Top 100+ Operating System Interview Questions In 2024

Enhance your operating system interview questions with a comprehensive guide from fundamental to mastery-level questions.

f 🐦 in

General Interview Questions   ⌃

Software Testing Interview Questions

Manual Testing Interview Questions

Automation Testing Interview Questions

Agile Interview Questions

CI/CD Interview Questions

BDD Interview Questions

Mobile Testing Interview Questions

Top 70+ Scrum Master Interview Questions [2024]

Top 100+ Operating System Interview Questionsns [2024]

Top 40+ Quality Assurance (QA) Interview Questions

CI/CD Tools Interview Questions   ⌄

Testing Types Interview Questions   ⌄

Testing Framework Interview Questions   ⌄

Programming Languages Interview Questions   ⌄

Development Framework Interview Questions   ⌄

OVERVIEW

An operating system is a critical component of any computer system, essential for personal and enterprise computing environments. Operating system communication and operation between computer hardware and software eventually help in the easy running of the tasks and better use of resource management.

To become an operating system developer, individuals must pass interviews that typically cover a range of Operating System-related questions. These operating system interview questions are designed to prepare individuals for interviews by covering a spectrum of OS-related topics. From fundamental concepts to advanced principles, these operating system interview questions help candidates like you to understand OS basics and tackle complexities. Whether you are a beginner or an experienced IT professional, this resource aims to boost your confidence to excel in your next OS interviews.

> 💡 Operating System Interview Questions Sheet
>
> **Note :** We have compiled all the questions in a sheet Feel free to comment on it. Check it out now!!

Fundamental Level Operating System Interview Questions

In the below set of operating system interview questions, you will learn fundamental aspects of OS components, which are essential for understanding the basics of operating systems and their other elements, such as deadlock, process and process table, and more.

1. What Is an Operating System?

An operating system (OS) is essential software that combines your computer's hardware and software resources to ensure smooth integration. The OS facilitates smooth communication and operation of software applications by bridging users and computer hardware. The OS manages, handles, and coordinates a computer's overall activities and resource sharing. The computer would be a useless box as it serves as the system's foundation.

2. What Is Meant by Deadlock?

Deadlock occurs in a system when two or more processes cannot proceed because each is waiting for a resource held by another process, which is also waiting for a resource held by another process in the cycle. This situation leads to a deadlock, where no progress can be made by any processes involved.

For example, consider two trains on a single-track railway line, each waiting for the other to move before they can proceed. If neither train moves, they are deadlocked. Similarly, in operating systems, deadlock happens when multiple processes hold resources and wait for others to release the necessary resources, creating a circular

3. What Four Conditions Are Both Necessary and Sufficient for a Deadlock To Occur?

The following are the four conditions:

- **Mutual Exclusion Condition:** It states that the resources must not be shared.

- **Hold and Wait Condition:** This condition states that a process must wait for additional resources while still holding allocated resources.

- **Condition of Non-Preemption:** Resources cannot be removed while processes utilize them.

- **Circular Wait Condition:** It elucidates the second condition. It indicates that the system processes are arranged in a circular list or chain, with each process waiting for a resource held by the next process in the chain.

4. What Does the Time-Sharing System Include?

In a time-sharing system, the CPU rapidly switches between multiple jobs or processes to give the illusion of simultaneous execution, a method called multitasking. This rapid switching allows each user or program to have a share of the CPU's time, making it appear that multiple programs are running simultaneously.

5. Explain What Throughput Means

Throughput is the total number of processes that finish their execution successfully in a given period. It evaluates how well the system manages and finishes tasks throughout a period, showing its effectiveness.

6. What Is IPC? What Are the Different IPC Mechanisms?

IPC, or Interprocess Communication, involves utilizing shared resources such as memory between processes or threads. Through IPC, the operating system facilitates communication among different processes. Its primary function is to exchange data between multiple threads within one or more programs or processes under the supervision of the OS.

- **Different IPC Mechanisms:**

  - Pipes

  - Message Queuing

  - Semaphores

  - Sockets

  - Shared Memory

  - Signals

7. What Are the Different Types of OS?

There are mainly five different types of OS:

- Batched OS (e.g., Payroll System, Transactions Process, etc.)

- Multiprogrammed OS (e.g., Windows OS, UNIX OS, etc.)

- Time Sharing OS (e.g., Multics, etc.)

- Distributed OS (e.g., LOCUS, etc.)

- Real-Time OS (e.g., PSOS, VRTX, etc.)

8. Explain the Concept of Reentrancy

Reentrancy is a technique for multiprogrammed time sharing systems that saves memory. A reentrant procedure allows multiple users to share a single program copy simultaneously. It has two main aspects: the program code remains unmodifiable, and each user process maintains its separate local data set.

