CDAC MUMBAI

Concepts of Operating System Assignment 2

# Part A

What will the following commands do?

* echo "Hello, World!"

Ans:- Print “Hello, World!” to the terminal.

* name="Productive"

Ans:- Creates a variable name and assigns it the value Productive.

* touch file.txt

Ans:- Creates an empty file named file.txt or updates its timestamp if it already exists

* ls -a

Ans:- Lists all files and directories in the current directory, including hidden ones (those starting with . ).

* rm file.txt

Ans:- Removes the file file.txt permanently.

* cp file1.txt file2.txt

Ans:- Copies file1.txt to file2.txt . If file2.txt exists, it will be overwritten.

* mv file.txt /path/to/directory/

Ans:- Moves file.txt to the specified directory.

* chmod 755 script.sh

Ans:- Grants the owner full permissions (read, write, execute) and gives others read and execute permissions on script.sh .

* grep "pattern" file.txt

Ans:- Searches for occurrences of "pattern" in file.txt and prints matching lines.

* kill PID

Ans:- Terminates the process with the specified Process ID (PID).

* mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt

Ans:- a) Creates a directory mydir

1. Changes into mydir
2. Creates an empty file file.txt
3. Writes "Hello, World!" into file.txt
4. Displays the contents of file.txt

* cat file1.txt file2.txt | sort | uniq

Ans:- Concatenates file1.txt and file2.txt , sorts them, and removes duplicate lines

* ls -l | grep "^d"

Ans:- Lists directories (entries starting with d in long format output)

* ls -l | grep ".txt"

Ans:- Lists files in long format and filters only those containing ". Txt" in their names

* grep -r "pattern" /path/to/directory/

Ans:- Searches for "pattern" recursively in all files under /path/to/directory/ .

* cat file1.txt file2.txt | sort | uniq –d

Ans:- Concatenates file1.txt and file2.txt , sorts them, and displays only duplicate lines

* chmod 644 file.txt

Ans:- Grants the owner read and write permissions, while others get read-only access to file.txt .

* cp -r source\_directory destination\_directory

Ans:- Recursively copies source\_directory to destination\_directory , preserving contents.

* find /path/to/search -name "\*.txt"

Ans:- Finds all .txt files in /path/to/search and its subdirectories.

* chmod u+x file.txt

Ans:- Gives the owner ( u ) execute permission on file.txt .

* echo $PATH

Ans:- Displays the system's PATH environment variable, listing directories where executable files are searched for.

# Part B

Identify True or False:

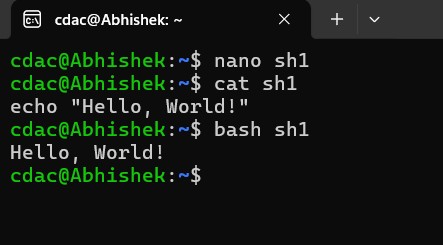
1. **True -**ls is used to list files and directories in a directory.
2. **True -**mv is used to move files and directories.
3. **False-**cd is used to copy files and directories.
4. **True -**pwd stands for "print working directory" and displays the current directory.
5. **True -**grep is used to search for patterns in files.
6. **True -**chmod 755 file.txt gives read, write, and execute permissions to the owner, and read and execute permissions to group and others.
7. **True -**mkdir -p directory1/directory2 creates nested directories, creating directory2 inside True -directory1 if directory1 does not exist.
8. **True -**rm -rf file.txt deletes a file forcefully without confirmation.

Identify the Incorrect Commands:

1. **Incorrect-**chmodx is used to change file permissions.
2. **Incorrect-**cpy is used to copy files and directories.
3. **Incorrect-**mkfile is used to create a new file.
4. **Incorrect-**catx is used to concatenate files.
5. **Incorrect-**rn is used to rename files.

