PART A-PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS

Q1 Contrast the scheduling policies you might use when trying to optimize a time-sharing system with those you would use to optimize a multiprogrammed batch system.

Ans:**Time-Sharing System**

1. **Objective**: Minimize response time to ensure interactive user experience.

2. Scheduling Policy
Round**-Robin**: Each process gets a fixed time slice (quantum) to execute. If it doesn't finish within this time, it goes back to the queue.

    - **Priority Scheduling**: Processes are assigned priorities, and the CPU is allocated to the highest priority process. Lower priority processes may have to wait longer.
    - **Multilevel Queue Scheduling**: Processes are divided into different queues based on their priority or type (e.g., system processes, interactive processes), and each queue has its own scheduling algorithm.
2. **Context Switching**: Frequent context switching to ensure all users get a fair share of CPU time.
3. **User Interaction**: High, as the system needs to be responsive to user inputs.

Multiprogrammed Batch System

1. **Objective**: Maximize CPU utilization and throughput.
2. **Scheduling Policy**:
    - **First-Come, First-Served (FCFS)**: Processes are executed in the order they arrive. Simple but can lead to the "convoy effect" where short processes wait for long ones.
    - **Shortest Job Next (SJN)**: The process with the shortest execution time is selected next. This minimizes average waiting time but requires knowing the execution time in advance.
    - **Priority Scheduling**: Similar to time-sharing but focused on job characteristics like CPU burst time, I/O requirements, etc.
    - **Multilevel Queue Scheduling**: Similar to time-sharing but queues might be based on job types (e.g., CPU-bound vs. I/O-bound).
3. **Context Switching**: Less frequent compared to time-sharing, as the focus is on completing jobs efficiently.

4. **User Interaction**: Low, as jobs are typically submitted and executed without user intervention.

Q2) Distinguish between hard real time systems and soft real time systems

Ans:

HARD REAL TIME SYSTEM	SOFT REAL TIME SYSTEM
In hard real time system, the size of data file is small or medium.	In soft real time system, the size of data file is large.
In this system response time is in millisecond.	In this system response time are higher.
Peak load performance should be predictable.	In soft real time system, peak load can be tolerated.
In this system safety is critical.	In this system safety is not critical.
A hard real time system is very restrictive.	A Soft real time system is less restrictive.
In case of an error in a hard real time system, the computation is rolled back.	In case of an soft real time system, computation is rolled back to previously established a checkpoint.
Satellite launch, Railway signaling system etc.	DVD player, telephone switches, electronic games etc.
Guarantees response within a specific deadline.	Does not guarantee response within a specific deadline.

Q3) Consider a computer system that has cache memory, main memory (RAM) and disk, and an operating system that uses virtual memory. It takes 1 nsec to access a word from the cache, 10 nsec to access a word from the RAM, and 10 ms to access a word from the disk. If the cache hit rate is 95 percent and main memory hit rate (after a cache miss) is 99 percent, what is the average time to access a word?

Ans: **Memory Hierarchy and Access Times**

1. Cache Memory:
    - **Access Time**: 1 nanosecond (nsec)

    - **Purpose**: Cache memory is the fastest type of memory in a computer system. It stores frequently accessed data and instructions to speed up processing.
2. Main Memory (RAM):
    - **Access Time**: 10 nanoseconds (nsec)
    - **Purpose**: RAM is the primary memory used by the CPU to store data that is actively being used or processed. It is slower than cache but faster than disk storage.
3. Disk Storage:
    - **Access Time**: 10 milliseconds (ms)
    - **Purpose**: Disk storage (such as HDD or SSD) is used for long-term storage of data. It is much slower than both cache and RAM.

Virtual Memory

Virtual memory is a memory management technique used by operating systems to extend the apparent amount of physical memory available. It allows the system to use a portion of the disk as if it were additional RAM1. This is achieved through a process called paging, where the operating system moves data between RAM and disk storage as needed2.

How It Works

1. **Paging**: When the system runs out of RAM, it moves some of the data to a special file on the disk called the page file or swap space. This process is known as paging3.
2. **Address Translation**: The Memory Management Unit (MMU) in the CPU translates virtual addresses to physical addresses, allowing programs to use more memory than is physically available2.

