Concepts of Operating System

Assignment 2

**Part A**

**What will the following commands do?**

echo "Hello, World!" ---print Hello World on console

.name="Productive" --- assign variable name with “Productive” word

.touch file.txt --- create an empty file

.ls -a ---- list all the directory and files of the cuurent directory along with the hidden ones.

.rm file.txt --- delete the file

.cp file1.txt file2.txt --- overwrites file2 with file1 content

.mv file.txt /path/to/directory/ --- move file to the mentioned directory

.chmod 755 script.sh – give the owner of script read , write and execute permissions and give group users and others only read and write permission

.grep "pattern" file.txt ---- search and print lines of file containing pattern

.kill PID – abort the process associated with process id PID

.mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt --- first ,create a directory called mydir .second, navigate to that directory.third, create a file in same directory .fourth, write result of echo to file created and fifith, display the content of file.

.ls -l | grep ".txt" ---- lists the files that have “.txt” extension present in the current directory

.cat file1.txt file2.txt | sort | uniq --- append file1 with file2 content and sort the content in asceding order and finally filter out adjacent duplicate lines, outputting only unique lines from the sorted input. If there are multiple identical lines, only one instance of the line is kept in the output.

.ls -l | grep "^d" – list only directories in the current directory

.grep -r "pattern" /path/to/directory/ --- search recursively all files and subdirectories within a specified directory for the given pattern. Prints each line that contains the pattern.

.cat file1.txt file2.txt | sort | uniq –d --- concatenate file1 with file2 and print only duplicate lines.

.chmod 644 file.txt --- give owner of file with read and write permission and groups and others with only read permission.

.cp -r source\_directory destination\_directory --- copy the entire source directory content to destination directory

.find /path/to/search -name "\*.txt" ---- search for all files with the .txt extension within the specified directory and its subdirectories.

.chmod u+x file.txt --- give owner execute permission for file

.echo $PATH — print PATH environment variable value of the system

**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(Correct -chmod)

2.

cpy is used to copy files and directories. -- Incorrect(Correct -cp)

3.

mkfile is used to create a new file. – Incorrect (correct- nano,touch)

4.

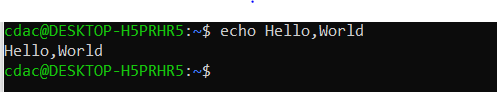
catx is used to concatenate files. ---Incorrect(correct- cat)

5.

rn is used to rename files.--- Incorrect(correct- mv )

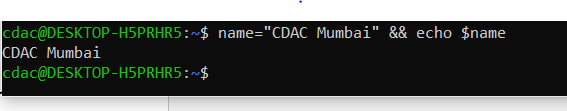
**Part C**

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

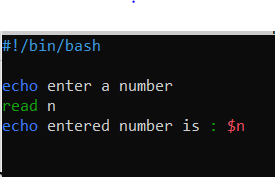


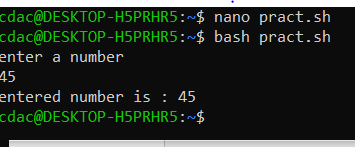
Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the

value of the variable.



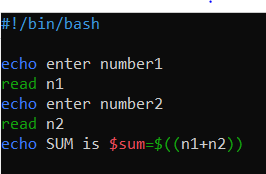
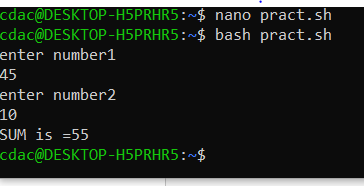
Question 3: Write a shell script that takes a number as input from the user and prints it.





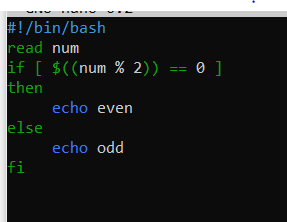
Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the

result.

