Operation Software Programing Q and A

---

1. What is a process and process table?

A process is an instance of a program in execution. For example, a Web Browser is a process, and a shell (or command prompt) is a process. The operating system is responsible for managing all the processes that are running on a computer and allocates each process a certain amount of time to use the processor. In addition, the operating system also allocates various other resources that processes will need, such as computer memory or disks. To keep track of the state of all the processes, the operating system maintains a table known as the process table. Inside this table, every process is listed along with the resources the process is using and the current state of the process.

2. What are the different states of the process?

Processes can be in one of three states: running, ready, or waiting. The running state means that the process has all the resources it needs for execution and it has been given permission by the operating system to use the processor. Only one process can be in the running state at any given time. The remaining processes are either in a waiting state (i.e., waiting for some external event to occur such as user input or disk access) or a ready state (i.e., waiting for permission to use the processor). In a real operating system, the waiting and ready states are implemented as queues that hold the processes in these states.

Cleancode

3. What is a Thread?

A thread is a single sequence stream within a process. Because threads have some of the properties of processes, they are sometimes called lightweight processes. Threads are a popular way to improve the application through parallelism. For example, in a browser, multiple tabs can be different threads. MS Word uses multiple threads, one thread to format the text, another thread to process inputs, etc.

4. What are the differences between process and thread?

A thread has its own program counter (PC), a register set, and a stack space. Threads are not independent of one another, like processes. As a result, threads share with other threads their code section, data section, and OS resources like open files and signals.

5. What are the benefits of multithreaded programming?

- ➢ It makes the system more responsive and enables resource sharing.

- It leads to the use of multiprocess architecture.
- It is more economical and preferred.

6. What is Thrashing?

Thrashing is a situation when the performance of a computer degrades or collapses. Thrashing occurs when a system spends more time processing page faults than executing transactions. While processing page faults is necessary in order to appreciate the benefits of virtual memory, thrashing has a negative effect on the system. As the page fault rate increases, more transactions need processing from the paging device. The queue at the paging device increases, resulting in increased service time for a page fault.

7. What is Buffer?

A buffer is a memory area that stores data being transferred between two devices or between a device and an application.

8. What is virtual memory?

Virtual memory creates an illusion that each user has one or more contiguous address spaces, each beginning at address zero. The sizes of such virtual address spaces are generally very high. The idea of virtual memory is to use disk space to extend the RAM. Running processes don't need to care whether the memory is from RAM or disk. The illusion of such a large amount of memory is created by subdividing the virtual memory into smaller pieces, which can be loaded into physical memory whenever they are needed by a process.

9. Explain the main purpose of an operating system?

- 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.

10. What is demand paging?

The process of loading the page into memory on demand (whenever a page fault occurs) is known as demand paging.

11. What is a kernel?

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.

12. What are the different scheduling algorithms?

- First-Come, First-Served (FCFS) Scheduling.
- Shortest-Job-Next (SJN) Scheduling.
- Priority Scheduling.
- Shortest Remaining Time.
- Round Robin(RR) Scheduling.
- Multiple-Level Queues Scheduling.

Cleancode

13. Describe the objective of multi-programming.

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.

14. What is the time-sharing system?

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.

15. What problem we face in computer system without OS?

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

16. Give some benefits of multithreaded programming?

A thread is also known as a lightweight process. The idea is to achieve parallelism by dividing a process into multiple threads. Threads within the same process run in shared memory space,

17. Briefly explain FCFS.

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.

18. What is the RR scheduling algorithm?

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

Cleancode

19. Enumerate the different RAID levels?

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

20. What is Banker's algorithm?

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.

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

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.

22. What are overlays?

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".

23. What is fragmentation?

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.

Cleancode

24. What is the basic function of paging?

- 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

25. How does swapping result in better memory management?

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

26. Write a name of classic synchronization problems?

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

27. What is the Direct Access Method?

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).

Cleancode

28. When does thrashing occur?

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

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

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.

30. What is multitasking?

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.

31. What is caching?

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.

32. What is spooling?

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.

33. What is the functionality of an Assembler?

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.

Cleancode

34. What are interrupts?

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).

35. What is GUI?

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.

36. What is preemptive multitasking?

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.

37. What is a pipe and when is it used?

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.

38. What are the advantages of semaphores?

- 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.

39. What is a bootstrap program in the OS?

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

Cleancode

40. What is IPC?

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.

41. What are the different IPC mechanisms?

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.

