Assignment-2

Thursday, August 21, 2025 12:28 AM

CDAC MUMBAI

Concepts of Operating System Assignment 2

Part A

What will the following commands do?

- echo "Hello, World!"
- ➢ Prints The Characters inside Double Quote.

- name="Productive"
- ➢ Assigns Productive to the variable name.

- touch file.txt
- ➢ Create’s a File file.txt.

- ls -a
- ➢ Lists

- rm file.txt
- ➢ Removes the file.txt from the desired location/directory

- cp file1.txt file2.txt
- ➢ Copies the file content from File1.txt to File2.txt.

- mv file.txt /path/to/directory/
- ➢ Moves File1.txt to the desired directory of our choice.

- chmod 755 script.sh
- ➢ Assigns the rights of rwx to owner, rx to group and others

- grep "pattern" file.txt
- ➢ Finds the word pattern inside the File File.txt.

- kill PID
- ➢ PID stands for "PROCESSID", what it does is that it actually Kills the Process which is running by its ProcessId.

- mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt
- ➢ Create a Directory named "mydir" and then by using the cd command we move into the Directory after using the "&&" we create a File named File.txt with the contents of "Hello, World" and using the cat command we display the content of that file.

- ls -l | grep ".txt"
- ➢ This command shows the permissions invoked for particular indicviduals for the files with extension ".txt".
- ➢ ls -l:- List's all the File along with the permission's invoked to particular Individual.
- ➢ | :- This is called as the Pipe which is used to join two commands or to initiate two commands simultaneously.
- ➢ grep ".txt" :- grep command actually finds a particular keyword in our case we are trying to find files with extension ".txt"

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.
   ➢ .

8. **rm -rf file.txt** deletes a file forcefully without confirmation.
   ➢ True

<u>Identify the Incorrect Commands</u>:

- **chmodx** is used to change file permissions.

➢ **Correct Command is:- chmod; in the order of [OWNER-GROUP-OTHER]**

- **cpy** is used to copy files and directories.

➢ **Correct Command is:- cp**

- **mkfile** is used to create a new file.

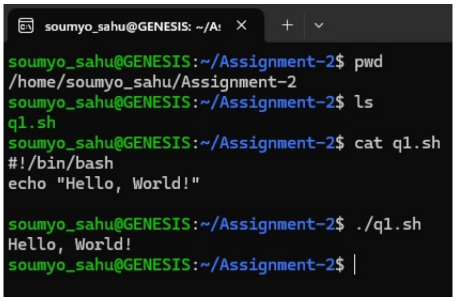
➢ **Correct Command is:- touch**

- **catx** is used to concatenate files.
- **Correct Command is:-"cat >>" it appends or it updates, Concatenate (combine files)**

- **rn** is used to rename files.
- **Correct Command is:- mv to rename**

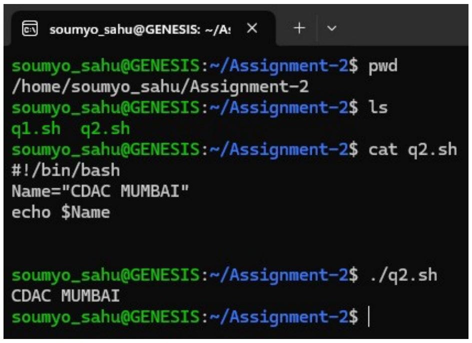
Part C

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



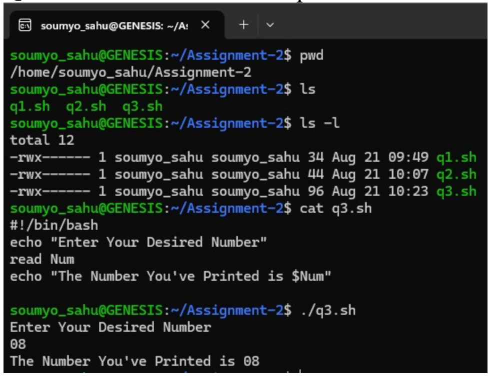
```
soumyo_sahu@GENESIS:~/Assignment-2$ pwd
/home/soumyo_sahu/Assignment-2
soumyo_sahu@GENESIS:~/Assignment-2$ ls
q1.sh
soumyo_sahu@GENESIS:~/Assignment-2$ cat q1.sh
#!/bin/bash
echo "Hello, World!"

soumyo_sahu@GENESIS:~/Assignment-2$ ./q1.sh
Hello, World!
soumyo_sahu@GENESIS:~/Assignment-2$ |
```