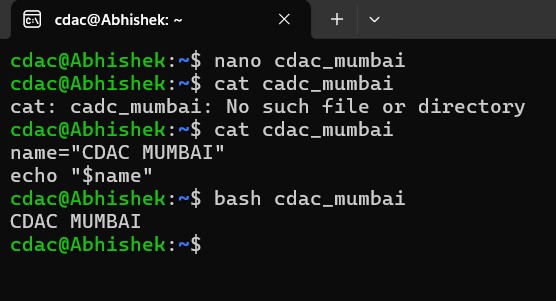
# Part C

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

Ans:-

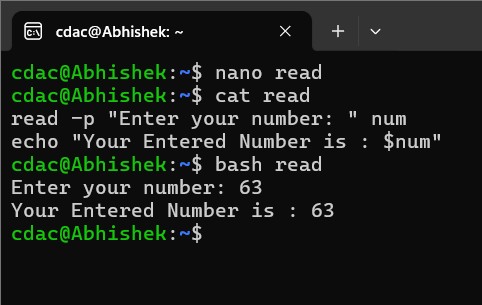
Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.

Ans:-

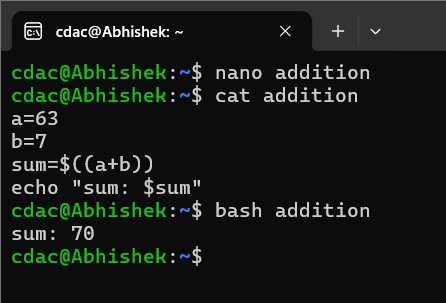


Question 3: Write a shell script that takes a number as input from the user and prints it.

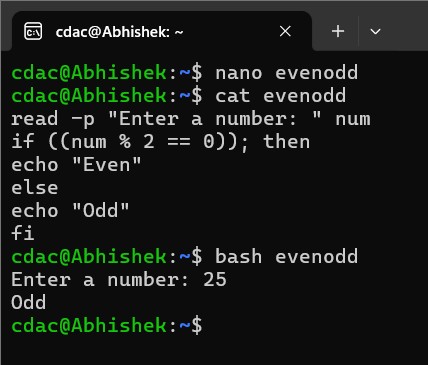
Ans:-



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

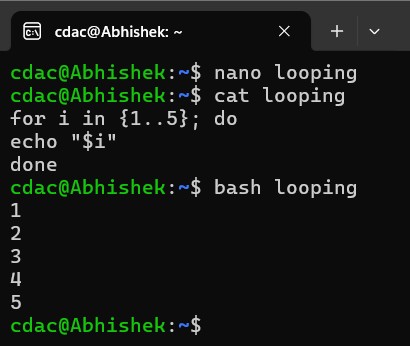
Ans :-

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

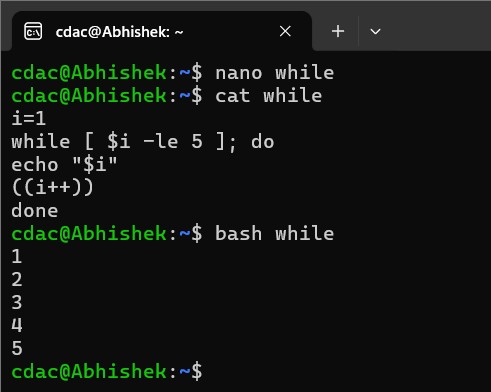
Ans:-

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

Ans:-

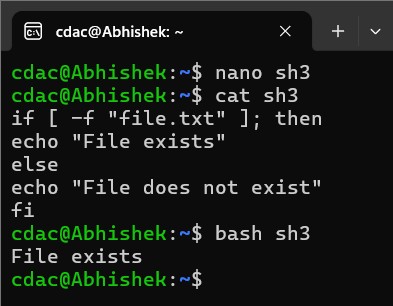


Question 7: Write a shell script that uses a while loop to print numbers from 1 to 5.

