

TOP 50

OPERATING SYSTEM

INTERVIEW QUESTION

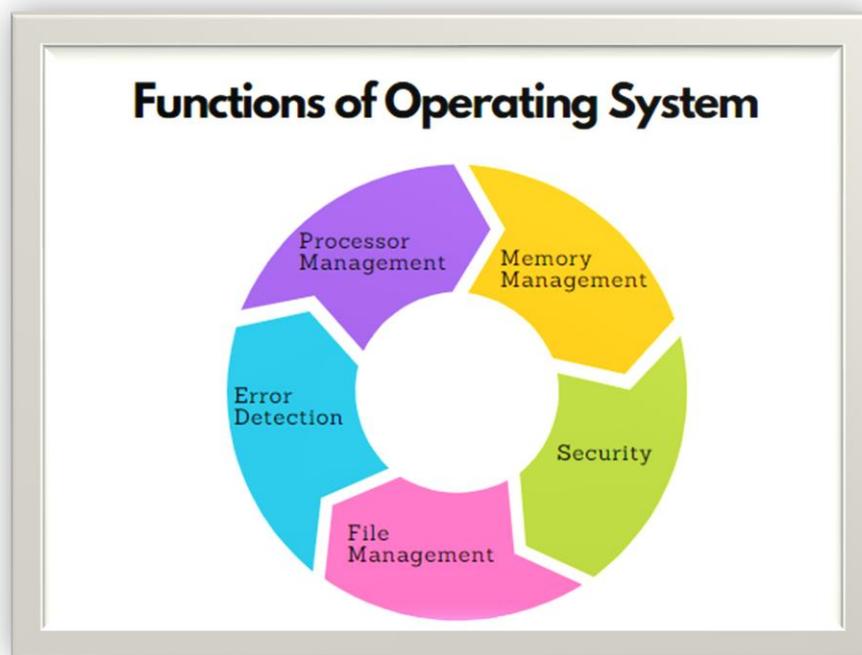


Created by- **TOPPERWORLD**

Q 1. Explain the main purpose of an operating system?

Ans : An operating system acts as an intermediary between the user of a computer and computer hardware.

The purpose of an operating system is to provide an environment in which a user can execute programs conveniently and efficiently. An operating system is a software that manages computer hardware.

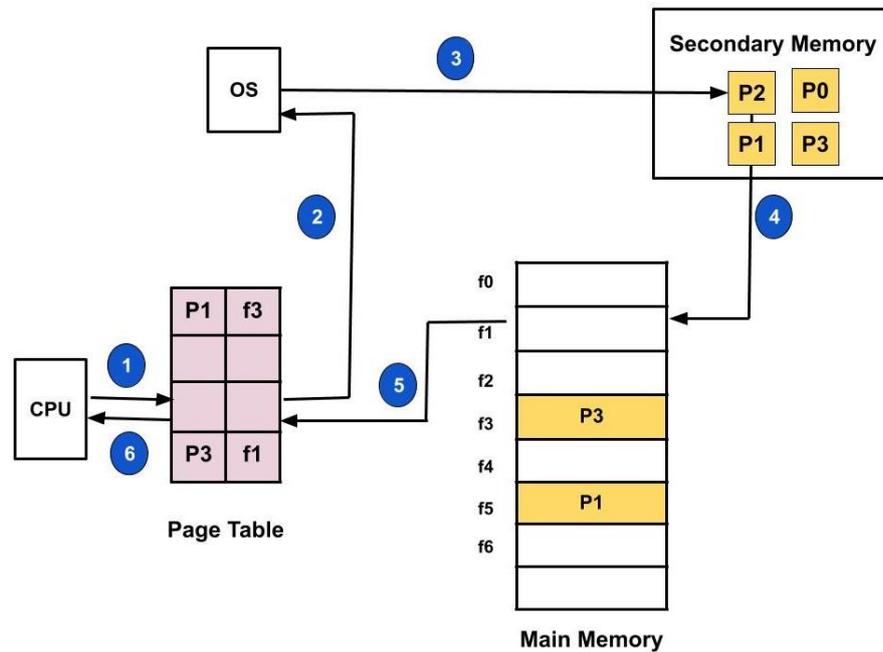


The hardware must provide appropriate mechanisms to ensure the correct operation of the computer system and to prevent user programs from interfering with the proper operation of the system.

Q 2 . What is demand paging?

Ans : Demand paging can be described as a memory management technique that is used in operating systems to improve memory usage and system performance.

Demand paging is a technique used in virtual memory systems where pages enter main memory only when requested or needed by the CPU.



