CDAC MUMBAI

Concepts of Operating System

Assignment 2

Part A

What will the following commands do?

1. **echo "Hello, World!"**
   - Prints "Hello, World!" to the terminal.
2. **name="Productive"**
   - Assigns the string "Productive" to the variable name
3. **touch file.txt**
   - Creates an empty file named file.txt if it doesn't exist or updates its timestamp if it already exists.
4. **ls -a**
   - Lists all files and directories, including hidden files (those starting with .)
5. **rm file.txt**
   - Deletes file.txt permanently.
6. **cp file1.txt file2.txt**
   - Copies file1.txt to file2.txt. If file2.txt exists, it will be overwritten.
7. **mv file.txt /path/to/directory/**
   - Moves file.txt to /path/to/directory/. It can also rename a file if the target path is a filename.
8. **chmod 755 script.sh**
   - Changes the permissions of script.sh to rwxr-xr-x (owner: read, write, execute; group & others: read, execute).
9. **grep "pattern" file.txt**
   - Searches for "pattern" in file.txt and prints matching lines.
10. **kill PID**
    - Terminates the process with the given Process ID

11. **mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt**
    - Creates a directory mydir navigates into it, creates file.txt, writes "Hello, World!" into it, and displays its content.

12. **ls -l | grep ".txt"**
    - Lists files in long format and filters those containing .txt in their names.

13. **cat file1.txt file2.txt | sort | uniq**
    - Concatenates file1.txt and file2.txt sorts the combined output, and removes duplicate lines.

14. **ls -l | grep "^d"**
    - Lists directories (since directories start with d in ls -l output).

15. **grep -r "pattern" /path/to/directory/**
    - Recursively searches for "pattern" in all files under /path/to/directory/.

16. **cat file1.txt file2.txt | sort | uniq –d**
    - Combines file1.txt and file2.txt sorts them, and prints only duplicate lines.

17. **chmod 644 file.txt**
    - Sets permissions of file.txt to rw-r--r-- (owner: read & write; group & others: read).

18. **cp -r source_directory destination_directory**
    - Recursively copies source_directory and its contents to destination_directory.

19. **find /path/to/search -name "*.txt"**
    - Finds all .txt files in /path/to/search and its subdirectories.

20. **chmod u+x file.txt**
    - Gives the owner (u) execute (x) permission on file.txt.

21. **echo $PATH**
    - Displays the system's PATH variable, which lists directories where executable files are searched for.

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.**
    - False

Identify the Incorrect Commands:

- **chmodx is used to change file permissions**.
    - The correct command is chmod, not chmodx.

- **cpy is used to copy files and directories**
    - The correct command is cp, not cpy.

 mkfile is used to create a new file.

- There is no mkfile command in Linux.
- The correct way to create a file is touch filename or echo "" > filename.

 catx is used to concatenate files.

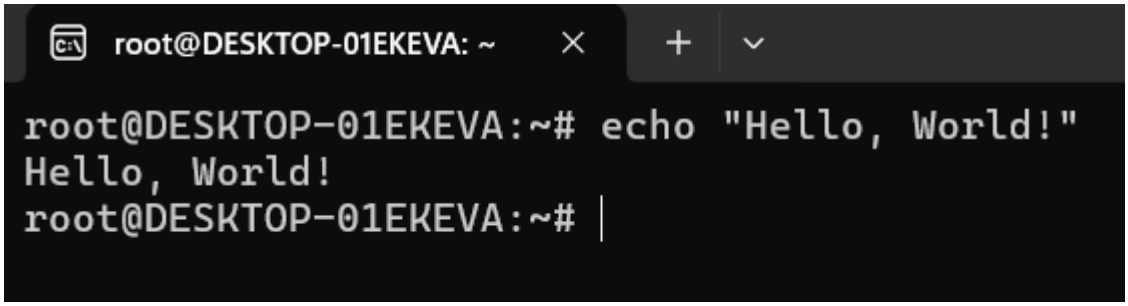
- The correct command is cat, not catx.

 rn is used to rename files.

