* **PART A**

**What will the following commands do?**

**• echo "Hello, World!"**>> Prints Hello,World!

**• name="Productive"**  
>> Stores the value to name.  
   
**• touch file.txt**>> Creates a file named file.txt

**• ls -a**>> Lists all the files and directories with hidden files.

**• rm file.txt**>> removes the file.txt from the directory.

**• cp file1.txt file2.txt**>> Copies the 2 mentioned files.

**• mv file.txt /path/to/directory/**   
>> moves the file into the mentioned directory.

**• chmod 755 script.sh**   
>> It is used to change the permissions of the file. Octal numbers are used  
 7=owner has r,w,x permissions;5=group has r,-,x permissions;5= others has r,-,x permissions .

**• grep "pattern" file.txt**   
>> it is used to fetch specific words from a file. In this case pattern is obtained.

**• kill PID**>> Kills the processes/jobs of specific ID.

**• mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt**   
>> Step 1: creates a new directory named mydir.  
 Step 2: changes the directory.  
 Step 3 : creates a file named file.txt  
 Step 4: enter Hello World! in file.txt  
 Step 5: used to see the contents of the file.

**• ls -l | grep ".txt"**  
>> Step 1: lists all files and directories with all info  
 Step 2: finds all the files with .txt extension.

**• cat file1.txt file2.txt | sort | uniq**   
>> Combines the content of the 2 files and sorts the data alphabetically an filters out the duplicates.

**• ls -l | grep "^d"**   
>> lists all the files and directories with info.

**• grep -r "pattern" /path/to/directory/**   
>> recursively searches for “pattern” in the mentioned directory.

**• cat file1.txt file2.txt | sort | uniq –d**>> combines the content of the two files and sorts the

**• chmod 644 file.txt**>> changes the permissions of file.txt  
 6= owner has read and write.  
 4= group has read only.  
 4= others has read only.

**• cp -r source\_directory destination\_directory**>> It copies src directory and dest directory.

**• find /path/to/search -name "\*.txt"**   
>> Finds specific file with .txt extension.

**• chmod u+x file.txt**   
>> gives permission to user for execution.

• **echo $PATH**  
>> prints value saved to 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.**  
>>False

**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:**

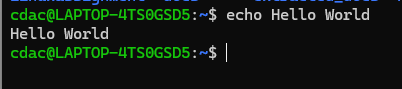
**1. chmodx is used to change file permissions.**  
>> chmod

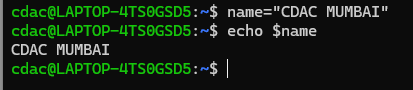
**2. cpy is used to copy files and directories.**>> cp

**3. mkfile is used to create a new file.**  
>> touch/nano

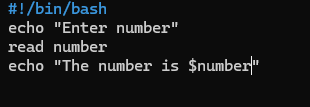
**4. catx is used to concatenate files.**   
>> cat

**5. rn is used to rename files.**>> mv

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

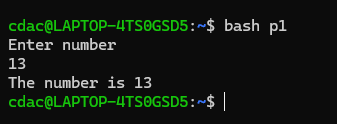
**Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.   
**

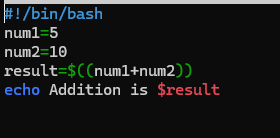
**Question 3: Write a shell script that takes a number as input from the user and prints it.**

****

**‘**

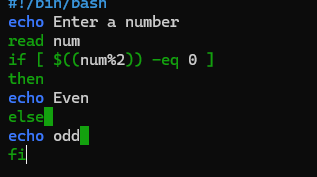
**Output**

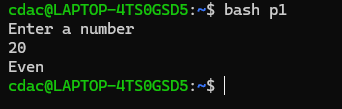
****

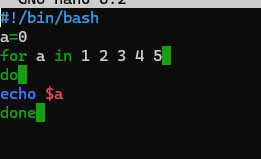
**Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.   
  
**

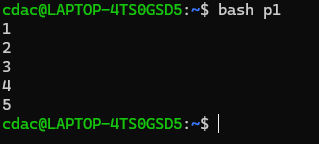
**output**

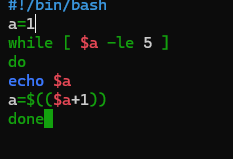
****

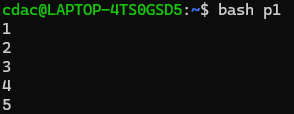
**Question 5: *Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".***

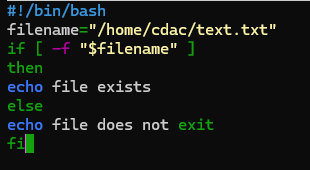
**Output:  
**

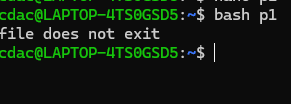
**Question 6: Write a shell script that uses a for loop to print numbers from 1 to 5.   
**