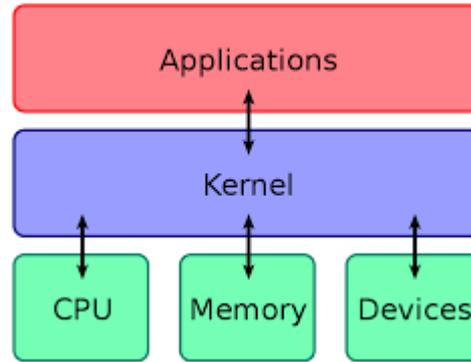
In demand paging, the operating system loads only the necessary pages of a program into memory at runtime, instead of loading the entire program into memory at the start.

Q 3 . What is a kernel?

Ans : A kernel is the central component of an operating system that manages the operations of computers and hardware.

It basically manages operations of memory and CPU time.





It is a core component of an operating system.

Kernel acts as a bridge between applications and data processing performed at the hardware level using inter-process communication and system calls.

Q 4 . What are the different scheduling algorithms?

Ans :

- 1) First-Come, First-Served (FCFS) Scheduling.
- 2) Shortest-Job-Next (SJN) Scheduling.
- 3) Priority Scheduling.
- 4) Shortest Remaining Time.
- 5) Round Robin(RR) Scheduling.
- 6) Multiple-Level Queues Scheduling.

Q 5 . Describe the objective of multi-programming.

Ans : Multi-programming increases CPU utilization by organizing jobs (code and data) so that the CPU always has one to execute.

The main objective of multi-programming is to keep multiple jobs in the main memory.

If one job gets occupied with IO, the CPU can be assigned to other jobs.

Q 6 . What is the time-sharing system?

Ans : Time-sharing is a logical extension of multiprogramming.

The CPU performs many tasks by switches that are so frequent that the user can interact with each program while it is running.

A time-shared operating system allows multiple users to share computers simultaneously.

Q 7. What problem we face in computer system without OS?

Ans :

- Poor resource management
- Lack of User Interface
- No File System
- No Networking
- Error handling is big issue etc.

Q 8 . Give some benefits of multithreaded programming?

Ans : A thread is also known as a lightweight process. The idea is to achieve parallelism by dividing a process into multiple threads.

- 1) **Responsiveness** – Multithreading in an interactive application may allow a program to continue running even if a part of it is blocked or is performing a lengthy operation, thereby increasing responsiveness to the user. In a non multi threaded environment, a server listens to the port for some request and when the request comes, it processes the request and



then resume listening to another request. The time taken while processing of request makes other users wait unnecessarily. Instead a better approach would be to pass the request to a worker thread and continue listening to port.

2) Resource Sharing – Processes may share resources only through techniques such as-

- Message Passing
- Shared Memory

Such techniques must be explicitly organized by programmer. However, threads share the memory and the resources of the process to which they belong by default. The benefit of sharing code and data is that it allows an application to have several threads of activity within same address space.

3) Economy – Allocating memory and resources for process creation is a costly job in terms of time and space. Since, threads share memory with the process it belongs, it is more economical to create and context switch threads. Generally much more time is consumed in creating and managing processes than in threads. In Solaris, for example, creating process is 30 times slower than creating threads and context switching is 5 times slower.

4) Scalability – The benefits of multi-programming greatly increase in case of multiprocessor architecture, where threads may be running parallel on multiple processors. If there is only one thread then it is not possible to divide the processes into smaller tasks that different processors can perform. Single threaded process can run only on one processor regardless of how many processors are available. Multi-threading on a multiple CPU machine increases parallelism.

5) Better Communication System – To improve the inter-process communication, thread synchronization functions can be used. Also, when need to share huge amounts of data across multiple threads of execution inside the same address space then provides extremely high bandwidth and low communication across the various tasks within the application.

6) Microprocessor Architecture Utilization – Every thread could be execute in parallel on a distinct processor which might be considerably amplified in a microprocessor architecture. Multithreading enhances concurrency on a multi CPU machine. Also the CPU switches among threads very quickly in a single processor architecture where it creates the illusion of parallelism, but at a particular time only one thread can running.

Q 9 . Briefly explain FCFS.

Ans : FCFS stands for First Come First served. In the FCFS scheduling algorithm, the job that arrived first in the ready queue is allocated to the CPU and then the job that came second and so on.

FCFS is a non-preemptive scheduling algorithm as a process that holds the CPU until it either terminates or performs I/O.

Thus, if a longer job has been assigned to the CPU then many shorter jobs after it will have to wait.

Q 10 . What is the RR scheduling algorithm?

Ans : A round-robin scheduling algorithm is used to schedule the process fairly for each job in a time slot or quantum and interrupting the job if it is not completed by then the job comes after the other job which is arrived in the quantum time makes these scheduling fairly.

- ◆ Round-robin is cyclic in nature, so starvation doesn' t occur
- ◆ Round-robin is a variant of first-come, first-served scheduling
- ◆ No priority or special importance is given to any process or task
- ◆ RR scheduling is also known as Time slicing scheduling

Q 11 . Enumerate the different RAID levels?

