CDAC MUMBAI

Concepts of Operating System

Assignment 2

Part A

What will the following commands do?

name="Productive" = > assign the value to name variable

touch file.txt = > create new file with name of file.txt

ls -a => show all the file in directory including hidden

rm file.txt => remove / delete file

cp file1.txt file2.txt = > creating a copy

mv file.txt /path/to/directory/ = > here file.txt is moving in directory

chmod 755 script.sh = >changing permission to script.sh(owner all permission group)

grep "pattern" file.txt = >searching word pattern in file.txt

kill PID = > killing the process

mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt = > directory is created (mydir) and created new file inside that directory and hellow, world is over ridden in file.txt and content of file.txt is printed over terminal

ls -l | grep ".txt"=> here we are printing all the file which is contains .txt

cat file1.txt file2.txt | sort | uniq => here we are printing unique(non repatative) words inside file1.txt and file2.txt

ls -l | grep "^d" printing all the directories

grep -r "pattern" /path/to/directory/ =>recursive searching from word pattern in all the files inside the directory

cat file1.txt file2.txt | sort | uniq –d =.> here we are sorting and printing only duplicate content

chmod 644 file.txt => changing permission of file.txt

cp -r source_directory destination_directory => this is copy command

find /path/to/search -name "*.txt" => searching for .txt file inside all the directory

chmod u+x file.txt => changing user permission for user

echo $PATH = it display the directory where the system search for directory

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 -> **true** if directory1 does not exist.

8. rm -rf file.txt deletes a file forcefully without confirmation. -> **true**

Identify the Incorrect Commands:

1. chmodx is used to change file permissions. -> **false**

2. cpy is used to copy files and directories. -> **false**

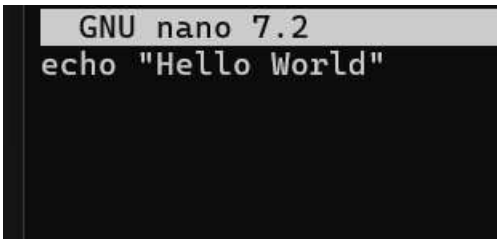
3. mkfile is used to create a new file. -> **false**

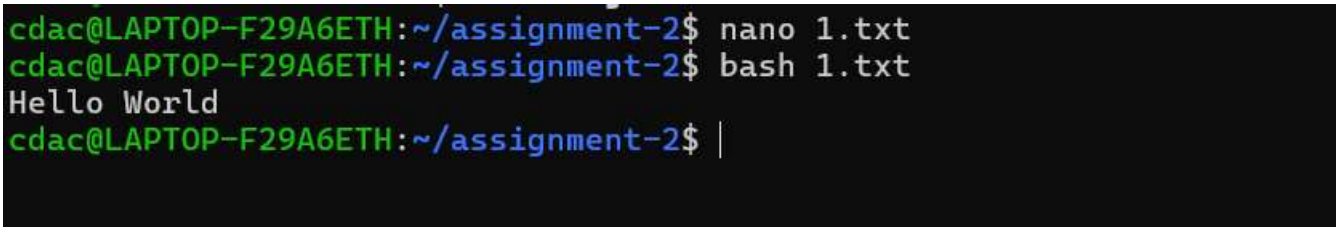
4. catx is used to concatenate files.-> **false**

5. rn is used to rename files. -> false

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.

```
name="CDAC Mumbai"
echo $name
```

```
cdac@LAPTOP-F29A6ETH:~/assignment-2$ nano 1.txt
cdac@LAPTOP-F29A6ETH:~/assignment-2$ bash 1.txt
CDAC Mumbai
cdac@LAPTOP-F29A6ETH:~/assignment-2$
```

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

```
echo "Enter a number:"
read name

echo $name
```

```
cdac@LAPTOP-F29A6ETH:~/assignment-2$ nano 1.txt
cdac@LAPTOP-F29A6ETH:~/assignment-2$ bash 1.txt
Enter a number:
5
5
cdac@LAPTOP-F29A6ETH:~/assignment-2$
```

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

```
sum=`expr 5 + 6`
echo "$sum"
```

```
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ nano sum
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ bash sum
11
cdac@LAPTOP-F29A6ETH:~/firstAssignment$
```

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

```
echo "Enter a number"
read n

if (( n % 2 == 0 ))
then
        echo "even number"
else
        echo "odd number"
fi
```

```
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ bash sumOfOddNo.
Enter a number
5
odd number
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ bash sumOfOddNo.
Enter a number
4
even number
cdac@LAPTOP-F29A6ETH:~/firstAssignment$
```

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

```
for((i=1;i<=5;i++))
do
        echo "$i"
done
```

```
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ nano for
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ bash for
1
2
3
4
5
cdac@LAPTOP-F29A6ETH:~/firstAssignment$
```

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

```
i=1
while [ $i -le 5 ]
do
    echo $i
    ((i++))
done
```

```
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ nano for
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ bash for
1
2
3
4
5
cdac@LAPTOP-F29A6ETH:~/firstAssignment$
```

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".

```
if [ -f "file.txt" ]; then
    echo "File exists"
else
    echo "File does not exist"
fi
```

```
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ nano for
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ bash for
File does not exist
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ tuch file.txt
Command 'tuch' not found, did you mean:
  command 'tucd' from deb tcm (2.20+TSQD-7)
  command 'touch' from deb coreutils (9.4-2ubuntu2)
Try: sudo apt install <deb name>
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ touch file.txt
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ bash for
File exists
cdac@LAPTOP-F29A6ETH:~/firstAssignment$
```

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.

```
echo "Enter a number : "
read num
if [ $num -gt 10 ]; then
    echo "The number is greater than 10."
else
    echo "The number is 10 or less."
fi
```

```
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ nano for
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ bash for
Enter a number :
5
The number is 10 or less.
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ bash for
Enter a number :
20
The number is greater than 10.
cdac@LAPTOP-F29A6ETH:~/firstAssignment$
```

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.

```
for i in {1..5}
do
    for j in {1..5}
    do
        printf "%4d" $((i * j))
    done
    echo
done
```

```
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ nano for
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ bash for
   1   2   3   4   5
   2   4   6   8  10
   3   6   9  12  15
   4   8  12  16  20
   5  10  15  20  25
cdac@LAPTOP-F29A6ETH:~/firstAssignment$
```

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.

```
while true
do
    echo "Enter a number: "
    read num
    if [ $num -lt 0 ]
    then
        echo "Negative number entered. Exiting..."

    fi
    echo "Square: $((num * num))"
done
```

```
cdac@LAPTOP-F29A6ETH:~/firstAssignment$ bash for
Enter a number:
6
Square: 36
Enter a number:
4
Square: 16
Enter a number:
-1
Negative number entered. Exiting...
```

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.

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?

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.