**Output:  
 **

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

**Output:  
**

**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".   
>>  
 Script:  
**

**Output:  
**

* **Part D :**

***Common Interview Questions (Must know)***

**1. What is an Operating System, and What are Its Primary Functions?**

* **Answer: An operating system (OS) is system software that manages computer hardware and software resources and provides common services for computer programs. Its primary functions include process management, memory management, file system management, device management, and providing a user interface.**

**2. Explain the Difference Between Process and Thread.**

* **Answer: A process is an instance of a program in execution, whereas a thread is a smaller unit of a process that can be scheduled and executed independently. Threads within the same process share resources like memory, while processes have separate memory spaces.**

**3. What is Virtual Memory, and How Does It Work?**

* **Answer: Virtual memory is a memory management technique that gives an application the impression of a large contiguous block of memory while actually using smaller, fragmented physical memory and disk space. It uses techniques like paging and segmentation.**

**4. Describe the Difference Between Multiprogramming, Multitasking, and Multiprocessing.**

* **Answer:**
  + **Multiprogramming: Running multiple programs simultaneously by managing their execution.**
  + **Multitasking: Extends multiprogramming by allowing multiple tasks to run concurrently on a single CPU.**
  + **Multiprocessing: Using multiple CPUs to execute multiple processes simultaneously.**

**5. What is a File System, and What are Its Components?**

* **Answer: A file system organizes and manages files and directories on storage devices. Its components include files, directories, file descriptors, and metadata. It also defines how data is stored and retrieved.**

**6. What is a Deadlock, and How Can It Be Prevented?**

* **Answer: A deadlock is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource that another process holds. Deadlock prevention can be done by ensuring that at least one of the necessary conditions for deadlock (mutual exclusion, hold and wait, no preemption, circular wait) is not met.**
* **Book Reference: *Operating System Concepts* (Chapter 7 - Deadlocks)**

**7. Explain the Difference Between a Kernel and a Shell.**

* **Answer: The kernel is the core part of an OS that directly interacts with hardware and manages resources. The shell is a user interface that allows users to interact with the kernel through commands.**

**8. What is CPU Scheduling, and Why is It Important?**

* **Answer: CPU scheduling is the process of determining which processes will run on the CPU and for how long. It is crucial for ensuring efficient CPU utilization and system responsiveness.**

**9. How Does a System Call Work?**

* **Answer: A system call provides an interface between a user program and the operating system. It allows user programs to request services from the OS, such as I/O operations or memory allocation.**

**10. What is the Purpose of Device Drivers in an Operating System?**

* **Answer: Device drivers act as intermediaries between the OS and hardware devices, translating OS instructions into device-specific commands.**

**11. Explain the Role of the Page Table in Virtual Memory Management.**

* **Answer: The page table is a data structure used in virtual memory systems to map virtual addresses to physical addresses. Each entry in the page table corresponds to a virtual page and contains the address of the corresponding physical page, along with other status information like access rights and whether the page is in memory or on disk.**

**12. What is Thrashing, and How Can It Be Avoided?**

* **Answer: Thrashing occurs when a system spends more time swapping pages in and out of memory than executing actual processes, leading to severe performance degradation. It can be avoided by using techniques like working set model, adjusting the degree of multiprogramming, or by using better page replacement algorithms.**

**13. Describe the Concept of a Semaphore and Its Use in Synchronization.**

* **Answer: A semaphore is a synchronization primitive used to control access to a common resource in concurrent programming. It can be used to solve problems like mutual exclusion and producer-consumer issues. Semaphores are counters that can be incremented or decremented atomically, ensuring that no two processes access the critical section simultaneously.**

**14. How Does an Operating System Handle Process Synchronization?**

* **Answer: Process synchronization is handled by using various synchronization primitives like semaphores, mutexes, and monitors. These tools help coordinate the order in which processes execute to prevent race conditions and ensure that shared resources are accessed in a controlled manner.**

**15. What is the Purpose of an Interrupt in Operating Systems?**

* **Answer: Interrupts allow the CPU to respond to events such as I/O operations or hardware malfunctions immediately, by pausing the current process, saving its state, and executing an interrupt service routine (ISR). This mechanism ensures efficient and responsive system operation.**

**16. Explain the Concept of a File Descriptor.**

* **Answer: A file descriptor is a low-level handle assigned to a file or other I/O resource, such as a pipe or network socket, used by processes to access the resource. Each process maintains a table of file descriptors, which the OS uses to identify the open files.**