Ans : A redundant array of independent disks is a set of several physical disk drives that the operating system sees as a single logical unit. It played a significant role in narrowing the gap between increasingly fast processors and slow disk drives. RAID has different levels:

- Level-0
- Level-1
- Level-2
- Level-3
- Level-4
- Level-5
- Level-6

Q 12. What is Banker’ s algorithm?

Ans : The banker ’ s algorithm is a resource allocation and deadlock avoidance algorithm that tests for safety by simulating the allocation for the predetermined maximum possible amounts of all resources, then makes an “s-state” check to test for possible activities, before deciding whether allocation should be allowed to continue.

Q 13 . State the main difference between logical and physical address space?

Ans :

Parameter	LOGICAL ADDRESS	PHYSICAL ADDRESS
Basic	generated by the CPU.	location in a memory unit.
Address Space	Logical Address Space is a set of all logical addresses	Physical Address is a set of all physical addresses mapped to

Parameter	LOGICAL ADDRESS	PHYSICAL ADDRESS
	generated by the CPU in reference to a program.	the corresponding logical addresses.
Visibility	Users can view the logical address of a program.	Users can never view the physical address of the program.
Generation	generated by the CPU.	Computed by MMU.
Access	The user can use the logical address to access the physical address.	The user can indirectly access physical addresses but not directly.

Q 14 . How does dynamic loading aid in better memory space utilization?

Ans : With dynamic loading, a routine is not loaded until it is called. This method is especially useful when large amounts of code are needed in order to handle infrequently occurring cases such as error routines.



Q 15 . What are overlays?

Ans : The concept of overlays is that whenever a process is running it will not use the complete program at the same time, it will use only some part of it.

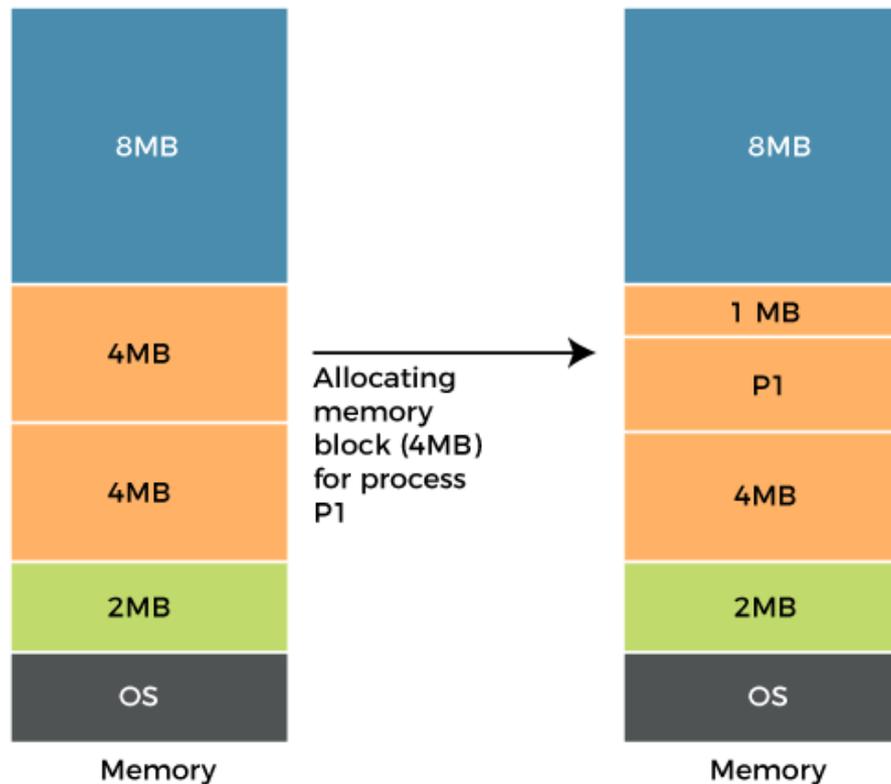
Then overlay concept says that whatever part you required, you load it and once the part is done, then you just unload it, which means just pull it back and get the new part you required and run it.

Formally, “The process of transferring a block of program code or other data into internal memory, replacing what is already stored” .

Q 16 . What is fragmentation?

Ans : Processes are stored and removed from memory, which makes free memory space, which is too little to even consider utilizing by different processes. Suppose, that process is not ready to dispense to memory blocks since its little size and memory hinder consistently staying unused is called fragmentation. This kind of issue occurs during a dynamic memory allotment framework when free blocks are small, so it can ’ t satisfy any request.





Q 17 . What is the basic function of paging?

Ans : Paging is a method or technique which is used for non-contiguous memory allocation. It is a fixed-size partitioning theme (scheme). In paging, both main memory and secondary memory are divided into equal fixed-size partitions. The partitions of the secondary memory area unit and the main memory area unit are known as pages and frames respectively.