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



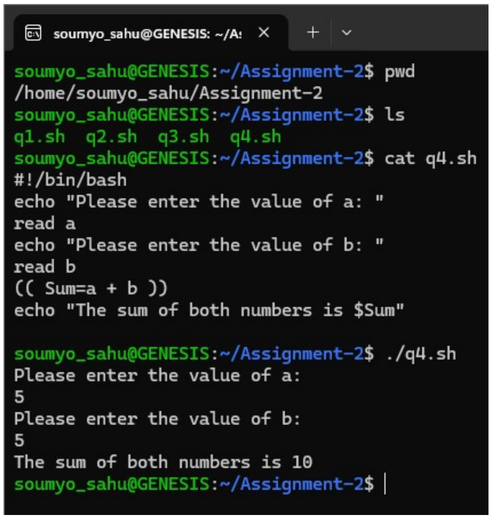
```
soumyo_sahu@GENESIS:~/Assignment-2$ pwd
/home/soumyo_sahu/Assignment-2
soumyo_sahu@GENESIS:~/Assignment-2$ ls
q1.sh  q2.sh
soumyo_sahu@GENESIS:~/Assignment-2$ cat q2.sh
#!/bin/bash
Name="CDAC MUMBAI"
echo $Name


soumyo_sahu@GENESIS:~/Assignment-2$ ./q2.sh
CDAC MUMBAI
soumyo_sahu@GENESIS:~/Assignment-2$ 
```

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



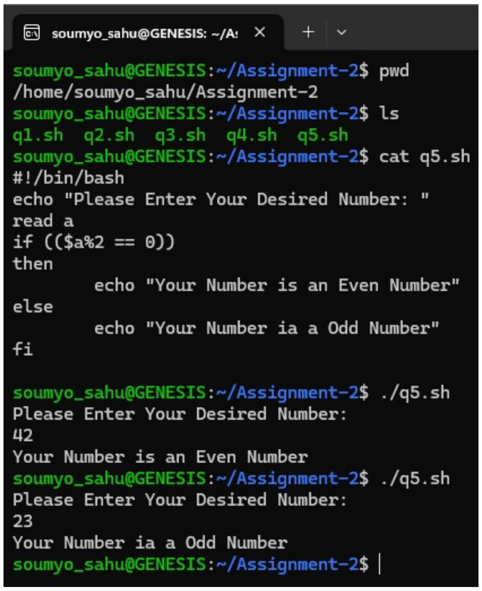
```
soumyo_sahu@GENESIS:~/Assignment-2$ pwd
/home/soumyo_sahu/Assignment-2
soumyo_sahu@GENESIS:~/Assignment-2$ ls
q1.sh  q2.sh  q3.sh
soumyo_sahu@GENESIS:~/Assignment-2$ ls -l
total 12
-rwx------ 1 soumyo_sahu soumyo_sahu 34 Aug 21 09:49 q1.sh
-rwx------ 1 soumyo_sahu soumyo_sahu 44 Aug 21 10:07 q2.sh
-rwx------ 1 soumyo_sahu soumyo_sahu 96 Aug 21 10:23 q3.sh
soumyo_sahu@GENESIS:~/Assignment-2$ cat q3.sh
#!/bin/bash
echo "Enter Your Desired Number"
read Num
echo "The Number You've Printed is $Num"

soumyo_sahu@GENESIS:~/Assignment-2$ ./q3.sh
Enter Your Desired Number
08
The Number You've Printed is 08
```