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

Aspect	Preemptive Scheduling	Non-preemptive Scheduling
CPU Allocation	CPU is allocated for a limited time to processes.	CPU is allocated to a process until it terminates or waits.
nterruption during execution	Executing process is interrupted for higher priority tasks.	Executing process is not interrupted; waits for completion.
Overhead	Overhead in switching and maintaining ready queue.	No overhead in switching from running to ready state.
Starvation Risk	Low-priority processes may starve due to high-priority ones.	Processes with large burst time may cause starvation.
Flexibility	Critical processes access CPU as they arrive in ready queue	Running process is not disturbed even for critical tasks.
Data Integrity and Cost Association	Maintains data integrity, associated with higher cost.	Data integrity concerns are less, lower cost association.

43. What is the zombie process?

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.

44. What are orphan processes?

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.

45. What are starvation and aging in OS?

- **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

46. Write about monolithic kernel?

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.

47. What is Context Switching?

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.

48. What is the difference between the Operating system and kernel?

Aspect	Operating System	Kernel
System Software	Operating System is system software.	The kernel is system software that is part of the operating system.
Interface	Provides an interface	Provides an interface

	between the user and hardware.	between the application and hardware.
Functionality	Provides protection, security, and various services.	Mainly responsible for memory, disk, process, and task management.
Necessity	All systems need an operating system to run.	All operating systems need a kernel to run.
Types	Single and multiuser OS, multiprocessor OS, real-time OS, Distributed OS.	Monolithic and Microkernel are types of kernels.
Boot Process	Loads after the computer boots up.	Loads first when the operating system boots up.

49. What is the difference between process and thread?

S.No	Process	Thread
1	Process means any program is in execution.	Thread means a segment of a process.
2	Less efficient in terms of communication.	More efficient in terms of communication.
3	Isolated. Threads share memory.	Threads share memory within the same process.
4	Called heavyweight.	Called lightweight processes.
5	Called lightweight processes.	Thread switching doesn't require OS intervention.
6	Blocking one process doesn't affect other processes.	Blocking one thread can affect other threads in the same task.
7	Own Process Control Block, Stack, and Address Space.	Thread shares the PCB with its parent, has its own TCB, Stack, and common Address space within the process.

Cleancode

50. What is PCB?

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.

51. When is a system in a safe state?

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.

52. What is Cycle Stealing?

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.

53. What are a Trap and Trapdoor?

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.

Cleancode

54. Write a difference between process and program?

S.No	Program	Process
1	Contains a set of instructions for a specific task.	An instance of an executing program.
2	Passive entity residing in secondary memory.	Active entity created during execution and loaded into main memory.
3	Exists in a single place until deleted.	Exists for a limited span and terminates after completing a task.
4	Static entity.	Dynamic entity.
5	Doesn't have resource requirements except memory for storage.	Requires CPU, memory, I/O, and other resources during execution.
6	Does not have a control block.	Has its own control block known as Process Control Block (PCB).

55. What is a dispatcher?

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

56. Define the term dispatch latency?

Dispatch latency can be described as the amount of time it takes for a system to respond to a request for a process to begin operation. With a scheduler written specifically to honor application priorities, real-time applications can be developed with a bounded dispatch latency.

57. What are the goals of CPU scheduling?

- ➢ Max CPU utilization [Keep CPU as busy as possible]Fair allocation of CPU.
- ➢ Max throughput [Number of processes that complete their execution per time unit]
- ➢ Min turnaround time [Time taken by a process to finish execution]
- ➢ Min waiting time [Time a process waits in ready queue]
- ➢ Min response time [Time when a process produces the first response]

Cleancode

58. What is a critical- section?

When more than one processes access the same code segment that segment is known as the critical section. The critical section contains shared variables or resources which are needed to be synchronized to maintain the consistency of data variables. In simple terms, a critical section is a group of instructions/statements or regions of code that need to be executed atomically such as accessing a resource (file, input or output port, global data, etc.).

59. Write the name of synchronization techniques?

- ➢ Mutexes
- ➢ Condition variables
- ➢ Semaphores
- ➢ File locks
- ➢ Intermediate OS Interview Questions.

