CDAC MUMBAI

Concepts of Operating System

Assignment 2

Part A

What will the following commands do?

• echo "Hello, World!"

Shell command these is will print/ display Hello, World!

• name="Productive"

These will assign string Productive to variable name

• touch file.txt

These command will create file.txt

• ls -a

List all files and directories in the current directory, including hidden files and directories.

• rm file.txt

Remove file.txt from current directory is present

• cp file1.txt file2.txt

Copy file1 content to file2 if present if no present the file2 will be created with

• mv file.txt /path/to/directory/

Move file.txt to destination/path provided

• chmod 755 script.sh

Change the permissions of the file named script.sh the owner has read, write, and execute permissions, the group has read and execute and others have read and execute permissions but no write permission.

• grep "pattern" file.txt

This command will search for the string "pattern" in the file named file.txt

• kill PID

This command kill the process or terminate the process with the specified process ID

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

performs a series of operations to create a directory, navigate into it, create a file, write content to the file, and then display the file's content.

• ls -l | grep ".txt"

list all files in the current directory that have a .txt extension

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

These command the contents of file1.txt and file2.txt, and then displays only the unique lines from the combined content

• ls -l | grep "^d"

Filters the output to only include lines that start with d. In the context of ls -l output, lines that start with d indicate directories.

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

searches for a specific pattern recursively within a directory and its subdirectories

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

the contents of file1.txt and file2.txt to identify and display duplicate lines that appear in the combined content

• chmod 644 file.txt

changes the permissions of file.txt the owner can read and write the file, the group can only read the file and others can only read the file.

• cp -r source\_directory destination\_directory

These command copy a directory and its entire contents, including all files and subdirectories

• find /path/to/search -name "\*.txt"

search for files with a .txt extension within a specified directory and its subdirectories.

• chmod u+x file.txt • echo $PATH

change the file permissions of file.txt, specifically granting the file's owner the permission to execute the file.

**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 Is correct command

2. cpy is used to copy files and directories.

cp source destination is correct

3. mkfile is used to create a new file.

nano filename

touch filename

these is correct command to create file

4. catx is used to concatenate files.

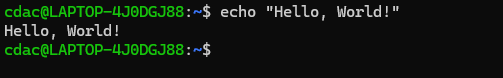
cat filename is correct command

5. rn is used to rename files.

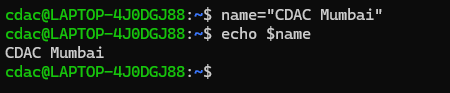
mv oldname newname these command can be use for rename

Part C

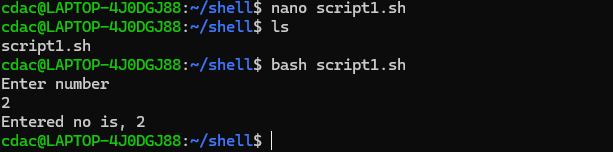
Question 1: Write a shell script that prints "Hello, World!" to the terminal.



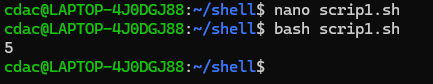
Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.

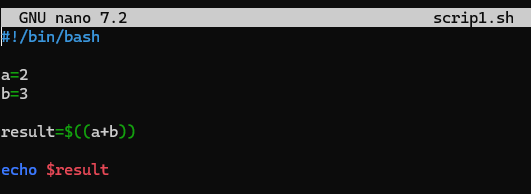


Question 3: Write a shell script that takes a number as input from the user and prints it.

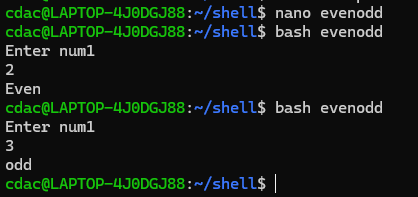


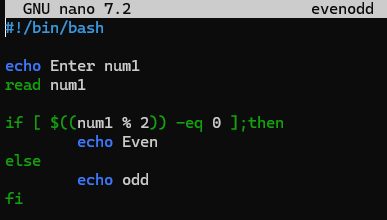
Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.



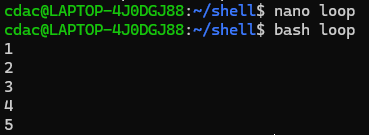


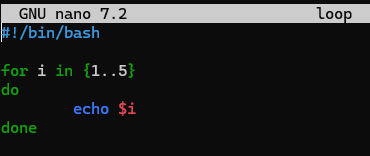
Question 5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".



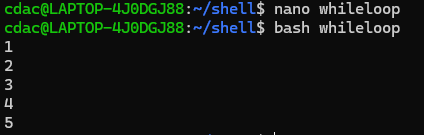


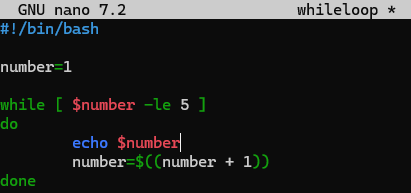
Question 6: Write a shell script that uses a for loop to print numbers from 1 to 5.



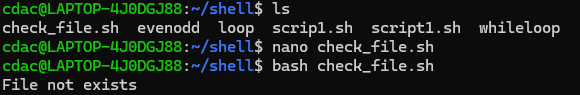


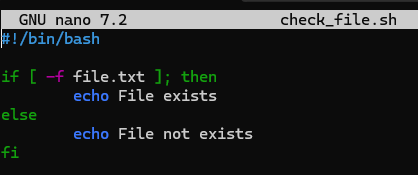
Question 7: Write a shell script that uses a while loop to print numbers from 1 to 5.