Q4) What is a distributed operating system? What are the advantages of a distributed operating system?
Ans: DOS: ● A distributed operating system (DOS) is an essential type of operating system. Distributed systems use many central processors to serve multiple real-time applications and users. As a result, data processing jobs are distributed between the processors.

● It connects multiple computers via a single communication channel. Furthermore, each of these systems has its own processor and memory. Additionally, these CPUs communicate via high-speed buses or telephone lines. Individual systems that communicate via a single channel are regarded as a single entity. They're also known as loosely coupled systems.

● This operating system consists of numerous computers, nodes, and sites joined together via LAN/WAN lines. It enables the distribution of full systems on a couple of centre processors, and it supports many real-time products and different users. Distributed operating systems can share their computing resources and I/O files while providing users with virtual machine abstraction.

Advantages: There are various advantages of the distributed operating system. Some of them are as follow:
● It may share all resources (CPU, disk, network interface, nodes, computers, and so on) from one site to another, increasing data availability across the entire system.

● It reduces the probability of data corruption because all data is replicated across all sites; if one site fails, the user can access data from another operational site.

● The entire system operates independently of one another, and as a result, if one site crashes, the entire system does not halt.

● It increases the speed of data exchange from one site to another site.

● It is an open system since it may be accessed from both local and remote locations.

● It helps in the reduction of data processing time

. ● Most distributed systems are made up of several nodes that interact to make them fault-tolerant. If a single machine fails, the system remains operational.

Q5) What is the purpose of system calls, and how do system calls relate to the OS and to the concept of dual-mode (kernel-mode and user-mode) operation?

Ans: System calls are essential for enabling user-level processes to request services from the operating system (OS). They act as the interface between user applications and the OS, allowing programs to perform tasks such as file operations, memory allocation, and process control1.

Purpose of System Calls

1. **Resource Management**: System calls manage hardware resources like CPU, memory, and I/O devices.
2. **Process Control**: They handle process creation, termination, and synchronization.
3. **File Management**: System calls manage file operations like reading, writing, and closing files.
4. **Communication**: They facilitate inter-process communication and networking.

Relation to OS and Dual-Mode Operation

The concept of dual-mode operation in an OS involves two distinct modes: user mode and kernel mode23.

- **User Mode**: In this mode, user applications run with limited privileges. They cannot directly access hardware or critical system resources to prevent accidental or malicious damage.
- **Kernel Mode**: This mode has unrestricted access to all system resources. The OS runs in kernel mode to execute privileged instructions and manage hardware.

When a user application needs to perform a task that requires higher privileges (like accessing hardware), it makes a system call.

Q6) Explain the difference between interrupt and exception.

Ans: Interrupt is one of the classes of Exception. There are 4 classes of Exception Interrupt, trap, fault and abort. Though, interrupt belongs to exceptions still there are many differences between them.

Interrupt: The term Interrupt is usually reserved for hardware interrupts. They are program control interruptions caused by external hardware events. Here, external means external to the CPU. Hardware interrupts usually come from many different sources such as timer chip, peripheral devices (keyboards, mouse, etc.), I/O ports (serial, parallel, etc.), disk drives, CMOS clock, expansion cards (sound card, video card, etc). That means hardware interrupts almost never occur due to some event related to the executing program.

Exception: Exception is a software interrupt, which can be identified as a special handler routine. Exceptions can be identified as an automatically occurring trap. Generally, there are no specific instructions associated with exceptions (traps are generated using a specific instruction). So, an exception occurs due to an “exceptional” condition that occurs during program execution

Interrupt	Exception
These are Hardware interrupts.	These are Software interrupts.
Occurrences of hardware interrupts usually disable other hardware interrupts.	This is not a true case in terms of Exception.

These are asynchronous external requests for service (like keyboard or printer needs service).	These are synchronous internal requests for service based upon abnormal events (think of illegal instructions, illegal address, overflow etc).
Being asynchronous, interrupts can occur at any place in the program.	Being synchronous, exceptions occur when there is an abnormal event in your program like, divide by zero or illegal memory location.
These are normal events and shouldn't interfere with the normal running of a computer.	These are abnormal events and often result in the termination of a program

Q7) It is sometimes difficult to achieve a layered approach if two components of the operating system are dependent on each other. Identify a scenario in which it is unclear how to layer two system components that require tight coupling of their functionalities.