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

User-Level Threads	Kernel-Level Threads
Implemented by Users: User-level threads are created and managed by user-level libraries or application code without direct kernel support.	Implemented by OS: Kernel-level threads are managed and supported directly by the operating system.
Not Recognized by OS: The operating system is unaware of user-level threads and manages them as part of a single process.	Recognized by OS: The OS is aware of kernel-level threads and schedules them independently.
Easy Implementation: Since they're managed at the user level, their implementation is simpler and more flexible.	Complex Implementation: Kernel-level threads are more complex to implement due to their direct management by the operating system.
Low Context Switch Time: Context switches between user-level threads within the same process are generally faster because they don't involve kernel intervention.	Higher Context Switch Time: Context switches between kernel-level threads involve the kernel and are generally slower compared to user-level threads.
No Hardware Support: User-level threads can be implemented without requiring specific hardware support.	Hardware Support Needed: Kernel-level threads often rely on specific hardware support for their operation.
	Blocking Operations: If one kernel-level thread in a process performs a blocking operation, other threads within the same process can continue execution without being affected.
	Independence: Kernel-level threads are designed to be more independent, allowing them to execute concurrently on multiple processors or cores.

Cleancode

61. Write down the advantages of multithreading?

Some of the most important benefits of MT are:

- ➢ Improved throughput. Many concurrent compute operations and I/O requests within a single process.
- ➢ Simultaneous and fully symmetric use of multiple processors for computation and I/O.
- ➢ Superior application responsiveness. If a request can be launched on its own thread, applications do not freeze or show the “hourglass”. An entire application will not block or otherwise wait, pending the completion of another request.
- ➢ Improved server responsiveness. Large or complex requests or slow clients don’t block other requests for service. The overall throughput of the server is much greater.

- Minimized system resource usage. Threads impose minimal impact on system resources. Threads require less overhead to create, maintain, and manage than a traditional process.
- Program structure simplification. Threads can be used to simplify the structure of complex applications, such as server-class and multimedia applications. Simple routines can be written for each activity, making complex programs easier to design and code, and more adaptive to a wide variation in user demands.
- Better communication. Thread synchronization functions can be used to provide enhanced process-to-process communication. In addition, sharing large amounts of data through separate threads of execution within the same address space provides extremely high-bandwidth, low-latency communication between separate tasks within an application.
-

62. Difference between Multithreading and Multitasking?

S.No	**Multithreading**	**Multitasking**
1	Multiple threads are executing at the same time at the same or different parts 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	Multithreading is the sharing of computing resources among threads of a single process.	Multitasking is the sharing of computing resources (CPU, memory, devices, etc.) among processes.

Cleancode

63. What are the drawbacks of semaphores?

- Priority Inversion is a big limitation of semaphores.
- Their use is not enforced but is by convention only.
- The programmer has to keep track of all calls to wait and signal the semaphore.
- With improper use, a process may block indefinitely. Such a situation is called Deadlock.

64. What is Peterson's approach?

It is a concurrent programming algorithm. It is used to synchronize two processes that maintain the mutual exclusion for the shared resource. It uses two variables, a bool array flag of size 2 and an int variable turn to accomplish it.

65. Define the term Bounded waiting?

A system is said to follow bounded waiting conditions if a process wants to enter into a critical section will enter in some finite time.

66. What are the solutions to the critical section problem?

There are three solutions to the critical section problem:

- Software solutions
- Hardware solutions
- Semaphores

67. What is a Banker's algorithm?

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.

cleancode

68. What is concurrency?

A state in which a process exists simultaneously with another process than those it is said to be concurrent.

69. Write a drawback of concurrency?

- It is required to protect multiple applications from one another.
- It is required to coordinate multiple applications through additional mechanisms.
- Additional performance overheads and complexities in operating systems are required for switching among applications.
- Sometimes running too many applications concurrently leads to severely degraded performance.

70. What are the necessary conditions which can lead to a deadlock in a system?

- **Mutual Exclusion:** There is a resource that cannot be shared.

- **Hold and Wait:** A process is holding at least one resource and waiting for another resource, which is with some other process.
- **No Preemption:** The operating system is not allowed to take a resource back from a process until the process gives it back.
- **Circular Wait:** A set of processes waiting for each other in circular form.

71. What are the issues related to concurrency?

- **Non-atomic:** Operations that are non-atomic but interruptible by multiple processes can cause problems.
- **Race conditions:** A race condition occurs of the outcome depends on which of several processes gets to a point first.
- **Blocking:** Processes can block waiting for resources. A process could be blocked for a long period of time waiting for input from a terminal. If the process is required to periodically update some data, this would be very undesirable.
- **Starvation:** It occurs when a process does not obtain service to progress.
- **Deadlock:** It occurs when two processes are blocked and hence neither can proceed to execute

Cleancode

72. Why do we use precedence graphs?

- A precedence graph is a directed acyclic graph that is used to show the execution level of several processes in the operating system. It has the following properties also:
- Nodes of graphs correspond to individual statements of program code.
- An edge between two nodes represents the execution order.
- A directed edge from node A to node B shows that statement A executes first and then Statement B executes

73. Explain the resource allocation graph?

The resource allocation graph is explained to us what is the state of the system in terms of processes and resources. One of the advantages of having a diagram is, sometimes it is possible to see a deadlock directly by using RAG.