**17. How Does a System Recover from a System Crash?**

* **Answer: Recovery from a system crash involves several steps, such as checking and restoring file system integrity, rolling back incomplete transactions, and rebooting the system. Techniques like journaling in file systems help minimize data loss and ensure consistency.**

**18. Describe the Difference Between a Monolithic Kernel and a Microkernel.**

* **Answer:**
  + **Monolithic Kernel: Includes all the essential OS services like process management, memory management, and device drivers within a single large kernel.**
  + **Microkernel: Only includes the most essential functions in the kernel, like communication between processes and basic I/O operations, while other services run in user space, enhancing modularity and security.**

**19. What is the Difference Between Internal and External Fragmentation?**

* **Answer:**
  + **Internal Fragmentation: Occurs when allocated memory may be slightly larger than requested memory, leading to unused memory within allocated regions.**
  + **External Fragmentation: Occurs when free memory is scattered in small blocks across the system, making it difficult to allocate large contiguous blocks.**

**20. How Does an Operating System Manage I/O Operations?**

* **Answer: The OS manages I/O operations by abstracting hardware specifics through device drivers, scheduling I/O tasks, and handling interrupts. The OS also provides interfaces to user programs for reading and writing data, ensuring efficient and fair access to I/O devices.**

**21. Explain the Difference Between Preemptive and Non-Preemptive Scheduling.**

* **Answer:**
  + **Preemptive Scheduling: Allows the OS to interrupt a running process and switch to another process, ensuring responsive multitasking.**
  + **Non-Preemptive Scheduling: Once a process starts running, it continues until it finishes or voluntarily yields control, reducing context switching overhead.**

**22. What is Round-Robin Scheduling, and How Does It Work?**

* **Answer: Round-Robin scheduling is a preemptive CPU scheduling algorithm where each process is assigned a fixed time slice (quantum). The CPU cycles through all processes, assigning each one the CPU for a time slice in a cyclic order, ensuring fairness.**

**23. Describe the Priority Scheduling Algorithm. How is Priority Assigned to Processes?**

* **Answer: Priority scheduling assigns the CPU to the process with the highest priority. Priorities can be assigned based on factors like process importance, required resources, or user-defined criteria. The algorithm can be preemptive or non-preemptive.**

**24. What is the Shortest Job Next (SJN) Scheduling Algorithm, and When is It Used?**

* **Answer: SJN is a non-preemptive scheduling algorithm that selects the process with the smallest execution time to run next. It minimizes average waiting time but can lead to starvation of longer processes.**

**25. Explain the Concept of Multilevel Queue Scheduling.**

* **Answer: Multilevel queue scheduling involves dividing the ready queue into several smaller queues, each with different priority levels or scheduling algorithms. Processes are permanently assigned to a queue based on some criteria (e.g., process type or priority).**

**26. What is a Process Control Block (PCB), and What Information Does It Contain?**

* **Answer: A PCB is a data structure maintained by the OS that contains all the information about a process, including process state, program counter, CPU registers, memory limits, and I/O status. It is essential for context switching.**

**27. Describe the Process State Diagram and the Transitions Between Different Process States.**

* **Answer: A process typically moves through states such as New, Ready, Running, Waiting, and Terminated. The OS manages transitions between these states based on events like process creation, I/O completion, and CPU scheduling.**

**28. How Does a Process Communicate with Another Process in an Operating System?**

* **Answer: Processes communicate using Inter-Process Communication (IPC) mechanisms like message passing, shared memory, pipes, and signals. These mechanisms allow data exchange and synchronization between processes.**

**29. What is Process Synchronization, and Why is It Important?**

* **Answer: Process synchronization ensures that multiple processes or threads can execute concurrently without causing data inconsistency or race conditions. It is crucial for maintaining data integrity and coordination between processes.**

**30. Explain the Concept of a Zombie Process and How It Is Created.**

* **Answer: A zombie process is a process that has completed execution but still has an entry in the process table. It is created when a child process terminates, but its parent has not yet read its exit status using wait(), leaving the process descriptor in a defunct state.**

**31. Describe the Difference Between Internal Fragmentation and External Fragmentation.**

* **Answer:**
  + **Internal Fragmentation: Wasted space within allocated memory blocks.**
  + **External Fragmentation: Wasted space between allocated memory blocks that cannot be used for new allocations due to small size.**

**32. What is Demand Paging, and How Does It Improve Memory Management Efficiency?**

* **Answer: Demand paging loads pages into memory only when they are needed, reducing memory usage and allowing more processes to execute.**