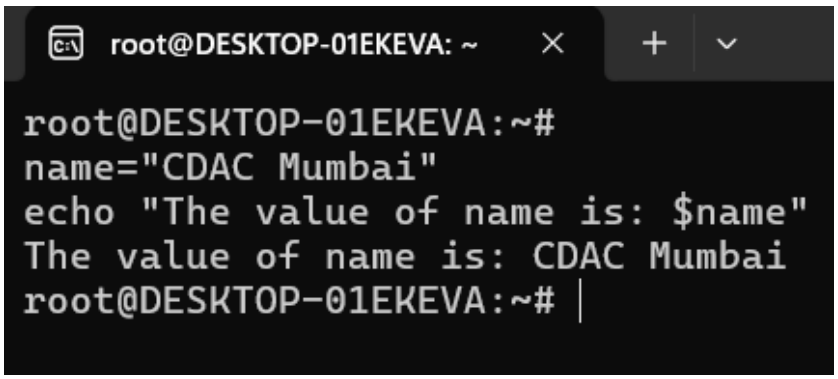
- There is no rn command in Linux.
- The correct way to rename a file is mv oldname newname.

Part C

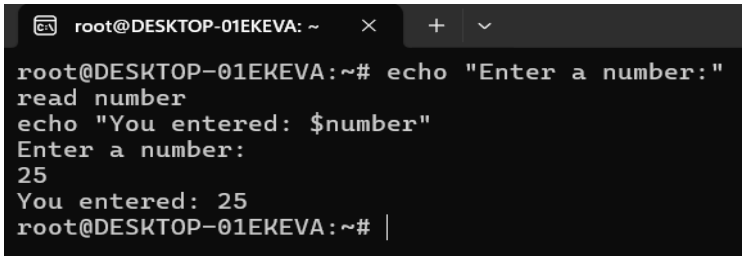
Question1: Write a shell script that prints "Hello, World!" to the terminal.



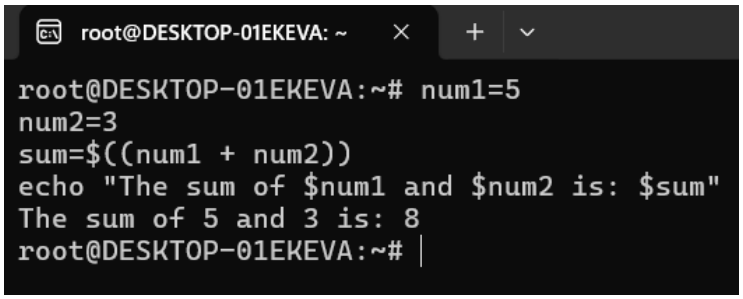
Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable. Question



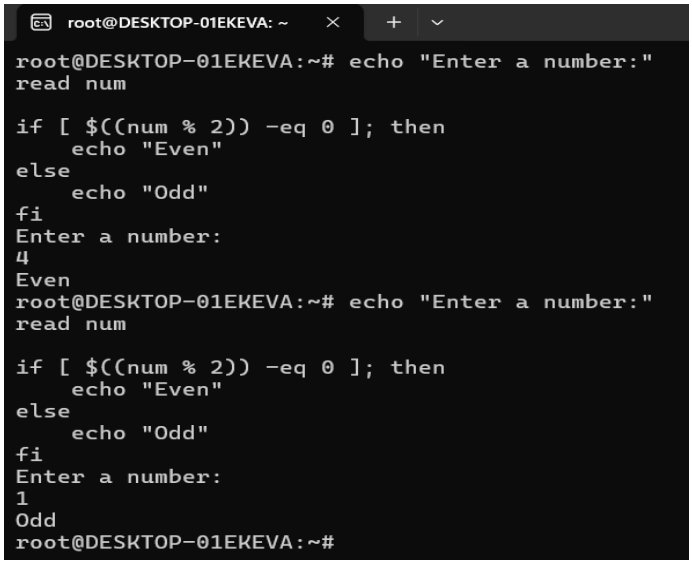
Question 3: Write a shell script that takes a number as input from the user and prints it.