Paging is a memory management method accustomed fetch processes from the secondary memory into the main memory in the form of pages. in paging, each process is split into parts wherever the size of every part is the same as the page size. The size of the last half could also be but the page size. The pages of the process area unit hold on within the frames of main memory relying upon their accessibility.

Q 18 . How does swapping result in better memory management?

Ans : Swapping is a simple memory/process management technique used by the operating system(os) to increase the utilization of the processor by moving some blocked processes from the main memory to the secondary memory thus forming a queue of the temporarily suspended processes and the execution continues with the newly arrived process.

During regular intervals that are set by the operating system, processes can be copied from the main memory to a backing store and then copied back later.

Swapping allows more processes to be run that can fit into memory at one time.



Q

19 . Write a name of classic synchronization problems?

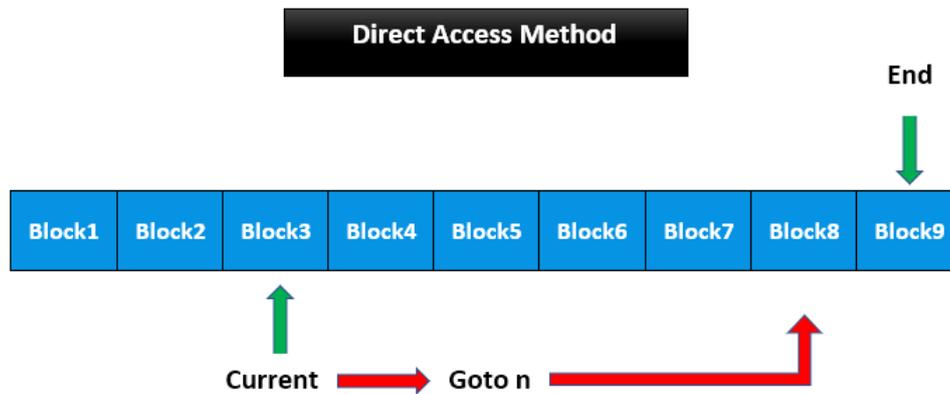
Ans :

- Bounded-buffer
- Readers-writers
- Dining philosophers
- Sleeping barber

Q 20 . What is the Direct Access Method?

Ans : The direct Access method is based on a disk model of a file, such that it is viewed as a numbered sequence of blocks or records. It allows arbitrary blocks to be read or written.

Direct access is advantageous when accessing large amounts of information. Direct memory access (DMA) is a method that allows an input/output (I/O) device to send or receive data directly to or from the main memory, bypassing the CPU to speed up memory operations.



The process is managed by a chip known as a DMA controller (DMAC).

Q 21 . When does thrashing occur?

Ans : Thrashing occurs when processes on the system frequently access pages, not available memory.

Q 22. What is the best page size when designing an operating system?

Ans : The best paging size varies from system to system, so there is no single best when it comes to page size.

There are different factors to consider in order to come up with a suitable page size, such as page table, paging time, and its effect on the overall efficiency of the operating system.



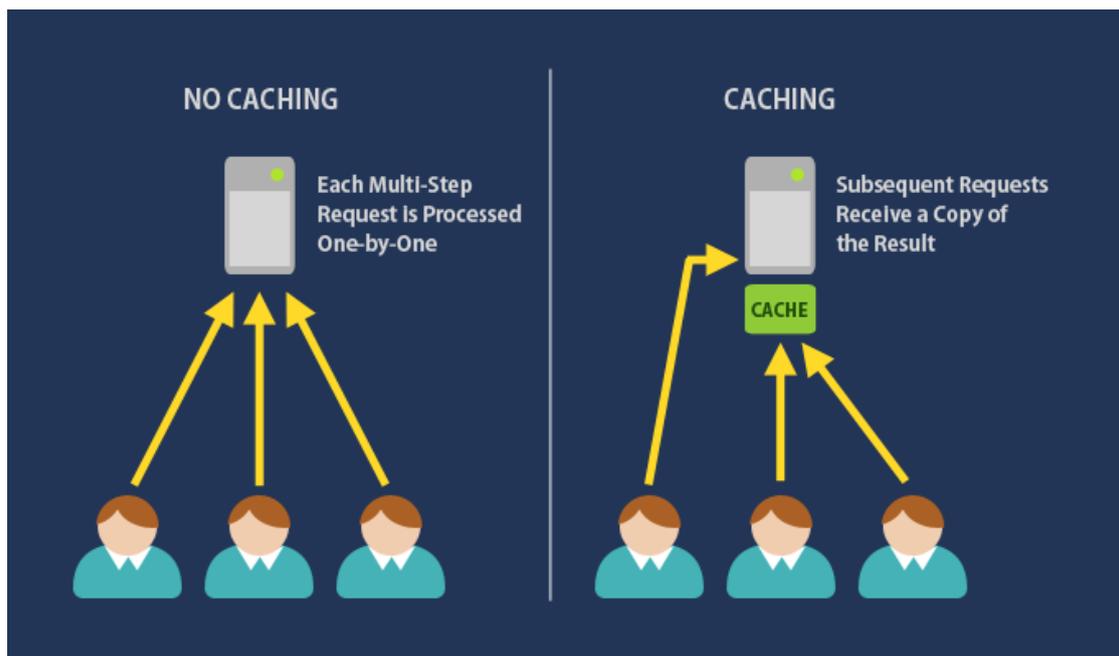
Q 23 . What is multitasking?