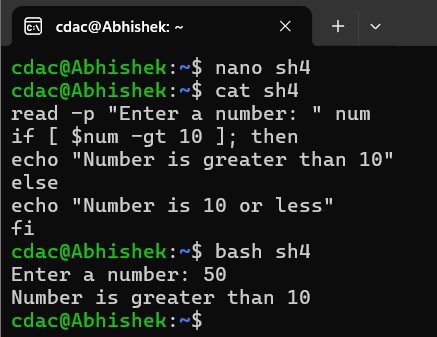
Ans:-

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".

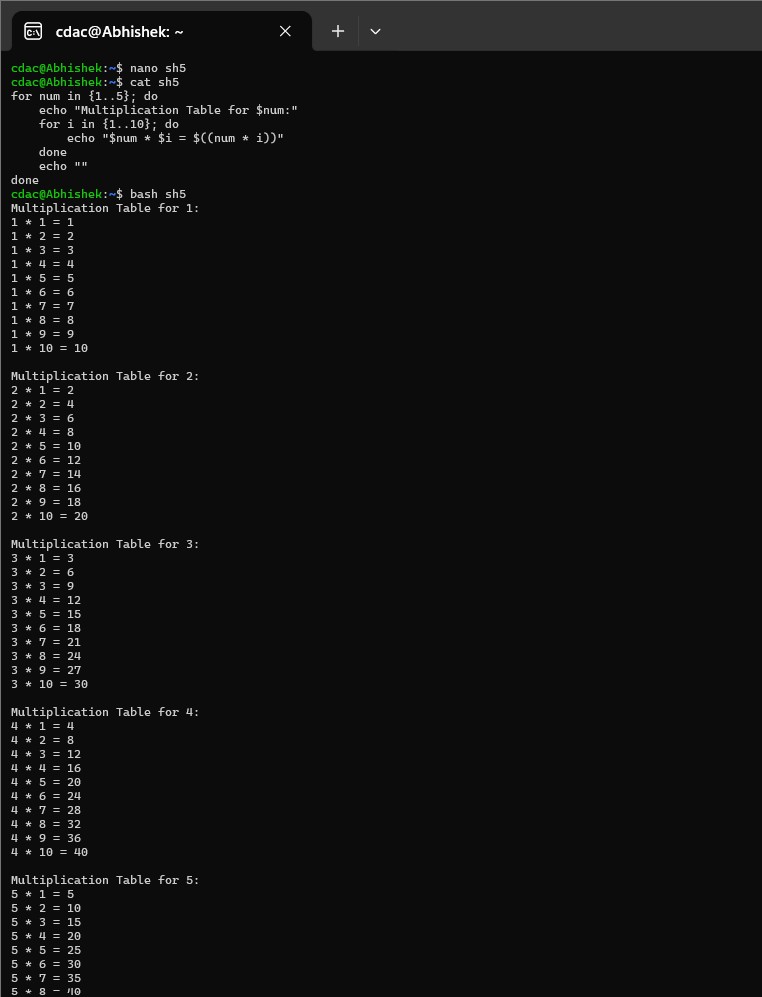
Ans:-



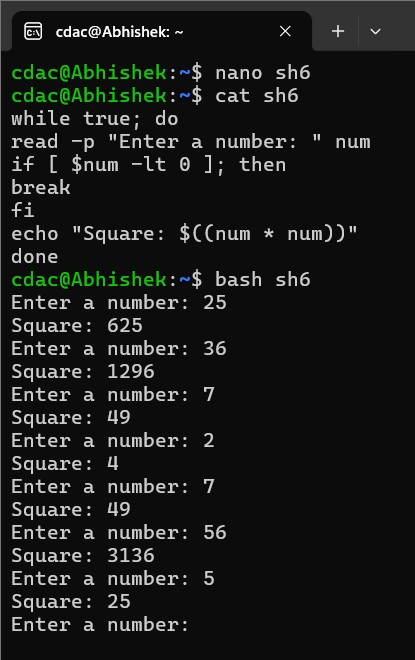
Question 9: Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.

