CDAC MUMBAI

Concepts of Operating System Assignment 2

What will the following commands do?

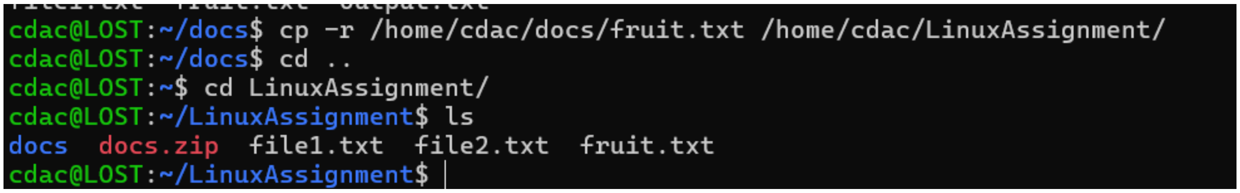
Part A

- echo "Hello, World!"
  Ans: The echo command is used to print the output in the terminal, So the above command will print Hello, World! In the terminal.
- name="Productive"
  Ans: The shell is capable to remembering the data so the above command will store the string "Productive" in the name variable.
- touch file.txt
  Ans: The touch command will create a text file with the name mentioned such as file with the extension as text.
- ls -a
  Ans: The ls command is used to list the files present into a directory and the -a is used to display the hidden files from the directory mostly the system file are hidden in the directory.
- rm file.txt
  Ans: The rm command is used to remove the file from the directory so the above command will delete the file.txt from the directory permanently.
- cp file1.txt file2.txt
  Ans: The cp command is used to copy text from the text file, Hence the above command will copy the text present in file1 and will paste it into the file2.txt. Then the file2 will have the same content as file1.
- mv file.txt /path/to/directory/
  Ans: The mv command is used to move a file from one directory to the destined directory the user wished to set the path. In the above command the file.txt is moved to →path →to→directory.
- chmod 755 script.sh
- grep "pattern" file.txt
  Ans: The grep command is used to group and display the specific string or character mentioned into the command the above command will display the lines of text where the pattern words has occurred.
- kill PID
  Ans:
- mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt
  Ans: The mkdir is used to create a directory named as mydir with an attached command to open the created directory mydir by cd mydir then create a file with the touch command and insert the line echo "Hello, World!" into the file.txt by using > operator and then combine the file with cat which will display the final output as Hello, World! In the terminal.
- ls -l | grep ".txt"
  Ans: The ls command is used to display the content present in the direcrtory such as file and directory as well combining ls -l with the grep ".txt" will display all the files with the txt extension from the directory.

- **cat file1.txt file2.txt | sort | uniq**
  Ans: The cat command is used to combine the content of file1.txt and file2.txt and the pipe operations are used to sort and display the unique strings from the both files.
- **ls -l | grep "^d"**
  Ans: ^d is used for directory entry.
- **grep -r "pattern" /path/to/directory/**
  Ans: The grep -r is used to recursively search for the string containing “pattern” from the path given and display the line containing the pattern word.
- **cat file1.txt file2.txt | sort | uniq –d**
  Ans: The cat command will combine the text from file1 and file2 then it will sort the line which are same in the both file and uniq -d will display the only lines of text which are same in the both files other all text lines will be discarded.

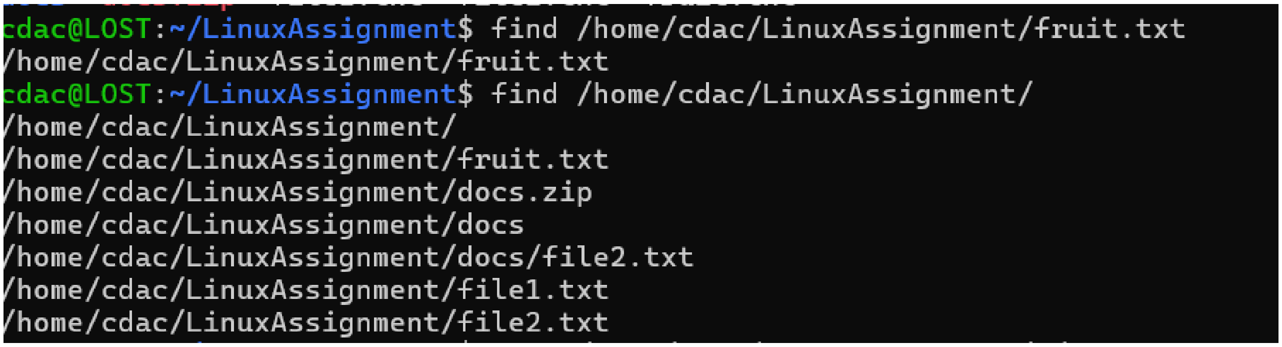
- **chmod 644 file.txt**

- **cp -r source_directory destination_directory**
  Ans: The cp -r command will copy a file from the source directory to the destination directory