The permanent part consists of the code, while the temporary part includes pointers back to the calling program and its local variables. Each execution instance, known as an activation, executes the code in the permanent part with its copy of local variables and parameters. The activation record associated with each instance is typically stored on the stack.

9. What Is a Process and Process Table?

A process is an active program instance, like a web browser or command prompt. The operating system manages all running processes, allocating CPU time and resources such as memory and disk access. To monitor the status of all processes, the operating system maintains a structured repository called the process table. This table lists each active process, the resources it utilizes, and its current operational state.

10. How Are Server Systems Classified?

Server systems can be categorized into computer-server systems and file-server systems. In computer-server systems, clients interact with an interface to send requests for actions to be performed by the server. In file-server systems, clients have access to create, retrieve, update, and delete files stored on the server.

11. What Is a Thread?

A thread represents a single sequential flow of execution within a process. Threads, often called lightweight processes, share the same memory space, including code and data sections and operating system resources like open files and signals. However, each thread has its program counter (PC), register set, and stack space, allowing it to execute independently within the context of the parent process. Threads are commonly used to enhance application performance through parallelism; for instance, multiple tabs in a browser or different functional aspects of software like MS Word can each operate as separate threads.

12. What Are the Differences Between Process and Thread?

Threads, being lightweight processes, share resources within the same address space of the parent process, including code and data sections. Each thread has its program counter (PC), registers, and stack space, enabling independent execution within the process context. Unlike processes, threads are not fully independent entities and can communicate and synchronize more efficiently, making them suitable for concurrent and parallel execution in multithreaded environments.

13. What Kinds of Threads Exist?

There are two ways in which threads can be implemented:

- Threads controlled by the user.
- Threads handled at the kernel level.

14. What Is Included in the Multi-Processor System?

Multi-processor systems include two or more CPUs within a single system. This configuration helps the system to process multiple tasks simultaneously by improving performance and efficiency. Multi-processor systems have a shared memory structure,

enabling all CPUs to access the same memory space and allowing efficient communication and data sharing between processors.

15. What Advantages Does a Multiprocessor System Offer?

Some of the advantages of a multiprocessor system are:

- These systems are frequently utilized to enhance performance in concurrently running multiple programs.

- Adding more processors means more tasks can be finished within the same time frame.

- One can also experience a significant increase in productivity, and it is also cost-efficient because all processors utilize the same resources.

- It just enhances the dependability of the computer system.

16. Explain the structure of RAID in an operating system.

Redundant Arrays of Independent Disks(RAID) is a technology that combines multiple physical hard drives into a single logical unit to improve data storage performance, reliability, and capacity. It uses techniques such as data striping (spreading data across multiple disks), mirroring (creating identical copies of data on separate disks), or parity (calculating and storing error-checking information) to achieve these benefits.

RAID is employed to enhance data security, system speed, storage capacity, and overall efficiency of data storage systems. It aims to ensure data redundancy, which helps minimize the risk of data loss in case of disk failure.

17. What Are the Various Levels of RAID?

- **RAID 0**: Striping without redundancy: This configuration is implemented to enhance the server's performance.

- **RAID 1**: Reflecting and duplicating: Also called disk mirroring, this level is seen as the easiest method to achieve fault tolerance.

- **RAID 2**: employs memory-based error-correcting codes, typically employing hamming code parity as linear error correction.

- **RAID 3**: Bit-interleaved Parity: This tier mandates a separate parity drive for storing data.

- **RAID 4**: It stores all parity data on one specific drive.

- **RAID 5**: Block-interleaved distributed Parity: This level offers superior performance compared to disk mirroring while also providing fault tolerance.

- **RAID 6 P+Q Redundancy:** Typically, this level offers protection against up to two drive failures.

18. What Is Process Scheduling?

Process scheduling in a multiprogramming environment involves the operating system (OS) managing the allocation of CPU resources among multiple processes. This task includes tracking the CPU's status and deciding which process should run next based on scheduling algorithms. The scheduler is responsible for this oversight, ensuring efficient utilization of CPU resources by allocating the CPU to processes and reclaiming it when processes are complete or are suspended.

19. What Activities Are Executed by OS in Device Management?

In device management, the operating system (OS) oversees communication with various hardware devices using specialized drivers. It maintains an inventory of all connected devices, manages access permissions for processes to use specific devices, and allocates CPU time and memory resources to optimize device usage. The OS also handles error conditions and ensures the reliability and stability of device operations within the system.

20. What Is SMP?

SMP, or Symmetric Multi-Processing, represents a prevalent configuration in multiprocessor systems where each processor executes an identical operating system

instance. These instances collaborate as necessary to ensure efficient resource utilization.