Ans:-

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.

Ans:-

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.

Ans:-

# Part D

Common Interview Questions (Must know)

1. What is an operating system, and what are its primary functions?

Ans :- **:** An operating system (OS) is system software that manages hardware resources and provides services for computer programs. Its primary functions include:

Process management: Scheduling and controlling processes.

Memory management: Allocating and deallocating memory.

File system management: Organizing and managing data storage.

Device management: Controlling hardware devices.

Security and access control: Ensuring proper access to resources.

1. Explain the difference between process and thread.

Ans:- **Process:** A program in execution, having its own memory space.

**Thread:** A lightweight process that shares the same memory space with other threads in the same process. Multiple threads can exist within a single process.

1. What is virtual memory, and how does it work?

Ans :- Virtual memory is an abstraction that allows programs to use more memory than physically available by swapping data between RAM and disk storage. It works by using a page table to map virtual addresses to physical addresses.

1. Describe the difference between multiprogramming, multitasking, and multiprocessing.

Ans:-  **Multiprogramming:** Running multiple programs simultaneously by switching between them.

**Multitasking:** The OS executes multiple tasks (processes or threads) concurrently.

**Multiprocessing:** Use of multiple processors (CPUs) to run tasks in parallel.

1. What is a file system, and what are its components?

Ans:- **:** A file system is a method for storing and organizing computer files. Its components include:

File control block (FCB): Information about files.

Directories: Organize files.

Data blocks: Store actual file data.

1. What is a deadlock, and how can it be prevented?

Ans:- A deadlock occurs when two or more processes are blocked forever, waiting for each other. It can be prevented by using techniques like resource allocation graphs or the Banker's algorithm.

1. Explain the difference between a kernel and a shell.

Ans:- **Kernel:** The core part of the OS that manages hardware and system resources.

**Shell:** A user interface that allows users to interact with the OS.

1. What is CPU scheduling, and why is it important?

Ans:- CPU scheduling is the method the OS uses to decide which process runs next on the CPU. It is important for optimizing CPU usage and system performance.

1. How does a system call work?

Ans:- A system call is a request from a program to the OS for services such as file manipulation, process control, or I/O operations

1. What is the purpose of device drivers in an operating system?

Ans:- **:** Device drivers are programs that enable the operating system to communicate with hardware devices (e.g., printers, hard drives).

1. Explain the role of the page table in virtual memory management.

Ans:- A page table maps virtual memory addresses to physical memory addresses, enabling efficient memory management.

1. What is thrashing, and how can it be avoided?

Ans:- Thrashing occurs when the OS spends more time swapping data between disk and memory than executing processes. It can be avoided by managing the degree of multiprogramming or using page replacement algorithms.

1. Describe the concept of a semaphore and its use in synchronization.

Ans:- A semaphore is a synchronization tool used to control access to a shared resource by multiple processes in concurrent programming.

1. How does an operating system handle process synchronization?

Ans:- The OS ensures that processes coordinate and avoid conflicts when accessing shared resources, typically using locks, semaphores, or monitors.

1. What is the purpose of an interrupt in operating systems?

Ans:- An interrupt is a signal to the processor indicating an event that needs immediate attention, allowing the OS to pause the current process and handle the interrupt.

1. Explain the concept of a file descriptor.

Ans:- A file descriptor is a unique identifier used by the OS to access an open file or other input/output resources.

1. How does a system recover from a system crash?

Ans:- After a system crash, the OS typically uses a technique like journaling to recover data or performs a file system check to restore consistency.

1. Describe the difference between a monolithic kernel and a microkernel.

Ans:-  **Monolithic Kernel:** A large, single kernel that directly controls hardware and system resources.