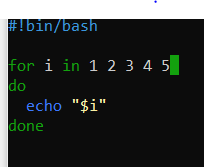
 

Question 5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise

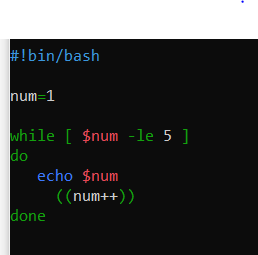
prints "Odd".



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

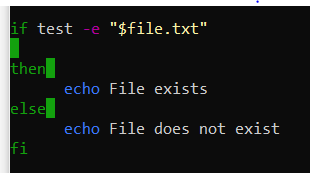


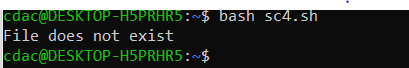
Question 7: Write a shell script that uses a while loop to print numbers from 1 to 5.



Question 8: Write a shell script that checks if a file named "file.txt" exists in the current directory. If it

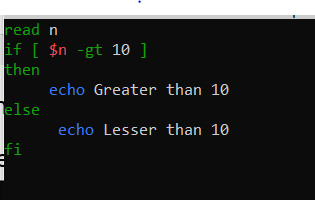
does, print "File exists", otherwise, print "File does not exist".

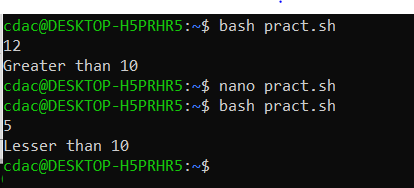




Question 9: Write a shell script that uses the if statement to check if a number is greater than 10 and

prints a message accordingly.

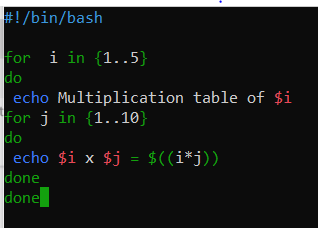




Question 10: Write a shell script that uses nested for loops to print a multiplication table for numbers

from 1 to 5. The output should be formatted nicely, with each row representing a number and each

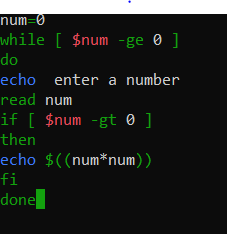
column representing the multiplication result for that number.

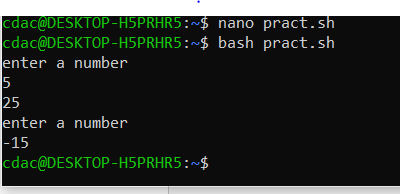


Question 11: Write a shell script that uses a while loop to read numbers from the user until the user enters

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 |

| P3 | 2 | 6 |

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

Solution—

|  |  |  |
| --- | --- | --- |
| P1 | P2 | P3 |

0 5 8 14

Waiting time=TAT – burst time

P1= 0-0=0

P2=5-1=4

P3=8-2=6

Avg. WT= (0+4+6)/3= 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.

Solution—

|  |  |  |  |
| --- | --- | --- | --- |
| P1 | P3 | P4 | P2 |

0 3 4 8 13

TAT= completion time- arrival time

P1= 3-0=3

P2=13-1=12

P3=4-2=2

P4=8-3=5

Avg. TAT= (3+12+2+5)/4=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.

Solution---

Using preemptive Priority scheduling

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P1 | P2 | P4 | P3 |  |

0 6 10 12 19

WT:

P1= 0-0

P2=6-1=5

P3=12-2=10

P4=10-3=7

Avg WT=5.5

4. Consider the following processeswith 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.

Solution ---

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P1 | P2 | P4 | P2 |

0 2 4 6 8 10 12 13 14

TAT:

P1= 10-0=10

P2=14-1=13

P3=6-2=4

P4=13-3= 10

Avg TAT= (10+13+4+10)/4= 37/4= 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?

Solution ---

the final values of x in the parent and child processes after the fork() call will be 6.