Ans : Multitasking is a logical extension of a multiprogramming system that supports multiple programs to run concurrently.

In multitasking, more than one task is executed at the same time. In this technique, the multiple tasks, also known as processes, share common processing resources such as a CPU.

Q 24 . What is caching?

Ans : The cache is a smaller and faster memory that stores copies of the data from frequently used main memory locations.



There are various different independent caches in a CPU, which store instructions and data. Cache memory is used to reduce the average time to access data from the Main memory.

Q 25 . What is spooling?

Ans : Spooling refers to simultaneous peripheral operations online, spooling refers to putting jobs in a buffer, a special area in memory, or on a disk where a device can access them when it is ready. Spooling is useful because devices access data at different rates.

Q 26. What is the functionality of an Assembler?

Ans : The Assembler is used to translate the program written in Assembly language into machine code. The source program is an input of an assembler that contains assembly language instructions. The output generated by the assembler is the object code or machine code understandable by the computer.

Q 27 . What are interrupts?

Ans : The interrupts are a signal emitted by hardware or software when a process or an event needs immediate attention.

It alerts the processor to a high-priority process requiring interruption of the current working process.

In I/O devices one of the bus control lines is dedicated to this purpose and is called the Interrupt Service Routine (ISR).



Q 28 . What is GUI?

Ans : GUI is short for Graphical User Interface. It provides users with an interface wherein actions can be performed by interacting with icons and graphical symbols.

Q 29 . What is preemptive multitasking?

Ans : Preemptive multitasking is a type of multitasking that allows computer programs to share operating systems (OS) and underlying hardware resources.

It divides the overall operating and computing time between processes, and the switching of resources between different processes occurs through predefined criteria.

Q 30 . What is a pipe and when is it used?

Ans : A Pipe is a technique used for inter-process communication.

A pipe is a mechanism by which the output of one process is directed into the input of another process.

Thus it provides a one-way flow of data between two related processes.

Q 31 . What are the advantages of semaphores?

Ans :

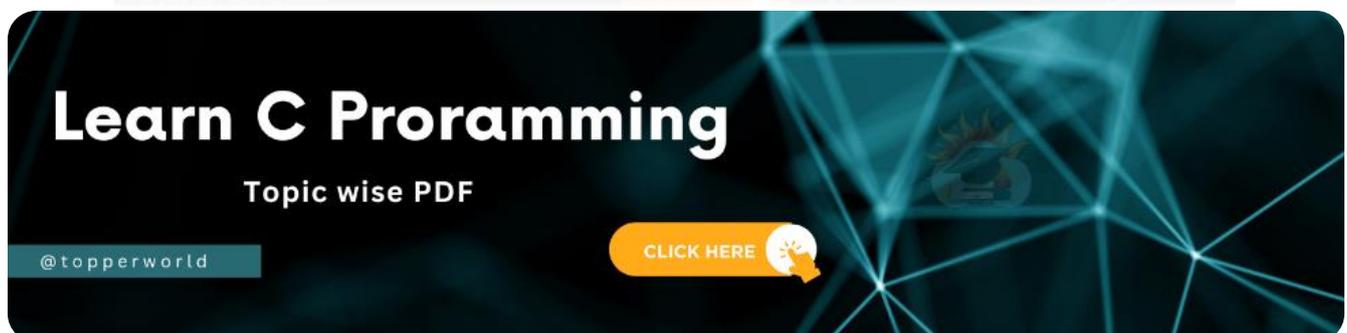
- They are machine-independent.
- Easy to implement.
- Correctness is easy to determine.
- Can have many different critical sections with different semaphores.
- Semaphores acquire many resources simultaneously.
- No waste of resources due to busy waiting.

Q 32 . What is a bootstrap program in the OS?