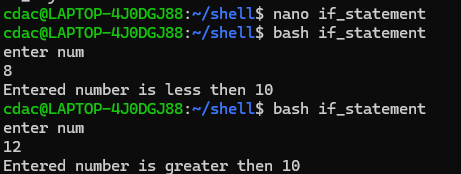


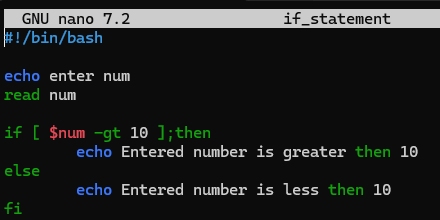
Question 8: Write a shell script that checks if a file named "file.txt" exists in the current directory. If it does, print "File exists", otherwise, print "File does not exist".



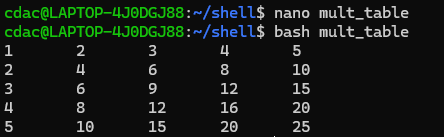


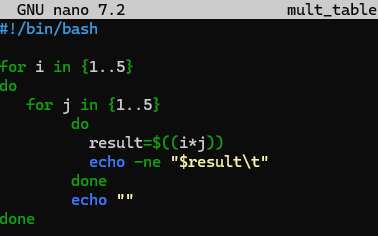
Question 9: Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.



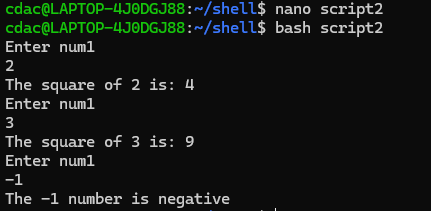


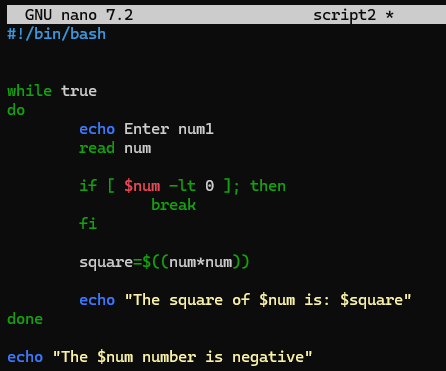
Question 10: Write a shell script that uses nested for loops to print a multiplication table for numbers from 1 to 5. The output should be formatted nicely, with each row representing a number and each column representing the multiplication result for that number.





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.





Part E

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

| Process | Arrival Time | Burst Time |

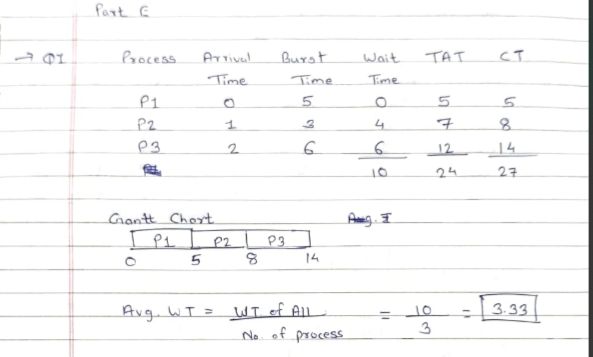
|---------|--------------|------------|

| P1 |0 |5 |

| P2 |1 |3 |

| P1 |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 |

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

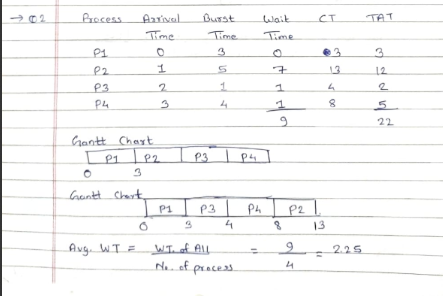
| P1 |0 | 3 |

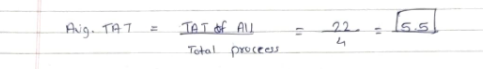
| P2 | 1 | 5 |

| 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 |

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

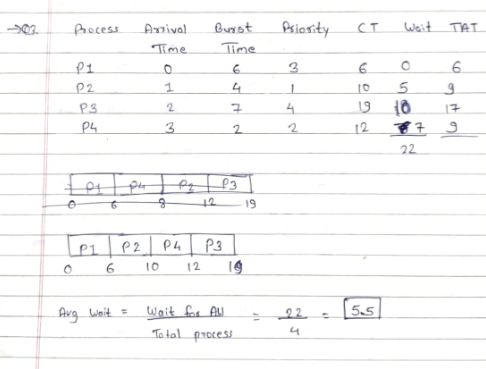
| P1 | 0 | 6 | 3 |

| 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 |

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

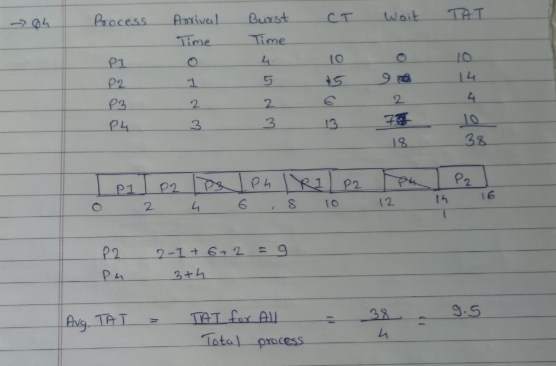
| P1 | 0 | 4 |

| 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?