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



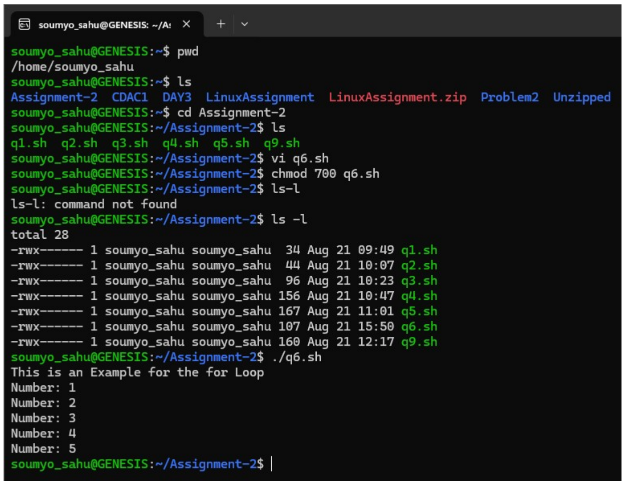
```
soumyo_sahu@GENESIS:~/Assignment-2$ pwd
/home/soumyo_sahu/Assignment-2
soumyo_sahu@GENESIS:~/Assignment-2$ ls
q1.sh  q2.sh  q3.sh  q4.sh
soumyo_sahu@GENESIS:~/Assignment-2$ cat q4.sh
#!/bin/bash
echo "Please enter the value of a: "
read a
echo "Please enter the value of b: "
read b
(( Sum=a + b ))
echo "The sum of both numbers is $Sum"

soumyo_sahu@GENESIS:~/Assignment-2$ ./q4.sh
Please enter the value of a:
5
Please enter the value of b:
5
The sum of both numbers is 10
soumyo_sahu@GENESIS:~/Assignment-2$ 
```

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



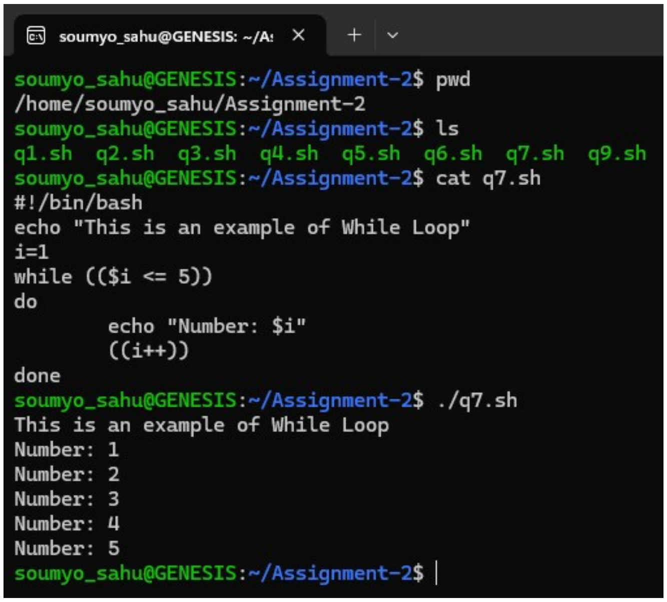
```
soumyo_sahu@GENESIS:~/Assignment-2$ pwd
/home/soumyo_sahu/Assignment-2
soumyo_sahu@GENESIS:~/Assignment-2$ ls
q1.sh  q2.sh  q3.sh  q4.sh  q5.sh
soumyo_sahu@GENESIS:~/Assignment-2$ cat q5.sh
#!/bin/bash
echo "Please Enter Your Desired Number: "
read a
if (($a%2 == 0))
then
        echo "Your Number is an Even Number"
else
        echo "Your Number ia a Odd Number"
fi

soumyo_sahu@GENESIS:~/Assignment-2$ ./q5.sh
Please Enter Your Desired Number:
42
Your Number is an Even Number
soumyo_sahu@GENESIS:~/Assignment-2$ ./q5.sh
Please Enter Your Desired Number:
23
Your Number ia a Odd Number
soumyo_sahu@GENESIS:~/Assignment-2$ |
```

