

Assignment 2

Concepts of Operating System

Name: Mukta Wagh_KH

PartA

What will the following commands do?

1) echo "Hello, World!"

Ans:

It will print the given string i.e. "Hello, World!" On the terminal.

2) name="Productive"

Ans:

Name is the variable, productive is a value, and this is basically assigning value to variable.

3) touch file.txt

Ans:

It will create a new file name as file.txt.

4) ls -a

Ans:

Lists all files including hidden files.

5) `rm file.txt`

Ans:

Remove the given file.

6) `cp file1.txt file2.txt`

Ans:

Copies file1.txt to file2.txt

7) `mv file.txt /path/to/directory/`

Ans:

Move the file file.txt to the given directory.

8) `chmod 755 script.sh`

Ans:

Change the mode meaning give or deny the permission of read write and execution to the owner, group and other users. 7 meaning give all permission to the owner like read write and execution ,5 meaning only permission of read and execution to the group and other users.

9) `grep "pattern" file.txt`

Ans:

Search for the word "pattern" in file.txt.

10) kill PID

Ans:

Used to terminate the process using process ID.

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

Ans:

Here we use shell meta character && and it verify that each step is executes then and only if the previous one succeeds.

Mkdir mydir: it creates a directory name as mydir

Cd mydir: change directory to mydir

Touch file.txt: creates an empty file file.txt.

Echo hello world: print the given string

>: Redirection ,It allow you to use output of one command as an input of other command

Cat file.txt: displays the content of file.txt.

Overall, it creates a directory, move it, creates a file, writes text into it, and then displays its contents.

12) ls -l | grep ".txt"

Ans:

Ls -l: lists the file and directory in details in current directory.

13) `cat file1.txt file2.txt | sort | uniq`

Ans:

Cat: combine these two files

| sort | uniq: piping sort these files meaning sort the combine content and remove duplicates

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

Ans:

`grep "^d"`: filters the output of `ls -l` and matches only lines where the first char is d.

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

Ans:

`grep -r`: The `-r` tells `grep` to search recursively.

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

Ans:

`uniq -d`: The `-d` with `uniq` shows only the lines that appear more than once in the sorted output.

17) `chmod 644 file.txt`

Ans:

It will change the permission of file.txt so that owner can read and write and group and others can read only.

18) `cp -r source_directory destination_directory`

Ans:

Copy the entire directory and its content into the given destination_directory.

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

Ans:

Find: it searches the file in the directory

Overall, it displays a list of all .txt files located within the specified path

20) `chmod u+x file.txt`

Ans:

u+x : u refer to user and x refer to execute so; the file owner gives permission on file.txt making it executable.

21) `echo $PATH`

Ans:

\$PATH: The environment variable that contains a list of directories that are searched for executables files and displays it.

Part B

Identify True or False:

1. ls is used to list files and directories in a directory.

Ans: TRUE

2. mv is used to move files and directories.

Ans: TRUE

3. cd is used to copy files and directories.

Ans: FALSE

4. pwd stands for "print working directory" and displays the current directory.

Ans: TRUE

5. grep is used to search for patterns in files.

Ans: TRUE

6. chmod 755 file.txt gives read, write, and execute permissions to the owner, and read and execute permissions to group and others.

Ans: TRUE

7. `mkdir -p directory1/directory2` creates nested directories, creating `directory2` inside `directory1` if `directory1` does not exist.

Ans: TRUE

8. `rm -rf file.txt` deletes a file forcefully without confirmation

Ans: TRUE

Identify the Incorrect Commands:

1. `chmodx` is used to change file permissions.
2. `cpy` is used to copy files and directories.
3. `mkfile` is used to create a new file.
4. `catx` is used to concatenate files.
5. `rn` is used to rename files.

ANS: all are partially or completely incorrect

[Chmod not chmodx, cp not cpy, touch not mkfile, cat not catx , mv not rn]

PartC


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

```
cdac@DESKTOP-5UQ52PL: ~  
cdac@DESKTOP-5UQ52PL:~$ echo " Hello, World! "  
Hello, World!  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ echo " Hello, World! "  
Hello, World!  
cdac@DESKTOP-5UQ52PL:~$ nano abc.txt  
cdac@DESKTOP-5UQ52PL:~$ cat abc.txt  
Hello, World!  
cdac@DESKTOP-5UQ52PL:~$ █
```

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

```
cdac@DESKTOP-5UQ52PL: ~  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ name="CDAC Mumbai"  
cdac@DESKTOP-5UQ52PL:~$ echo $name  
CDAC Mumbai  
cdac@DESKTOP-5UQ52PL:~$ nano printname.txt  
cdac@DESKTOP-5UQ52PL:~$ cat printname.txt  
CDAC Mumbai  
cdac@DESKTOP-5UQ52PL:~$ █
```


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

 cdac@DESKTOP-5UQ52PL: ~

```
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ read -p "Enter a number: " number  
Enter a number: 11234 345 567 123  
cdac@DESKTOP-5UQ52PL:~$ echo "You entered : $number"  
You entered : 11234 345 567 123  
cdac@DESKTOP-5UQ52PL:~$
```

