CDAC MUMBAI

Concepts of Operating System Assignment 2

Part A

What will the following commands do?

- echo "Hello, World!" ➔**Prints “Hello,World” in terminal.**
- name="Productive" ➔**Assign value Productive to variable name.**
- touch file.txt ➔**Creates new file named file.txt**
- ls -a ➔**Lists all files also hidden files.**
- rm file.txt ➔ **Removes file file.txt from directory**
- cp file1.txt file2.txt ➔ **Copies file1.txt to file2.txt**
- mv file.txt /path/to/directory/ ➔**Moves file.txt to the given directory.**
- chmod 755 script.sh ➔**Gives all permissions to owner and read &execute to group and others.**
- grep "pattern" file.txt ➔ **Finds word “Pattern” in file.txt.**
- kill PID ➔**Terminate process of given process ID.**
- mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt ➔**Creates mydir, moves into it, creates file.txt, writes "Hello, World!" into it, and prints its contents.**
- ls -l | grep ".txt" ➔ **Lists all .txt files in long format.**
- cat file1.txt file2.txt | sort | uniq ➔**Merges files then sorts lines and removes duplicates.**
- ls -l | grep "^d" ➔ **Lists only directories.**
- grep -r "pattern" /path/to/directory/ ➔**Searches for "pattern" recursively in the directory.**
- cat file1.txt file2.txt | sort | uniq –d ➔ **Finds duplicate lines in both files.**
- chmod 644 file.txt ➔**R/W for owner and read only for others.**
- cp -r source_directory destination_directory ➔**Copies directory and its content.**
- find /path/to/search -name "*.txt" ➔**Finds all .txt files in the given path.**
- chmod u+x file.txt ➔**gives execute permission to the file owner.**
- echo $PATH ➔**Displays System executable path.**

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 cmd, Correct is chmod.**
2. cpy is used to copy files and directories. ➔ **Incorrect cmd, Correct is cp.**
3. mkfile is used to create a new file. ➔ **Incorrect cmd, Correct is touch.**
4. catx is used to concatenate files. ➔ **Incorrect cmd, Correct is cat.**
5. rn is used to rename files. ➔ **Incorrect cmd, Correct is mv.**

Part C

Question 1: Write a shell script that prints "Hello, World!" to the terminal. ➔ **echo “Hello,World”**

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

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

➔ **read num**
echo $num

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

➔ **echo $((5+3))**

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

➔ **read num**
if ((num % 2 == 0)); then echo "Even"; else echo "Odd"; fi

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

➔ **for i in {1..5}**
do
echo $i; done

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**

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".

➔
⇨ **[-f file.txt] && echo "File exists" || echo "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.

➔
⇨ **read num**
⇨ **if ((num > 10))**
⇨ **then echo "Greater than 10"**
⇨ **else echo "Less than or equal to 10"**
⇨ **fi**

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**
⇨ **echo -n "$((i * j)) "**
⇨ **done**
⇨ **echo**
⇨ **done**

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
read num
((num < 0)) && break
echo $((num * num))
done
```

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.

⇨ **Average Waiting Time is 5.33**

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.

⇨ **Avg Turnaround time is 5.75**

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.

⇨ **Avg waiting time is 4.25**

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.

⇨ **Avg Turnaround time is 7.75**

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?

⇨ **x in child will be 6**

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.