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



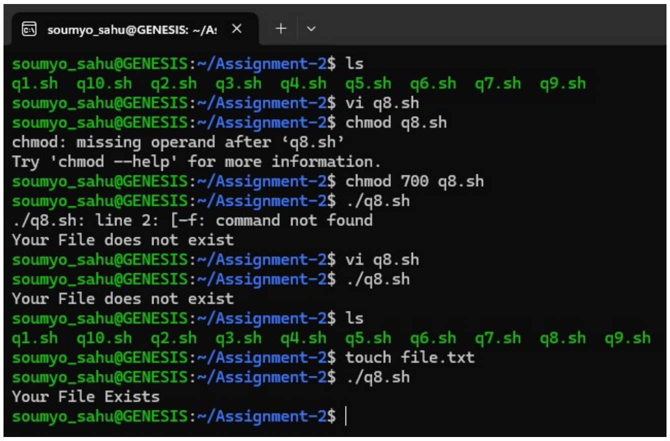
```
soumyo_sahu@GENESIS:~$ pwd
/home/soumyo_sahu
soumyo_sahu@GENESIS:~$ ls
Assignment-2  CDAC1  DAY3  LinuxAssignment  LinuxAssignment.zip  Problem2  Unzipped
soumyo_sahu@GENESIS:~$ cd Assignment-2
soumyo_sahu@GENESIS:~/Assignment-2$ ls
q1.sh  q2.sh  q3.sh  q4.sh  q5.sh  q9.sh
soumyo_sahu@GENESIS:~/Assignment-2$ vi q6.sh
soumyo_sahu@GENESIS:~/Assignment-2$ chmod 700 q6.sh
soumyo_sahu@GENESIS:~/Assignment-2$ ls-l
ls-l: command not found
soumyo_sahu@GENESIS:~/Assignment-2$ ls -l
total 28
-rwx------ 1 soumyo_sahu soumyo_sahu  34 Aug 21 09:49 q1.sh
-rwx------ 1 soumyo_sahu soumyo_sahu  44 Aug 21 10:07 q2.sh
-rwx------ 1 soumyo_sahu soumyo_sahu  96 Aug 21 10:23 q3.sh
-rwx------ 1 soumyo_sahu soumyo_sahu 156 Aug 21 10:47 q4.sh
-rwx------ 1 soumyo_sahu soumyo_sahu 167 Aug 21 11:01 q5.sh
-rwx------ 1 soumyo_sahu soumyo_sahu 107 Aug 21 15:50 q6.sh
-rwx------ 1 soumyo_sahu soumyo_sahu 160 Aug 21 12:17 q9.sh
soumyo_sahu@GENESIS:~/Assignment-2$ ./q6.sh
This is an Example for the for Loop
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
soumyo_sahu@GENESIS:~/Assignment-2$ |
```

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



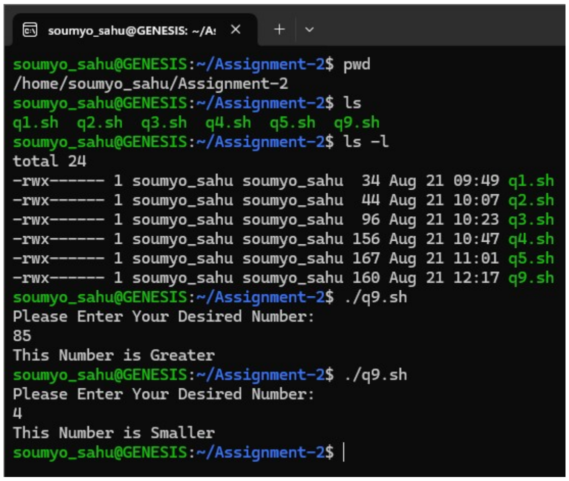
```
soumyo_sahu@GENESIS:~/Assignment-2$ pwd
/home/soumyo_sahu/Assignment-2
soumyo_sahu@GENESIS:~/Assignment-2$ ls
q1.sh  q2.sh  q3.sh  q4.sh  q5.sh  q6.sh  q7.sh  q9.sh
soumyo_sahu@GENESIS:~/Assignment-2$ cat q7.sh
#!/bin/bash
echo "This is an example of While Loop"
i=1
while (($i <= 5))
do
        echo "Number: $i"
        ((i++))
done
soumyo_sahu@GENESIS:~/Assignment-2$ ./q7.sh
This is an example of While Loop
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
soumyo_sahu@GENESIS:~/Assignment-2$ |
```

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".