**Microkernel:** A smaller kernel that only provides essential services, with other services running in user space.

1. What is the difference between internal and external fragmentation?

Ans:- **Internal Fragmentation:** Wasted space within allocated memory blocks.

**External Fragmentation:** Wasted space between allocated memory blocks.

1. How does an operating system manage I/O operations?

Ans:- The OS manages I/O operations through device drivers, buffering, and scheduling to ensure efficient data transfer.

1. Explain the difference between preemptive and non-preemptive scheduling.

Ans\_; **Preemptive:** The OS can interrupt a running process to allocate CPU time to another process.

**Non-preemptive:** A process runs until it voluntarily relinquishes control

1. What is round-robin scheduling, and how does it work?

Ans:- Round-robin scheduling is a preemptive scheduling algorithm where each process is assigned a fixed time slice (quantum) to run before being swapped out.

1. Describe the priority scheduling algorithm. How is priority assigned to processes?

Ans:- Processes are assigned priorities, and the OS executes the process with the highest priority. Priorities can be static or dynamic.

1. What is the shortest job next (SJN) scheduling algorithm, and when is it used?

Ans:- **:** SJN schedules the process with the shortest burst time next. It is ideal for minimizing average waiting time but requires knowing process durations in advance.

1. Explain the concept of multilevel queue scheduling.

Ans:- Processes are divided into multiple queues based on priority or other criteria. Each queue may have its own scheduling algorithm.

1. What is a process control block (PCB), and what information does it contain?

Ans:- A PCB is a data structure containing information about a process, such as its state, program counter, CPU registers, memory limits, and I/O status.

1. Describe the process state diagram and the transitions between different process states.

Ans:- **:** A diagram illustrating the different states a process can be in (e.g., running, waiting, ready, terminated) and transitions between them.

1. How does a process communicate with another process in an operating system?

Ans:- Processes communicate through mechanisms such as pipes, message queues, shared memory, or sockets.

1. What is process synchronization, and why is it important?

Ans:- Process synchronization ensures that processes do not interfere with each other when accessing shared resources, typically using locks, semaphores, or condition variables.

1. Explain the concept of a zombie process and how it is created.

Ans:- A zombie process is a process that has completed execution but still has an entry in the process table. It occurs when the parent doesn't read the exit status.

1. Describe the difference between internal fragmentation and external fragmentation.

Ans:- **Internal Fragmentation:** Wasted space within allocated memory blocks.

**External Fragmentation:** Wasted space between allocated memory blocks.

1. What is demand paging, and how does it improve memory management efficiency?

Ans:- Demand paging is a memory management technique where pages are only loaded into memory when needed, improving memory efficiency.

1. Explain the role of the page table in virtual memory management.

Ans:- The MMU is a hardware component that maps virtual addresses to physical addresses using the page table, enabling virtual memory.

1. How does a memory management unit (MMU) work?

Ans:- A system call allows user programs to request services such as file manipulation, process control, or I/O operations from the OS.

1. What is thrashing, and how can it be avoided in virtual memory systems?

Ans:- **:** A race condition occurs when the outcome of a program depends on the order of execution of concurrent processes. It can be prevented using synchronization techniques like locks.

1. What is a system call, and how does it facilitate communication between user programs and the operating system?

Ans:- The fork() system call creates a new child process by duplicating the parent process.

1. Describe the difference between a monolithic kernel and a microkernel.

Ans:- When a process terminates, its resources are reclaimed, and the process state is set to terminated. The parent is notified through signals or the wait() system call.

1. How does an operating system handle I/O operations?

Ans:- The long-term scheduler decides which processes should be admitted into the ready queue, influencing the degree of multiprogramming

1. Explain the concept of a race condition and how it can be prevented.

Ans:- The parent process can use the wait() system call to wait for the child process to finish execution.

1. Describe the role of device drivers in an operating system.

Ans:- The exit status indicates whether a child process completed successfully or encountered an error

# 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.

1. 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.

1. 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.

1. 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.

1. 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.