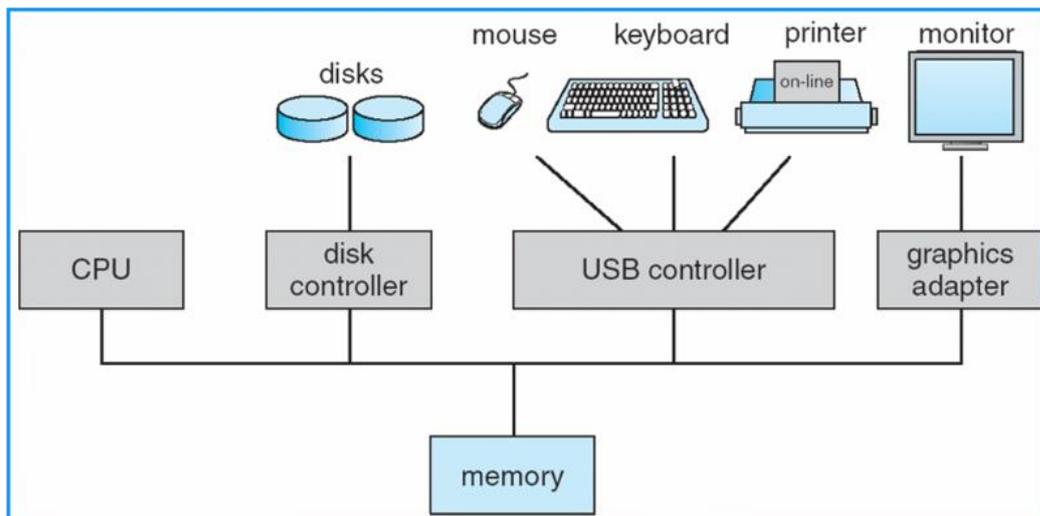
Ans : Bootstrapping is the process of loading a set of instructions when a computer is first turned on or booted. During the startup process, diagnostic tests are performed, such as the power-on self-test (POST), which set or checks configurations for devices and implements routine testing for the connection of peripherals, hardware, and external memory devices.



Learn C Programming
Topic wise PDF
@topperworld
CLICK HERE



Learn C Programming
Topic wise PDF
@topperworld
CLICK HERE



The bootloader or bootstrap program is then loaded to initialize the OS.

Q 33 . What is IPC?

Ans : Inter-process communication (IPC) is a mechanism that allows processes to communicate with each other and synchronize their actions.

The communication between these processes can be seen as a method of cooperation between them.

Q 34 . What are the different IPC mechanisms?

Ans : These are the methods in IPC:

- **Pipes (Same Process):** This allows a flow of data in one direction only. Analogous to simplex systems (Keyboard). Data from the output is usually buffered until the input process receives it which must have a common origin.
- **Named Pipes (Different Processes):** This is a pipe with a specific name it can be used in processes that don't have a shared common process origin. E.g. FIFO where the details written to a pipe are first named.
- **Message Queuing:** This allows messages to be passed between processes using either a single queue or several message queues. This is

managed by the system kernel these messages are coordinated using an API.

- **Semaphores:** This is used in solving problems associated with synchronization and avoiding race conditions. These are integer values that are greater than or equal to 0.
- **Shared Memory:** This allows the interchange of data through a defined area of memory. Semaphore values have to be obtained before data can get access to shared memory.
- **Sockets:** This method is mostly used to communicate over a network between a client and a server. It allows for a standard connection which is computer and OS independent

Q 35 .What is the difference between preemptive and non-preemptive scheduling?

Ans :

- In preemptive scheduling, the CPU is allocated to the processes for a limited time whereas, in Non-preemptive scheduling, the CPU is allocated to the process till it terminates or switches to waiting for the state.
- The executing process in preemptive scheduling is interrupted in the middle of execution when a higher priority one comes whereas, the executing process in non-preemptive scheduling is not interrupted in the middle of execution and waits till its execution.
- In Preemptive Scheduling, there is the overhead of switching the process from the ready state to the running state, vice-verse, and maintaining the ready queue. Whereas the case of non-preemptive scheduling has no overhead of switching the process from running state to ready state.

- In preemptive scheduling, if a high-priority process frequently arrives in the ready queue then the process with low priority has to wait for a long, and it may have to starve. On the other hand, in non-preemptive scheduling, if CPU is allocated to the process having a larger burst time then the processes with a small burst time may have to starve.
- Preemptive scheduling attains flexibility by allowing the critical processes to access the CPU as they arrive in the ready queue, no matter what process is executing currently. Non-preemptive scheduling is called rigid as even if a critical process enters the ready queue the process running CPU is not disturbed.
- Preemptive Scheduling has to maintain the integrity of shared data that 's why it is cost associative which is not the case with Non-preemptive Scheduling.

Q 36 . What is the zombie process?

Ans : A process that has finished the execution but still has an entry in the process table to report to its parent process is known as a zombie process. A child process always first becomes a zombie before being removed from the process table. The parent process reads the exit status of the child process which reaps off the child process entry from the process table.

Q 37 . What are orphan processes?

Ans : A process whose parent process no more exists i.e. either finished or terminated without waiting for its child process to terminate is called an orphan process.

Q 38 . What are starvation and aging in OS?

Ans :

Starvation: Starvation is a resource management problem where a process does not get the resources it needs for a long time because the resources are being allocated to other processes.