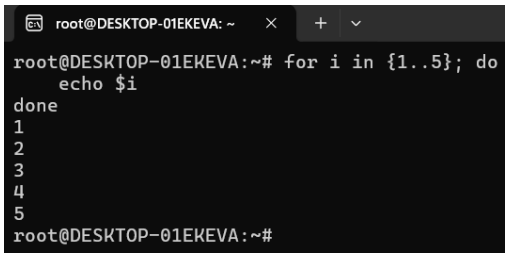
Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result. Question



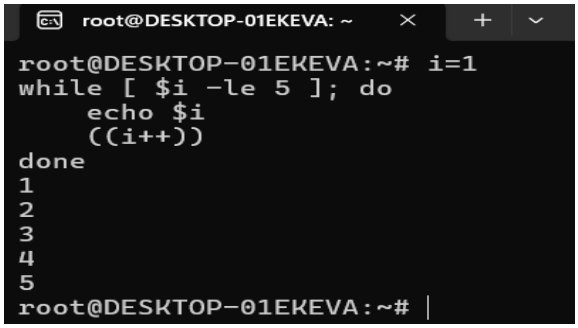
Question 5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd". Question



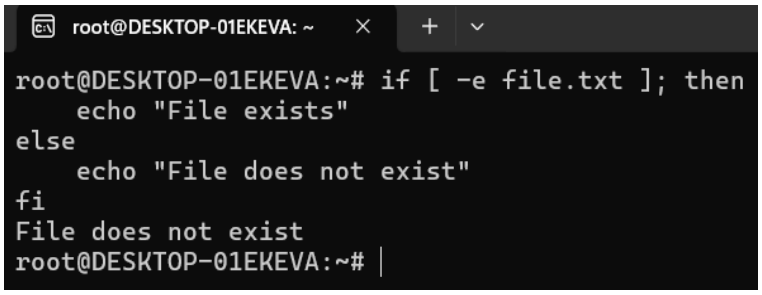
Question 6: Write a shell script that uses a for loop to print numbers from 1 – 5



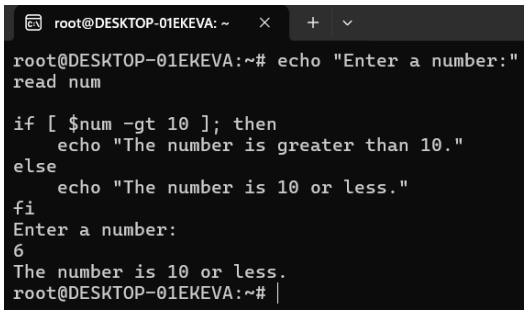
Question 7: Write a shell script that uses a while loop to print numbers from 1 to 5.



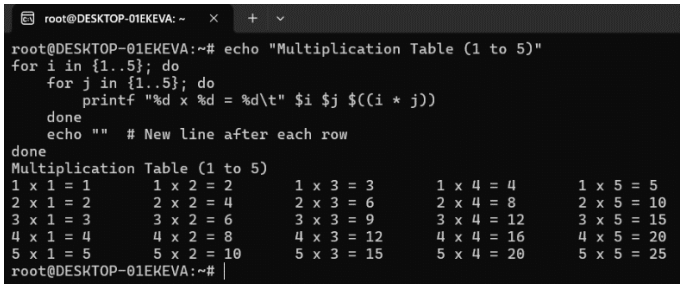
Question 8: Write a shell script that checks if a file named "file.txt" exists in the current directory. If it does, print "File exists", otherwise, print "File does not exist".



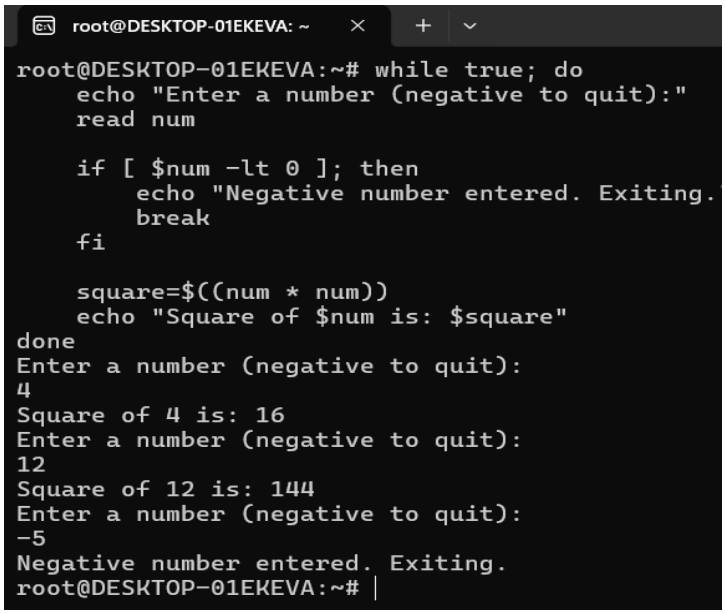
Question 9: Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.