74. What is a deadlock?

Deadlock is a situation when two or more processes wait for each other to finish and none of them ever finish. Consider an example when two trains are coming toward each other on the same track and there is only one track, none of the trains can move once they are in front of each other. A similar situation occurs in operating systems when there are two or more processes that hold some resources and wait for resources held by other(s).

75. What is the goal and functionality of memory management?

The goal and functionality of memory management are as follows;

- ➢ Relocation
- ➢ Protection
- ➢ Sharing
- ➢ Logical organization
- ➢ Physical organization

76. Write a difference between physical address and logical address?

S.No.	Parameters	Logical Address	Physical Address
1	Basic	Virtual address generated by CPU.	Location in memory.
2	Address	Set of logical addresses generated by CPU in reference to a program is Logical Address Space.	Set of physical addresses mapped to corresponding logical addresses is Physical Address.
3	Visibility	User can view the logical address of a program.	User cannot view the physical address of the program.
4	Access	User uses logical address to access physical address.	User cannot directly access the physical address.
5	Generation	Logical Address is generated by the CPU.	Physical Address is computed by Memory Management Unit (MMU).

76. Explain address binding?

The Association of program instruction and data to the actual physical memory locations is called Address Binding.

77. Write different types of address binding?

Address Binding is divided into three types as follows.

- ➢ Compile-time Address Binding
- ➢ Load time Address Binding
- ➢ Execution time Address Binding

78. Write an advantage of dynamic allocation algorithms?

- ➢ When we do not know how much amount of memory would be needed for the program beforehand.
- ➢ When we want data structures without any upper limit of memory space.
- ➢ When you want to use your memory space more efficiently.
- ➢ Dynamically created lists insertions and deletions can be done very easily just by the manipulation of addresses whereas in the case of statically allocated memory insertions and deletions lead to more movements and wastage of memory.
- ➢ When you want to use the concept of structures and linked lists in programming, dynamic memory allocation is a must.

79. Write a difference between internal fragmentation and external fragmentation?

S.No	Internal Fragmentation	External Fragmentation
1	In fixed-sized memory blocks, assigned to processes.	Variable-sized memory blocks assigned to processes.
2	Occurs when process is larger than memory.	Occurs when memory gaps exist between allocated blocks.
3	Best-fit block is a solution for internal fragmentation.	Solutions include compaction, paging, and segmentation.
4	Arises from fixed-sized memory partitions.	Arises from variable-sized memory partitions based on process sizes.
5	Difference between allocated and required memory space.	Unused spaces between non-contiguous memory fragments.

Cleancode

80. Define the Compaction?

The process of collecting fragments of available memory space into contiguous blocks by moving programs and data in a computer's memory or disk.

81. Write about the advantages and disadvantages of a hashed-page table?

Advantages

The main advantage is synchronization.

- ➢ In many situations, hash tables turn out to be more efficient than search trees or any other table lookup structure. For this reason, they are widely used in many kinds of computer software, particularly for associative arrays, database indexing, caches, and sets.

Disadvantages

- ➢ Hash collisions are practically unavoidable. when hashing a random subset of a large set of possible keys.
- ➢ Hash tables become quite inefficient when there are many collisions.
- ➢ Hash table does not allow null values, like a hash map.
- ➢ Define Compaction.

82. Write a difference between paging and segmentation

S.No.	Paging	Segmentation
1	Program divided into fixed-size pages.	Program divided into variable-size sections.
2	Operating system manages paging.	Compiler manages segmentation.
3	Page size determined by hardware.	Section size specified by the user.
4	Generally faster than segmentation.	Can be slower compared to paging.
5	Could result in internal fragmentation.	Could result in external fragmentation.
6	Logical address split into page number and offset.	Logical address split into segment number and offset.
7	Involves a page table with base addresses for pages.	Involves a segment table with segment number and offset.
8	Page table maintains page data.	Segment table maintains segment data.
9	OS maintains a free frame list.	OS maintains a list of holes in memory.
10	Paging mechanism is invisible to the user.	Segmentation is visible to the user.
11	Processor needs page number and offset for address calc.	Processor uses segment number and offset for address calc.