**33. Explain the Role of the Page Table in Virtual Memory Management.**

* **Answer: The page table plays a crucial role in virtual memory management by mapping virtual addresses to physical addresses. Each entry in the page table holds the frame number that corresponds to a page in memory. The page table allows the operating system to translate virtual addresses generated by the CPU into physical addresses used by the hardware. This mapping is essential for implementing paging, which enables efficient memory utilization.**

**34. How Does a Memory Management Unit (MMU) Work?**

* **Answer: The Memory Management Unit (MMU) is a hardware component responsible for handling virtual to physical address translation. It uses the page table to translate virtual addresses to physical addresses. The MMU checks the page table entry for the corresponding page and retrieves the physical frame number to complete the address translation. The MMU also plays a role in enforcing access controls and managing page faults.**

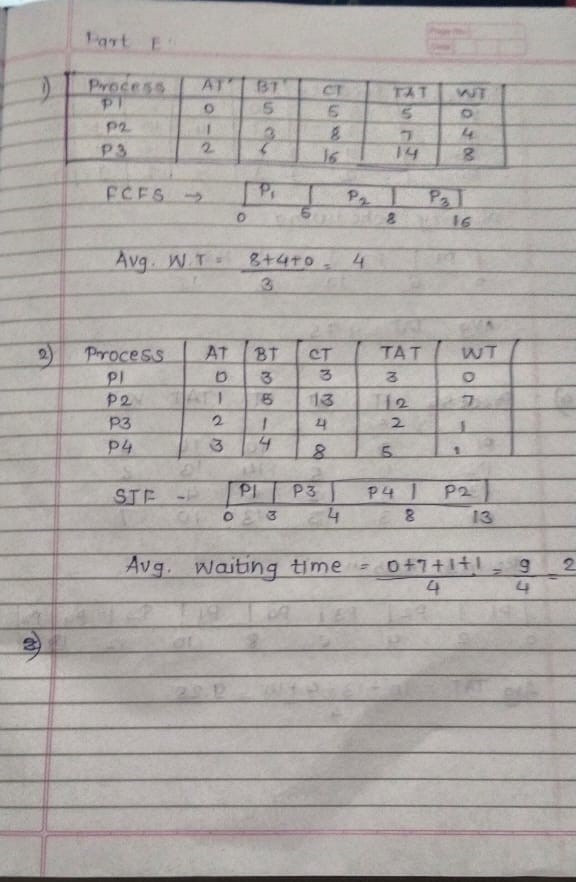
**35. What is Thrashing, and How Can It Be Avoided in Virtual Memory Systems?**

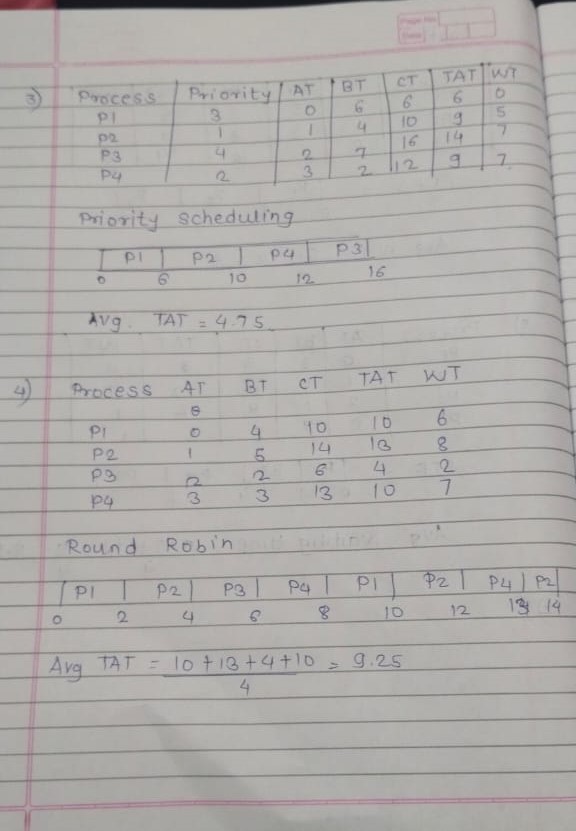
* **Answer: Thrashing occurs when a system spends more time paging (swapping pages in and out of memory) than executing actual processes. It can significantly degrade system performance. Thrashing can be avoided by using working set models to keep track of actively used pages, reducing the degree of multiprogramming, or employing better page replacement algorithms to minimize unnecessary page swaps.**

**36. What is a System Call, and How Does It Facilitate Communication Between User Programs and the Operating System?**

* **Answer: A system call is a mechanism that allows user-level programs to request services from the operating system. When a program needs to perform an operation that requires OS intervention, such as file access or process control, it makes a system call. The OS then executes the requested service and returns the result to the program. System calls provide a controlled interface between user programs and kernel functions.**

**Part E**

****

****