Ans: The **virtual memory subsystem** and the **storage subsystem** are typically tightly coupled and require careful design in a layered system due to the following interactions. Many **systems allow files to be mapped into the virtual memory space of an executing process**. **On the other hand, the virtual memory subsystem typically uses the storage system to provide the backing store for pages that do not currently reside in memory.**

Also, updates to the file system are sometimes buffered in **physical memory** before it is **flushed to disk**, thereby requiring **careful coordination of the usage of memory between the virtual memory subsystem and the file system**.

Q8) Explain Is OS is a resource manager. If so justify your answer

Ans: Operating system is a resource allocator which manages all resources and decides between conflicting requests for efficient and fair resources.

● The operating system provides for an orderly and controlled allocation of the processors, memories, and I/O devices among the various programs in the bottom-up view.

● Now-a-days all modern computers consist of processors, memories, timers, network interfaces, printers, and so many other devices.

● Operating system allows multiple programs to be in memory and run at the same time. Resource management includes multiplexing or sharing resources in two different ways: in time and in space.

● In time multiplexed, different programs take a chance of using CPU. First one tries to use the resource, then the next one that is ready in the queue and so on. For example: Sharing the printer one after another.

● In space multiplexing, Instead of the customers taking a chance, each one gets part of the resource. For example – Main memory is divided into several running programs, so each one can be resident at the same time.

Q9) How could a system be designed to allow a choice of operating systems from which to boot?What would the bootstrap program need to do?

Ans: Consider a system that would like to run both Windows XP and three different distributions of Linux (e.g., RedHat, Debian, and Mandrake). Each operating system will be stored on disk. During system boot-up, a special program (which we will call the boot manager) will determine which operating system to boot into. This means that rather initially booting to an operating system, the boot manager will first run during system startup. It is this boot manager that is responsible for determining which system to boot

into. Typically boot managers must be stored at certain locations of the hard disk to be recognized during system startup. Boot managers often provide the user with a selection of systems to boot into; boot managers are also typically designed to boot into a default operating system if no choice is selected by the user

Q10) Define essential properties of the following types of Operating system: i) Batch operating system ii) Interactive operating system iii) Time sharing operating system iv) Real time operating system v) Distributed operating system.

Ans: i) Batch operating system In a Batch operating system, the user does not have direct access to the computer and cannot directly interact with it either. In this type of OS, jobs are prepared for each user, and all those jobs have been imprinted or stored in the punch card-like structure, which is submitted to the computer operator. This kind of operating system mostly works on offline devices and once the punch card is submitted to the computer operator, the computer works according to the code or program written on the card.

ii) Interactive operating system In an Interactive operating system, there is a direct interaction between the user and the computer. Mostly, all personal computers use Interactive operating systems. In this kind of operating system, the user enters some command in the system and the system works according to it.

iii) Time sharing operating system It is similar to the multiprogramming system with some additional extensions and also known as Multitasking OS. In a Time sharing OS, the system is capable of handling multiple jobs simultaneously and here the processing time is shared among all the users. With Time sharing OS, users at different locations or terminals can access the same computer at the same time. Here, the CPU uses the switching mechanism that helps it to switch from one job to another so that each job gets equal processing time.
iv) Real time operating system An RTOS is a data processing system whose response time to the input is very short. RTOS is also known as the brain of the real-time system because of its immediate response to the input.

The response to the input in RTOS is displayed in a specific time period. Though the time period is very short it does not show any kind of disparity. There are 2 types of Real-Time operating systems

● Hard Real-Time System: In a Hard Real-Time system, if the response takes more time than the specified time interval, the system will fail. The secondary storage is also limited in these systems.

● Soft Real-Time System: The Soft Real-Time system does not fail the program even if the response takes more time than the specified time. It would just show the output, however, it can compromise the accuracy of the response.

v) Distributed operating system In this operating system, different computers interact with one another and communicate in order to exchange data. The Internet works on this system where everyone is linked with each other to communicate. It can also be termed as the Networking operating system, which supports a high level of communication

A Distributed OS uses multiple processors to perform multiple real-time applications on the user terminal. In this system, the processor does not share the memory because each CPU has its own local memory.