21. Briefly Explain FCFS.

FCFS, or First-come, first-served, is a scheduling algorithm where processes are serviced in the order they request CPU time, utilizing a FIFO (First In, First Out) queue to manage the execution sequence.

22. What Is the RR Scheduling Algorithm?

The RR (Round-Robin) scheduling algorithm is designed for time-sharing systems, where each process receives a small unit of CPU time (time quantum), typically ranging from 10 to 100 milliseconds before the CPU scheduler moves on to the next process in a circular queue.

23. What Is Batch Processing?

Batch processing is a technique where an Operating System collects programs and data together in a batch before processing starts. The OS performs the following activities related to batch processing:

- Defines a job consisting of a predefined sequence of commands, programs, and data as a single unit.
- Keeps multiple jobs in memory and executes them without manual intervention.
- Processes jobs in the order they are submitted (first-come, first-served).
- Releases memory once a job completes execution and stores its output in an output spool for later printing or processing.

24. What Is Spooling?

Spooling (Simultaneous Peripheral Operations On-line) involves buffering data from various I/O jobs in a designated memory or hard disk accessible to I/O devices.

In a distributed environment, an operating system handles spooling by:

- Managing the varying data access rates of I/O devices.

- Maintaining a spooling buffer as a waiting area for data allows slower devices to catch up.

- Facilitating parallel computation through spooling, enabling the computer to perform I/O operations concurrently, such as reading from a tape, writing to a disk, and printing to a printer while performing computational tasks.

25. What Is a Pipe, and When Is It Used?

A pipe is a method of inter-process communication (IPC) that establishes a unidirectional channel between two or more related processes. It allows the output of one process to be used as the input for another process. Pipes are used when processes need to communicate by passing data sequentially, commonly seen in command-line operations where the output of one command is piped to another command.

26. What Are the Different Kinds of Operations Possible on Semaphore?

There are two basic atomic operations possible:

- *Wait()*

- *Signal()*

27. What Is Thrashing?

Thrashing refers to the severe degradation of computer performance when the system spends more time handling page faults than executing actual transactions. While handling page faults is a normal part of using virtual memory, excessive page faults lead to thrashing, significantly negatively impacting system performance.

28. What Does a Bootstrap Program Do in an Operating System?

A bootstrap program is a program that starts the operating system when a computer system is turned on, being the initial code executed at startup. This process is often called booting. The proper functioning of the operating system depends on the bootstrap program. The program is stored in the boot sector at a specific location on the disk. It finds the kernel, loads it into the primary memory, and initiates its execution.

29. What Is Virtual Memory?

Virtual memory creates the illusion of a large, contiguous address space for each user, starting from address zero. It extends the available RAM using disk space, allowing running processes to operate seamlessly regardless of whether the memory comes from RAM or disk. This illusion of extensive memory is achieved by dividing virtual memory into smaller units, called pages, which can be loaded into physical memory as processes need.

30. What Is a Kernel?

A kernel is the central component of an operating system that is responsible for managing computer hardware and software operations. It oversees memory management, CPU time allocation, and device management. The kernel is the core interface between applications and hardware, facilitating tasks through inter-process communication and system calls.

31. Is It Possible to Have a Deadlock Involving Only One Process? Explain Your Answer.

No, a deadlock with just one process is not possible. Here's why: A deadlock situation arises if four specific conditions are met simultaneously within a system:

List the Coffman's conditions that lead to a deadlock:

- **Mutual Exclusion:** Only one process can use a critical resource at any given time.
- **Hold and Wait:** A process can hold some resources while waiting for others.
- **No Preemption:** Resources cannot be forcibly taken from a process holding them.

- **Circular Wait:** A closed chain of processes exists where each process holds at least one resource required by the next process.

32. How Are Server Systems Classified?

Server systems are categorized into computer-server systems and file-server systems. In computer-server systems, an interface is provided for clients to request actions or services. In file-server systems, clients are given access to create, retrieve, update, and delete files stored on the server.

33. What Does a Dispatcher Do?

The dispatcher is a component that hands over CPU control to the process chosen by the short-term scheduler. This procedure includes:

- Changing the situation.
- Changing to user mode
- Navigate to the appropriate spot in the user program to reset the program.
- Dispatch latency refers to the time required for the dispatcher to transition from one process to another.

34. What Is the Number of Fragmentation Types That Happen in an Operating System?

There exist two forms of fragmentation:

- Internal fragmentation happens when working with allocation units that have a fixed size.
- External Fragmentation happens when variable-size allocation units are being managed.

35. What Does the Memory-Management Unit (MMU) Do?

The MMU is a physical device that translates virtual addresses into physical ones. In this plan, the relocation register's value is added to each address produced by a user process before being sent to memory. User programs work with logical addresses and are unaware of the physical addresses.