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.

ANSWER-1 Given Data

TAT ⇒ CT − AT

Process	Arrival time	Burst Time	Completion Time (CT)	Turnaround Time (TAT)
P1	0	5	0 → 5 = 5	5 − 0 = 5
P2	1	3	5 → 3 = 8	8 − 1 = 7
P3	2	6	8 → 6 = 14	14 − 2 = 12

Waiting Time (WT) ⇒ TAT − Burst time

P1	5 − 5 = 0
P2	7 − 3 = 4
P3	12 − 6 = 6

⟹ Average Waiting Time (AWT)?

$$AWT = \frac{\Sigma WT}{\text{Total Processes}} = \frac{0+4+6}{3} = \frac{10}{3} = 3.33 \text{ units}$$

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.

ANSWER-2 Given Data

Process	AT	BT	Completion Time	TAT
P1	0	3	3	3-0=3
P2	1	5	4	4-2=2
P3	2	1	8	8-3=5
P4	3	4	13	13-1=12

P1 → P1 arrives first at time 0 & executes at 3, CT = 3

P2 → At 3, ready queue P2(BT=5), P3(BT=1), P4(BT=4) [P3 has shortest → CT(P3) = 3+1=4]

P3 → At 4, P2(BT=5), P4(BT)=4 so P4 has shortest [CT(P4) = 4+4=8]

P4 → Finally P2 executes & that's only remaing ~~time~~ process.

CT(P2) = 8+5 = 13

Average Turnaround Time (ATAT)

$$ATAT = \frac{\Sigma TAT}{\text{Total Process}} = \frac{3+2+5+12}{4} = \frac{22}{4} = 5.5 \text{ unit}$$

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.

ANSWER-3 Given DATA

Process	AT	BT	Priority	CT	TAT	WT
P1	0	6	3	6	6-0 = 6	6-6=0
P2	1	4	1	10	10-1 = 9	9-4=5
P3	2	7	4	12	12-3 = 9	9-2=7
P4	3	2	2	19	19-2 = 17	17-7=10

Process execules basend on priority

- Lower number = higher Priority.

P1 = 0 + 6 = 6 CT(P1)

P2 = P2 has highest prority ; CT(P2) = 6 + 4 = 10

P3 = P4 has highest priority (2); CT(P4) = 10 + 2 = 12

P3 executes as it's only left

- CT (P3) = 12 + 7 = 19

⇒ Average Waiting Time

$$AWT = \frac{\Sigma WT}{\text{Total Process}}$$

$$= \frac{0+5+7+10}{4} = \frac{22}{4} = 5.5 \text{ units}$$

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.

ANSWER-4

Process	AT	BT	CT	TAT
P1	0	4	10	10-0=10
P2	1	5	14	14-1=13
P3	2	2	6	6-2=4
P4	3	3	13	13-3=10

Gantt Chart

0	P1	2	P2	4	P3	6	P4	8	P1	10	P2	12	P4	13	P2	14

- $t = 0$, P1 executes for 2 units (P1 : 2/4)
- $t = 2$, P2 " (P2 : 2/5)
- $t = 4$, P3 " (P3 : 2/2)
- $t = 6$, P4 " (P4 : 2/3)
- $t = 8$, P1 " (P1 : 4/4)
- $t = 10$, P2 " (P2 : 4/5)
- $t = 12$, P4 executes for 1 more units (P4 : 3/3)
- $t = 13$, P2 executes for 1 more units (P2 : 5/5)

$\Rightarrow$ Average Turnaround Time (ATAT)

$$ATAT = \frac{\Sigma TAT}{\text{Total Process}}$$

$$= \frac{10+13+4+10}{4} = \frac{37}{4} = 9.25 \text{ units.}$$