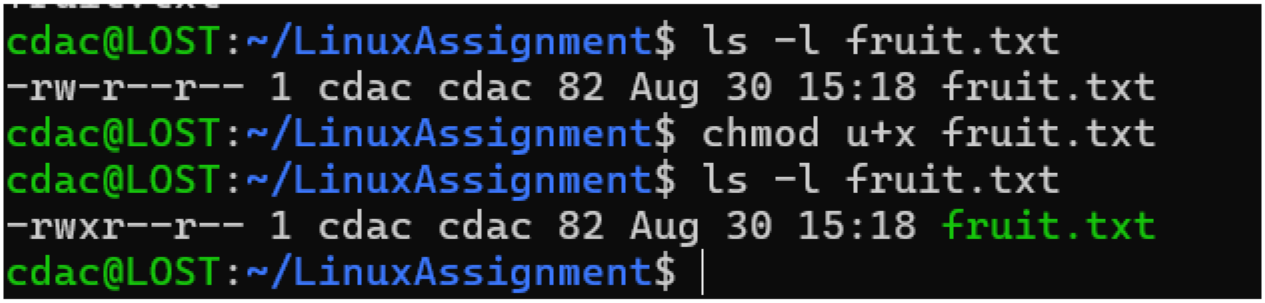
```
cdac@LOST:~/docs$ cp -r /home/cdac/docs/fruit.txt /home/cdac/LinuxAssignment/
cdac@LOST:~/docs$ cd ..
cdac@LOST:~$ cd LinuxAssignment/
cdac@LOST:~/LinuxAssignment$ ls
docs  docs.zip  file1.txt  file2.txt  fruit.txt
cdac@LOST:~/LinuxAssignment$
```

- **find /path/to/search -name "*.txt"**
  Ans: The find command will display the files found in the mentioned path directory



```
cdac@LOST:~/LinuxAssignment$ find /home/cdac/LinuxAssignment/fruit.txt
/home/cdac/LinuxAssignment/fruit.txt
cdac@LOST:~/LinuxAssignment$ find /home/cdac/LinuxAssignment/
/home/cdac/LinuxAssignment/
/home/cdac/LinuxAssignment/fruit.txt
/home/cdac/LinuxAssignment/docs.zip
/home/cdac/LinuxAssignment/docs
/home/cdac/LinuxAssignment/docs/file2.txt
/home/cdac/LinuxAssignment/file1.txt
/home/cdac/LinuxAssignment/file2.txt
```

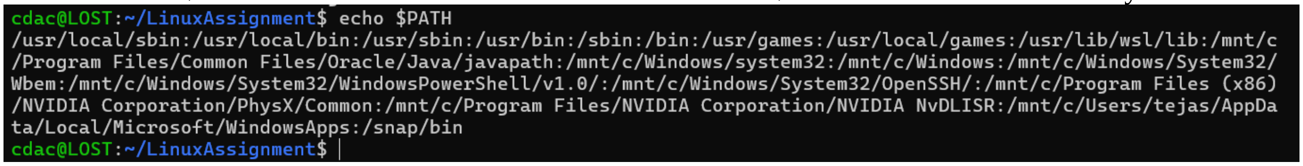
- **chmod u+x file.txt**
  Ans: The chmod command will add the execution permission to the user group.



```
cdac@LOST:~/LinuxAssignment$ ls -l fruit.txt
-rw-r--r-- 1 cdac cdac 82 Aug 30 15:18 fruit.txt
cdac@LOST:~/LinuxAssignment$ chmod u+x fruit.txt
cdac@LOST:~/LinuxAssignment$ ls -l fruit.txt
-rwxr--r-- 1 cdac cdac 82 Aug 30 15:18 fruit.txt
cdac@LOST:~/LinuxAssignment$
```

- **echo $PATH**

  Ans: The echo $PATH allows us to view th ecurrent value of the $PATH variable in the linux system



```
cdac@LOST:~/LinuxAssignment$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/lib/wsl/lib:/mnt/c
/Program Files/Common Files/Oracle/Java/javapath:/mnt/c/Windows/system32:/mnt/c/Windows:/mnt/c/Windows/System32/
Wbem:/mnt/c/Windows/System32/WindowsPowerShell/v1.0/:/mnt/c/Windows/System32/OpenSSH/:/mnt/c/Program Files (x86)
/NVIDIA Corporation/PhysX/Common:/mnt/c/Program Files/NVIDIA Corporation/NVIDIA NvDLISR:/mnt/c/Users/tejas/AppDa
ta/Local/Microsoft/WindowsApps:/snap/bin
cdac@LOST:~/LinuxAssignment$ 
```

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. **[FALSE]**
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. **[FALSE]**
6. **chmod 755 file.txt** gives read, write, and execute permissions to the owner, and read and execute permissions to group and others. **[FALSE]**
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. [chmod is used to change the file permission.]
2. **cpy** is used to copy files and directories. [cp is used to copy files an directories.]
3. **mkfile** is used to create a new file. [touch is used to create a new text file.]
4. **catx** is used to concatenate files. [cat is used to concatenate two files together.]
5. **rn** is used to rename files. [mx command is used to rename the file.]

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 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.