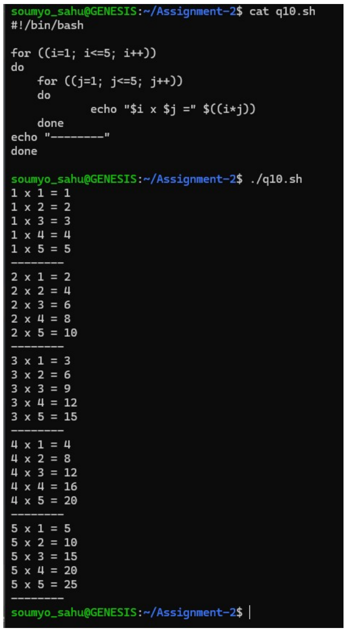
```
soumyo_sahu@GENESIS:~/Assignment-2$ ls
q1.sh  q10.sh  q2.sh  q3.sh  q4.sh  q5.sh  q6.sh  q7.sh  q9.sh
soumyo_sahu@GENESIS:~/Assignment-2$ vi q8.sh
soumyo_sahu@GENESIS:~/Assignment-2$ chmod q8.sh
chmod: missing operand after ‘q8.sh’
Try 'chmod --help' for more information.
soumyo_sahu@GENESIS:~/Assignment-2$ chmod 700 q8.sh
soumyo_sahu@GENESIS:~/Assignment-2$ ./q8.sh
./q8.sh: line 2: [-f: command not found
Your File does not exist
soumyo_sahu@GENESIS:~/Assignment-2$ vi q8.sh
soumyo_sahu@GENESIS:~/Assignment-2$ ./q8.sh
Your File does not exist
soumyo_sahu@GENESIS:~/Assignment-2$ ls
q1.sh  q10.sh  q2.sh  q3.sh  q4.sh  q5.sh  q6.sh  q7.sh  q8.sh  q9.sh
soumyo_sahu@GENESIS:~/Assignment-2$ touch file.txt
soumyo_sahu@GENESIS:~/Assignment-2$ ./q8.sh
Your File Exists
soumyo_sahu@GENESIS:~/Assignment-2$ |
```

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.



```
soumyo_sahu@GENESIS:~/Assignment-2$ pwd
/home/soumyo_sahu/Assignment-2
soumyo_sahu@GENESIS:~/Assignment-2$ ls
q1.sh  q2.sh  q3.sh  q4.sh  q5.sh  q9.sh
soumyo_sahu@GENESIS:~/Assignment-2$ ls -l
total 24
-rwx------ 1 soumyo_sahu soumyo_sahu  34 Aug 21 09:49 q1.sh
-rwx------ 1 soumyo_sahu soumyo_sahu  44 Aug 21 10:07 q2.sh
-rwx------ 1 soumyo_sahu soumyo_sahu  96 Aug 21 10:23 q3.sh
-rwx------ 1 soumyo_sahu soumyo_sahu 156 Aug 21 10:47 q4.sh
-rwx------ 1 soumyo_sahu soumyo_sahu 167 Aug 21 11:01 q5.sh
-rwx------ 1 soumyo_sahu soumyo_sahu 160 Aug 21 12:17 q9.sh
soumyo_sahu@GENESIS:~/Assignment-2$ ./q9.sh
Please Enter Your Desired Number:
85
This Number is Greater
soumyo_sahu@GENESIS:~/Assignment-2$ ./q9.sh
Please Enter Your Desired Number:
4
This Number is Smaller
soumyo_sahu@GENESIS:~/Assignment-2$ |
```

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.



```
soumyo_sahu@GENESIS:~/Assignment-2$ cat q10.sh
#!/bin/bash

for ((i=1; i<=5; i++))
do
    for ((j=1; j<=5; j++))
    do
            echo "$i x $j =" $((i*j))
    done
echo "--------"
done

soumyo_sahu@GENESIS:~/Assignment-2$ ./q10.sh
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
--------
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
--------
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
--------
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
--------
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
--------
soumyo_sahu@GENESIS:~/Assignment-2$ |
```

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 D

Common Interview Questions (Must know)