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

```
cdac@DESKTOP-5UQ52PL: ~  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ num1=20  
cdac@DESKTOP-5UQ52PL:~$ num2=30  
cdac@DESKTOP-5UQ52PL:~$ sum=$((num1+num2))  
cdac@DESKTOP-5UQ52PL:~$ echo "sum of $num1 and $num2 is : $sum"  
sum of 20 and 30 is : 50  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ nano summ.txt  
cdac@DESKTOP-5UQ52PL:~$ cat summ.txt  
num1=20  
num2=30  
sum=$((num1+num2))  
echo "sum of $num1 and $num2 is : $sum"  
cdac@DESKTOP-5UQ52PL:~$ bash summ.txt  
sum of 20 and 30 is : 50  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ read -p "enter the value of num1: " num1  
enter the value of num1: 20  
cdac@DESKTOP-5UQ52PL:~$ read -p "enter the value of num2: " num2  
enter the value of num2: 30  
cdac@DESKTOP-5UQ52PL:~$ sum=$((num1+num2))  
cdac@DESKTOP-5UQ52PL:~$ echo "the sum of $num1 and $num2 is : $sum"  
the sum of 20 and 30 is : 50  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ nano user.txt  
cdac@DESKTOP-5UQ52PL:~$ cat user.txt  
read -p "enter the value of num1: " num1  
read -p "enter the value of num2: " num2  
sum=$((num1+num2))  
echo "the sum of $num1 and $num2 is : $sum"  
cdac@DESKTOP-5UQ52PL:~$ bash user.txt  
enter the value of num1: 20  
enter the value of num2: 30  
the sum of 20 and 30 is : 50  
cdac@DESKTOP-5UQ52PL:~$
```

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

```
cdac@DESKTOP-5UQ52PL: ~  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ nano evenodd.txt  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ cat evenodd.txt  
read -p "enter the num: " num  
if ((num % 2 == 0)); then  
    echo "even"  
else  
    echo "odd"  
fi  
cdac@DESKTOP-5UQ52PL:~$ bash evenodd.txt  
enter the num: 23  
odd  
cdac@DESKTOP-5UQ52PL:~$ bash evenodd.txt  
enter the num: 45  
odd  
cdac@DESKTOP-5UQ52PL:~$ bash evenodd.txt  
enter the num: 1947645636  
even  
cdac@DESKTOP-5UQ52PL:~$
```

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

```
cdac@DESKTOP-5UQ52PL: ~  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ nano forloop.txt  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ cat forloop.txt  
a=0  
for a in 1 2 3 4 5  
do  
echo $a  
done  
cdac@DESKTOP-5UQ52PL:~$ bash forloop.txt  
1  
2  
3  
4  
5  
cdac@DESKTOP-5UQ52PL:~$
```

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

```
cdac@DESKTOP-5UQ52PL: ~  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ nano mno.txt  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ cat mno.txt  
num=1  
  
while [ $num -le 5 ]  
do  
    echo $num  
    ((num++))  
done  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ bash mno.txt  
1  
2  
3  
4  
5  
cdac@DESKTOP-5UQ52PL:~$
```

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-5UQ52PL: ~  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ nano abc.txt  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ cat abc.txt  
if [ -f "file.txt" ]  
then  
    echo "file exists"  
else  
    echo "file does not exist"  
fi  
cdac@DESKTOP-5UQ52PL:~$ bash abc.txt  
file exists  
cdac@DESKTOP-5UQ52PL:~$
```

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-5UQ52PL: ~  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ nano xyz.txt  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ cat xyz.txt  
echo "enter a number: "  
read num  
if [ "$num" -gt 10 ]  
then  
    echo "The number is greater than 10"  
else  
    echo "The number is not greater than 10"  
fi  
cdac@DESKTOP-5UQ52PL:~$ bash xyz.txt  
enter a number:  
13  
The number is greater than 10  
cdac@DESKTOP-5UQ52PL:~$ bash xyz.txt  
enter a number:  
3  
The number is not greater than 10  
cdac@DESKTOP-5UQ52PL:~$
```

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-5UQ52PL: ~  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ cat abc.txt  
echo "Multiplication Table (1 to 5)"  
  
echo -ne "  |"  
for i in {1..5}; do  
    echo -ne " $i "  
done  
echo -e "\n-----"  
  
for i in {1..5}; do  
    echo -ne "$i |"  
    for j in {1..5}; do  
        result=$((i * j))  
        printf "%3d " $result  
    done  
    echo ""  
done  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ bash abc.txt  
Multiplication Table (1 to 5)  
  | 1  2  3  4  5  
-----  
1 | 1  2  3  4  5  
2 | 2  4  6  8 10  
3 | 3  6  9 12 15  
4 | 4  8 12 16 20  
5 | 5 10 15 20 25  
cdac@DESKTOP-5UQ52PL:~$
```

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-5UQ52PL: ~  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ nano jkl.txt  
cdac@DESKTOP-5UQ52PL:~$  
cdac@DESKTOP-5UQ52PL:~$ cat jkl.txt  
while true; do  
    echo "Enter a number: "  
    read num  
    if [ "$num" -lt 0 ]  
    then  
        echo "Negative number entered. Exiting..."  
        break  
    fi  
  
    square=$((num* num))  
    echo "Square of $num is $square"  
done  
cdac@DESKTOP-5UQ52PL:~$ bash jkl.txt  
Enter a number:  
4  
Square of 4 is 16  
Enter a number:  
7  
Square of 7 is 49  
Enter a number:  
-23  
Negative number entered. Exiting...  
cdac@DESKTOP-5UQ52PL:~$
```