Aging: Aging is a technique to avoid starvation in a scheduling system. It works by adding an aging factor to the priority of each request. The aging factor must increase the priority of the request as time passes and must ensure that a request will eventually be the highest priority request.



Q 39. Write about monolithic kernel?

Ans : Apart from microkernel, Monolithic Kernel is another classification of Kernel. Like microkernel, this one also manages system resources between application and hardware, but user services and kernel services are implemented under the same address space.

It increases the size of the kernel, thus increasing the size of an operating system as well.

This kernel provides CPU scheduling, memory management, file management, and other operating system functions through system calls.

As both services are implemented under the same address space, this makes operating system execution faster.

Q 40 . What is Context Switching?

Ans : Switching of CPU to another process means saving the state of the old process and loading the saved state for the new process.

In Context Switching the process is stored in the Process Control Block to serve the new process so that the old process can be resumed from the same part it was left.



Q 41 . What is the difference between the Operating system and kernel?

Ans :

Operating System	Kernel
Operating System is system software.	The kernel is system software that is part of the Microkerneloperating system.
Operating System provides an interface b/w the user and the hardware.	The kernel provides an interface b/w the application and hardware.
It also provides protection and security.	Its main purpose is memory management, disk management, process management and task management.
All system needs a real-time operating real-time, and Microkernel system to run.	All operating system needs a kernel to run.
Type of operating system includes single and multiuser OS, multiprocessor OS, real-time OS, Distributed OS.	Type of kernel includes Monolithic and Microkernel.
It is the first program to load when the computer boots up.	It is the first program to load when the operating system loads

Q 42 . What is the difference between process and thread?

Ans :

S.NO	Process	Thread
1.	Process means any program is in execution.	Thread means a segment of a process.
2.	The process is less efficient in terms of communication.	Thread is more efficient in terms of communication.
3.	The process is isolated.	Threads share memory.
4.	The process is called heavyweight the process.	Thread is called lightweight process.
5.	Process switching uses, another process interface in operating system.	Thread switching does not require to call an operating system and cause an interrupt to the kernel.
6.	If one process is blocked then it will not affect the execution of other process	The second, thread in the same task could not run, while one server thread is blocked.

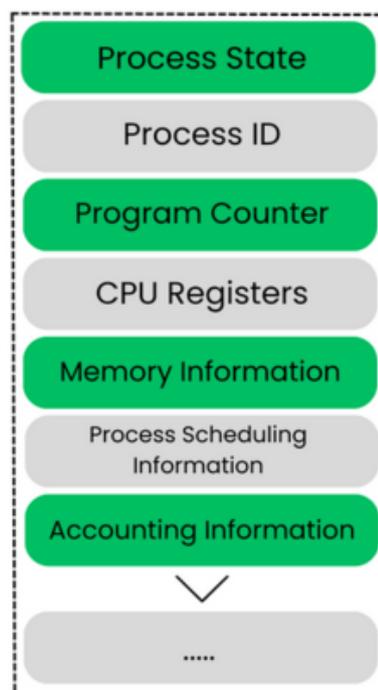
S.NO	Process	Thread
7.	The process has its own Process Control Block, Stack and Address Space.	Thread has Parents' PCB, its own Thread Control Block and Stack and common Address space.

Q 43 . What is PCB?

Ans : The process control block (PCB) is a block that is used to track the process' s execution status.

A process control block (PCB) contains information about the process, i.e. registers, quantum, priority, etc.

The process table is an array of PCBs, that means logically contains a PCB for all of the current processes in the system.



Q 44 . When is a system in a safe state?

Ans : The set of dispatchable processes is in a safe state if there exists at least one temporal order in which all processes can be run to completion without resulting in a deadlock.

Q 45 . What is Cycle Stealing?

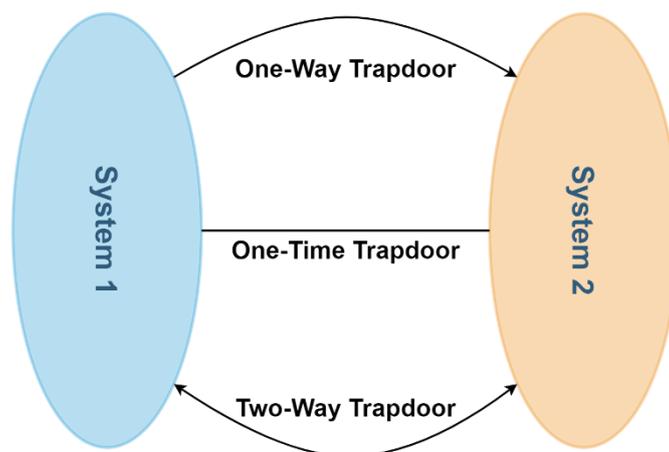
Ans : Cycle stealing is a method of accessing computer memory (RAM) or bus without interfering with the CPU.