1. What is an operating system, and what are its primary functions?
2. Explain the difference between process and thread.
3. What is virtual memory, and how does it work?
4. Describe the difference between multiprogramming, multitasking, and multiprocessing.
5. What is a file system, and what are its components?
6. What is a deadlock, and how can it be prevented?
7. Explain the difference between a kernel and a shell.
8. What is CPU scheduling, and why is it important?
9. How does a system call work?
10. What is the purpose of device drivers in an operating system?
11. Explain the role of the page table in virtual memory management.
12. What is thrashing, and how can it be avoided?
13. Describe the concept of a semaphore and its use in synchronization.
14. How does an operating system handle process synchronization?
15. What is the purpose of an interrupt in operating systems?
16. Explain the concept of a file descriptor.
17. How does a system recover from a system crash?
18. Describe the difference between a monolithic kernel and a microkernel.
19. What is the difference between internal and external fragmentation?
20. How does an operating system manage I/O operations?
21. Explain the difference between preemptive and non-preemptive scheduling.
22. What is round-robin scheduling, and how does it work?
23. Describe the priority scheduling algorithm. How is priority assigned to processes?
24. What is the shortest job next (SJN) scheduling algorithm, and when is it used?
25. Explain the concept of multilevel queue scheduling.
26. What is a process control block (PCB), and what information does it contain?
27. Describe the process state diagram and the transitions between different process states.
28. How does a process communicate with another process in an operating system?
29. What is process synchronization, and why is it important?
30. Explain the concept of a zombie process and how it is created.
31. Describe the difference between internal fragmentation and external fragmentation.
32. What is demand paging, and how does it improve memory management efficiency?
33. Explain the role of the page table in virtual memory management.
34. How does a memory management unit (MMU) work?
35. What is thrashing, and how can it be avoided in virtual memory systems?
36. What is a system call, and how does it facilitate communication between user programs and the operating system?
37. Describe the difference between a monolithic kernel and a microkernel.
38. How does an operating system handle I/O operations?
39. Explain the concept of a race condition and how it can be prevented.
40. Describe the role of device drivers in an operating system.
41. What is a zombie process, and how does it occur? How can a zombie process be prevented?
42. Explain the concept of an orphan process. How does an operating system handle orphan processes?
43. What is the relationship between a parent process and a child process in the context of process management?
44. How does the fork() system call work in creating a new process in Unix-like operating systems?
45. Describe how a parent process can wait for a child process to finish execution.
46. What is the significance of the exit status of a child process in the wait() system call?
47. How can a parent process terminate a child process in Unix-like operating systems?
48. Explain the difference between a process group and a session in Unix-like operating systems.
49. Describe how the exec() family of functions is used to replace the current process image with a new one.
50. What is the purpose of the waitpid() system call in process management? How does it differ from wait()?
51. How does process termination occur in Unix-like operating systems?
52. What is the role of the long-term scheduler in the process scheduling hierarchy? How does it influence the degree of multiprogramming in an operating system?
53. How does the short-term scheduler differ from the long-term and medium-term schedulers in terms of frequency of execution and the scope of its decisions?
54. Describe a scenario where the medium-term scheduler would be invoked and explain how it helps manage system resources more efficiently.

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:-

P	A.T	B.T	C.T	T.A.T	W.T	R.T
P1	0	5	5	5	0	0
P2	1	3	8	7	4	4
P3	2	6	14	12	6	6

Gant Chart :-

P1	P2	P3

Avg. W.T = 3.33

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

0 5 8 14

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:-

P	A.T	B.T	C.T	T.A.T	W.T	R.T
P1	0	3	3	3	0	0
P2	1	5	13	12	7	7
P3	2	1	4	2	1	1
P4	3	4	8	5	1	1

Gant Chart :-

P1	P3	P4	P2

Avg. T.A.T = 5.5

3. Consider the following processes with arrival times, burst times, and priorities (lower number indicates higher priority):

0 3 4 8 13

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:-

P	A.T	B.T	Priority	C.T	T.A.T	W.T	R.T
P1	0	6	3	6	6	0	0
P2	1	4	1	10	9	5	5
P3	2	7	4	19	17	10	10
P4	3	2	2	12	9	7	7

Gant Chart :-

P1	P2	P4	P3

Avg. W.T = 5.5

4. Consider the following processes with arrival times and burst times, and the time quantum for Round Robin scheduling is 2 units:

0 6 10 12 19

Avg. W.T = 5.5

4. Consider the following processes with arrival times and burst times, and the time quantum for Round Robin scheduling is 2 units:

0 6 10 12 19

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:-

P	A.T	B.T	C.T	T.A.T	W.T	R.T
P1	0	4	8	8	4	0
P2	1	5	14	13	8	1
P3	2	2	6	4	2	2
P4	3	3	13	10	7	5

Gant Chart :-

P1	P2	P3	P1	P4	P2	P4	P2

Avg. T.A.T = 8.75

0 2 4 6 8 10 12 13 14

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:-

Submission Guidelines:

- Document each step of your solution and any challenges faced.
- Upload it on your GitHub repository

Additional Tips:

- Experiment with different options and parameters of each command to explore their functionalities.
- This assignment is tailored to align with interview expectations, CCEE standards, and industry demands.
- If you complete this then your preparation will be skyrocketed.