:~$ cat ass.sh

#!/bin/bash

i=1

while [ $i -le 5 ]

do

echo "$i"

i=$((i+1))

done

cdac@DESKTOP-U0PFJ6O:~$ ./ass.sh

1

2

3

4

5

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-U0PFJ6O:~$ cat ass.sh

#!/bin/bash

num=12

if [ $num -gt 10 ]

then

echo "The number $num is greater than 10."

else

echo "The number $num is not greater than 10."

fi

cdac@DESKTOP-U0PFJ6O:~$ ./ass.sh

The number 12 is greater than 10.