83. Write a definition of Associative Memory and Cache Memory?

S.No.	Associative Memory	Cache Memory
1	Memory unit accessed by content.	Fast and small memory.
2	Reduces time to find stored items.	Reduces average memory access time.

3	Data accessed by content.	Data accessed by address.
4	Used when search time is very short.	Used when a particular group of data is accessed repeatedly.
5	Basic characteristic is logic circuit for matching content.	Basic characteristic is fast access.

84. What is “Locality of reference”?

The locality of reference refers to a phenomenon in which a computer program tends to access the same set of memory locations for a particular time period. In other words, Locality of Reference refers to the tendency of the computer program to access instructions whose addresses are near one another.

85. Write down the advantages of virtual memory?

- A higher degree of multiprogramming.
- Allocating memory is easy and cheap
- Eliminates external fragmentation
- Data (page frames) can be scattered all over the PM
- Pages are mapped appropriately anyway
- Large programs can be written, as the virtual space available is huge compared to physical memory.
- Less I/O required leads to faster and easy swapping of processes.
- More physical memory is available, as programs are stored on virtual memory, so they occupy very less space on actual physical memory.
- More efficient swapping.

Cleancode

86. How to calculate performance in virtual memory?

The performance of virtual memory of a virtual memory management system depends on the total number of page faults, which depend on “paging policies” and “frame allocation”.

Effective access time = (1-p) x Memory access time + p x page fault time

87. Write down the basic concept of the file system?

A file is a collection of related information that is recorded on secondary storage. Or file is a collection of logically related entities. From the user’s perspective, a file is the smallest allotment of logical secondary storage.

88. Write the names of different operations on file?

Operation on file:

- ➢ Create
- ➢ Open
- ➢ Read
- ➢ Write
- ➢ Rename
- ➢ Delete
- ➢ Append
- ➢ Truncate
- ➢ Close

89. Define the term Bit-Vector?

A Bitmap or Bit Vector is a series or collection of bits where each bit corresponds to a disk block. The bit can take two values: 0 and 1: 0 indicates that the block is allocated and 1 indicates a free block.

90. What is a File allocation table?

FAT stands for File Allocation Table and this is called so because it allocates different files and folders using tables. This was originally designed to handle small file systems and disks. A file allocation table (FAT) is a table that an operating system maintains on a hard disk that provides a map of the cluster (the basic units of logical storage on a hard disk) that a file has been stored in.

91. What is rotational latency?

Rotational Latency: Rotational Latency is the time taken by the desired sector of the disgeek to rotate into a position so that it can access the read/write heads. So the disk scheduling algorithm that gives minimum rotational latency is better.

92. What is seek time?

Seek Time: Seek time is the time taken to locate the disk arm to a specified track where the data is to be read or written. So the disk scheduling algorithm that gives a minimum average seek time is better.

Advanced OS Interview Questions

93. What is Belady's Anomaly?

Bélády's anomaly is an anomaly with some page replacement policies increasing the number of page frames resulting in an increase in the number of page faults. It occurs when the First In First Out page replacement is used.

94. What happens if a non-recursive mutex is locked more than once?

Deadlock. If a thread that had already locked a mutex, tries to lock the mutex again, it will enter into the waiting list of that mutex, which results in a deadlock. It is because no other thread can unlock the mutex. An operating system implementer can exercise care in identifying the owner of the mutex and return it if it is already locked by the same thread to prevent deadlocks.

Cleancode

95. What are the advantages of a multiprocessor system?

There are some main advantages of a multiprocessor system:

- Enhanced performance.
- Multiple applications.
- Multi-tasking inside an application.
- High throughput and responsiveness.
- Hardware sharing among CPUs.

96. What are real-time systems?

A real-time system means that the system is subjected to real-time, i.e., the response should be guaranteed within a specified timing constraint or the system should meet the specified deadline.

97. How to recover from a deadlock?

We can recover from a deadlock by following methods:

- Process termination
- Abort all the deadlock processes
- Abort one process at a time until the deadlock is eliminated

- ➢ Resource preemption
- ➢ Rollback
- ➢ Selecting a victim

98. What factors determine whether a detection algorithm must be utilized in a deadlock avoidance system?

One is that it depends on how often a deadlock is likely to occur under the implementation of this algorithm. The other has to do with how many processes will be affected by deadlock when this algorithm is applied.

99. Explain the main purpose of an operating system?

Operating systems exist for two main purposes. One is that it is designed to make sure a computer system performs well by managing its computational activities. Another is that it provides an environment for the development and execution of programs.