It is similar to direct memory access (DMA) for allowing I/O controllers to read or write RAM without CPU intervention.

Q 46 . What are a Trap and Trapdoor?

Ans : A trap is a software interrupt, usually the result of an error condition, and is also a non-maskable interrupt and has the highest priority.

Trapdoor is a secret undocumented entry point into a program used to grant access without normal methods of access authentication.



Q 47 . Write a difference between process and program?

Ans :

S.NO	Program	Process
1.	Program contains a set of instructions designed to complete a specific task.	Process is an instance of an executing program.
2.	Program is a passive entity as it resides in the secondary memory.	Process is an active entity as it is created during execution and loaded into the main memory.
3.	The program exists in a single place and continues to exist until it is deleted.	Process exists for a limited span of time as it gets terminated after the completion of a task.
4.	A program is a static entity.	The process is a dynamic entity.

LEARN C++
PROGRAMMING
Topic Wise PDF
www.topperworld.in
CLICK HERE

The advertisement features a dark blue background with glowing blue lines and dots. The text 'LEARN C++' is in orange and white, 'PROGRAMMING' is in white, and 'Topic Wise PDF' is in green. A hand icon points to a 'CLICK HERE' button. The website URL 'www.topperworld.in' is in the bottom left corner.

S.NO	Program	Process
5.	Program does not have any resource requirement, it only requires memory space for storing the instructions.	Process has a high resource requirement, it needs resources like CPU, memory address, and I/O during its lifetime.
6.	The program does not have any control block.	The process has its own control block called Process Control Block.

Q 48 . What is a dispatcher?

Ans :

The dispatcher is the module that gives process control over the CPU after it has been selected by the short-term scheduler. This function involves the following:

- Switching context
- Switching to user mode
- Jumping to the proper location in the user program to restart that program



Q 49 . Write a difference between a user-level thread and a kernel-level thread?

Ans :

User-level thread	Kernel level thread
User threads are implemented by users.	kernel threads are implemented by OS.
OS doesn' t recognize user-level threads.	Kernel threads are recognized by OS.
Implementation of User threads is easy.	Implementation of the perform kernel thread is complicated.
Context switch time is less.	Context switch time is more.
Context switch requires no hardware support.	Hardware support is needed.
If one user-level thread performs a blocking operation then entire process will be blocked.	If one kernel thread perform a the blocking operation then another thread can continue execution.
User-level threads are designed as dependent threads.	Kernel level threads are designed as independent threads.

Q 50 . Difference between Multithreading and Multitasking?

Ans :

S.No	Multi-threading	Multi-tasking
1.	Multiple threads are executing at the same time at the same or different part of the program.	Several programs are executed concurrently.
2.	CPU switches between multiple threads.	CPU switches between multiple tasks and processes.
3.	It is the process of a lightweight part.	It is a heavyweight process.
4.	It is a feature of the process.	It is a feature of the OS.
5.	Multi-threading is sharing of computing resources among threads of a single process.	Multitasking is sharing of computing resources(CPU, memory, devices, etc.) among processes.



ABOUT US

At TopperWorld, we are on a mission to empower college students with the knowledge, tools, and resources they need to succeed in their academic journey and beyond.

➤ **Our Vision**

- ❖ **Our vision is to create a world where every college student can easily access high-quality educational content, connect with peers, and achieve their academic goals.**
- ❖ **We believe that education should be accessible, affordable, and engaging, and that's exactly what we strive to offer through our platform.**

➤ **Unleash Your Potential**

- ❖ **In an ever-evolving world, the pursuit of knowledge is essential. TopperWorld serves as your virtual campus, where you can explore a diverse array of online resources tailored to your specific college curriculum.**
- ❖ **Whether you're studying science, arts, engineering, or any other discipline, we've got you covered.**
- ❖ **Our platform hosts a vast library of e-books, quizzes, and interactive study tools to ensure you have the best resources at your fingertips.**

➤ **The TopperWorld Community**

- ❖ Education is not just about textbooks and lectures; it's also about forming connections and growing together.
- ❖ TopperWorld encourages you to engage with your fellow students, ask questions, and share your knowledge.
- ❖ We believe that collaborative learning is the key to academic success.

➤ Start Your Journey with TopperWorld

- ❖ Your journey to becoming a top-performing college student begins with TopperWorld.
- ❖ Join us today and experience a world of endless learning possibilities.
- ❖ Together, we'll help you reach your full academic potential and pave the way for a brighter future.
- ❖ Join us on this exciting journey, and let's make academic success a reality for every college student.

“UNLOCK YOUR POTENTIAL”

With- **TOPPERWORLD**

Explore More



www.topperworld.in

DSA TUTORIAL

C TUTORIAL

C++ TUTORIAL

JAVA TUTORIAL

PYTHON TUTORIAL

Follow Us On



E-mail



topperworld.in@gmail.com