100. What is demand paging?

Demand paging is referred when not all of a process's pages are in the RAM, then the OS brings the missing(and required) pages from the disk into the RAM.

Cleancode

101. What are the advantages of a multiprocessor system?

With an increased number of processors, there is a considerable increase in throughput. It can also save more money because they can share resources. Finally, overall reliability is increased as well.

102. What are real-time systems?

Real-time systems are used when rigid time requirements have been placed on the operation of a processor. It has well defined and fixed time constraints.

103. What is a virtual memory?

Virtual memory is a memory management technique for letting processes execute outside of memory. This is very useful especially is an executing program cannot fit in the physical memory.

Operating System (OS) Interview Questions

104. Describe the objective of multiprogramming.

The main objective of multiprogramming is to have a process running at all times. With this design, CPU utilization is said to be maximized.

105. What is time- sharing system?

In a Time-sharing system, the CPU executes multiple jobs by switching among them, also known as multitasking. This process happens so fast that users can interact with each program while it is running.

106. What is SMP?

SMP is a short form of Symmetric Multi-Processing. It is the most common type of multiple-processor systems. In this system, each processor runs an identical copy of the operating system, and these copies communicate with one another as needed.

Cleancode

107. How are server systems classified?

Server systems can be classified as either computer-server systems or file server systems. In the first case, an interface is made available for clients to send requests to perform an action. In the second case, provisions are available for clients to create, access and update files.

108. What is asymmetric clustering?

In asymmetric clustering, a machine is in a state known as hot standby mode where it does nothing but to monitor the active server. That machine takes the active server's role should the server fails.

109. What is a thread?

A thread is a basic unit of CPU utilization. In general, a thread is composed of a thread ID, program counter, register set, and the stack.

110. Give some benefits of multithreaded programming.

- there is increased responsiveness to the user
- resource sharing within the process
- economy
- utilization of multiprocessing architecture

111.Briefly explain FCFS.

FCFS stands for First-come, first-served. It is one type of scheduling algorithm. In this scheme, the process that requests the CPU first is allocated the CPU first. Implementation is managed by a FIFO queue.

112. What is RR scheduling algorithm?

RR (round-robin) scheduling algorithm is primarily aimed for time-sharing systems. A circular queue is a setup in such a way that the CPU scheduler goes around that queue, allocating CPU to each process for a time interval of up to around 10 to 100 milliseconds.

113. What are necessary conditions which can lead to a deadlock situation in a system?

Deadlock situations occur when four conditions occur simultaneously in a system: Mutual exclusion; Hold and Wait; No preemption; and Circular wait.

Cleancode

114. Enumerate the different RAID levels.

- RAID 0 – Non-redundant striping
- RAID 1 – Mirrored Disks
- RAID 2 – Memory-style error-correcting codes
- RAID 3 – Bit-interleaved Parity
- RAID 4 – Block-interleaved Parity
- RAID 5 – Block-interleaved distributed Parity
- RAID 6 – P+Q Redundancy

115. Describe Banker's algorithm

Bankers_Algorithm

Bankers Algorithm

Banker's algorithm is one form of deadlock-avoidance in a system. It gets its name from a banking system wherein the bank never allocates available cash in such a way that it can no longer satisfy the needs of all of its customers.

116. What factors determine whether a detection-algorithm must be utilized in a deadlock avoidance system?

One is that it depends on how often a deadlock is likely to occur under the implementation of this algorithm. The other has to do with how many processes will be affected by deadlock when this algorithm is applied.

117. State the main difference between logical from physical address space.

Logical address refers to the address that is generated by the CPU. On the other hand, physical address refers to the address that is seen by the memory unit.

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

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.

119. What are overlays?

Overlays are used to enable a process to be larger than the amount of memory allocated to it. The basic idea of this is that only instructions and data that are needed at any given time are kept in memory.

Cleancode

120. What is the basic function of paging?

Paging is a memory management scheme that permits the physical address space of a process to be noncontiguous. It avoids the considerable problem of having to fit varied sized memory chunks onto the backing store.

121. What is fragmentation?

Fragmentation is memory wasted. It can be internal if we are dealing with systems that have fixed-sized allocation units, or external if we are dealing with systems that have variable-sized allocation units.

122. How does swapping result in better memory management?

During regular intervals that are set by the operating system, processes can be copied from main memory to a backing store, and then copied back later. Swapping allows more operations to be run that can fit into memory at one time.

Cleancode