36. What CPU Registers Can Be Seen in a Common Operating System Layout?

Some of the CPU registers are mentioned below:

- Storage devices are used to store electrical energy accumulated over time.

- Registers that store specific memory locations for faster access.

- Pointer to the top of the stack.

- Registers with multiple uses.

37. What Is the Process for Establishing a Dual Boot System?

Divide the hard drive to assign distinct sections for each OS to create a dual boot setup. Set up each OS on its assigned partition and employ a boot manager (like GRUB for Linux) to select between them when starting up.

38. Explain the Network Operating System.

A network operating system (NOS) is software that connects various devices and computers, allowing them to access shared resources. The main functions of a NOS are:

- Setting up and handling user accounts within the network.

- Regulating entry to network assets.

- Offering communication services among network devices.

- Observing and rectifying issues within the network.

- Setting up and overseeing network assets.

39. What Kinds of Network Operating Systems Exist?

There are mainly two different types of network operating systems:

- **Peer-to-peer:** This network operating system allows the sharing of resources and files within small networks with restricted resources. LANs make use of these systems.

- **Client/Server:** Network operating systems based on client-server architecture allow access to resources through a central server. Typically, these systems are costlier to set up and uphold and suitable for expansive networks that provide numerous services.

40. Could You Give Me a Few Instances of Network Operating Systems?

Some instances of network operating systems are:

- **Windows Server:** Developed by Microsoft, used for server applications across various environments.

- **UNIX and Linux:** Operating systems with extensive network capabilities widely used in server environments.

- **Novell NetWare:** Developed by Novell, historically significant in networking solutions.

- **IBM OS/2:** Developed by IBM, though less common today, was a network operating system.

- **Mac OS X Server:** Apple's server operating system provides network services.

Note : Enhance your operating systems and network environment to ensure robust performance. Try LambdaTest Now!

Proficient Level Operating System Interview Questions

The below set of operating system interview questions is designed to seek understanding beyond fundamental concepts. These operating system interview questions concern asymmetric clustering, paging, scheduling algorithms, and more.

41. What Is Asymmetric Clustering?

Asymmetric clustering involves a setup with two nodes: one primary (active) and one secondary (standby). The main node handles all operations and processes, while the secondary node remains inactive or performs minimal tasks until it needs to take over if the primary node fails.

42. What Is the Purpose of Paging in Operating Systems?

Paging is a memory management technique within operating systems, allowing processes to access more memory than is physically available. This method enhances system performance, optimizes resource utilization, and reduces the likelihood of page faults. In Unix-like systems, paging is also referred to as swapping.

The primary objective of paging through page tables is to enable efficient memory management by dividing it into smaller, fixed-sized units called pages. This approach allows the computer to allocate memory more effectively than contiguous memory blocks for each process.

43. Explain the Idea of Demand Paging

Demand paging is a method employed by operating systems to improve memory utilization. In demand paging, only the essential program pages are loaded into memory as needed instead of loading the entire program altogether. This method reduces unnecessary memory use and improves the system's overall efficiency.

44. What Is the Equation for Demand Paging?

There isn't a standard equation for demand paging like the one provided. Effective Access Time (EAT) can involve factors like memory access time, page fault rate, and disk access time, but it's not represented by the formula given: $EAT = (1 - p) * m + p * s$.

45. What Is the Definition of RTOS?

An RTOS is intended for real-time tasks that require data processing to be finished within a set and short timeframe. Real-time operating systems are highly effective in carrying out tasks that require speedy completion. It efficiently manages the execution, monitoring, and control procedures. It also requires less memory and uses fewer resources.

46. What Do Schedulers Do?

Schedulers are system software responsible for managing the execution of processes in a computer system. They ensure efficient utilization of the CPU by determining which processes should run when they should run, and for how long. There are generally three types of schedulers:

- **Long-term scheduler (Job scheduler):** Selects which processes are to be brought into the ready queue for execution, considering factors like system load and process priorities.

- **Short-term scheduler (CPU scheduler):** Determines which ready, in-memory process will be executed next and allocates CPU time to them.

- **Medium-term scheduler:** Handles processes that have been swapped out of main memory to reduce the degree of multiprogramming, typically due to memory constraints.

47. What Does Priority-Based Scheduling Involve?

In batch systems, a non-preemptive priority scheduling algorithm is commonly utilized. Every process has a priority assigned to it, and the one with the highest priority is the first to execute it. If several processes have equal priority, they are carried out in the

order they arrived. Memory requirements, time requirements, or other resource needs can help establish priorities.

48. What Is the Two-State Process Model?

The two-stage process model consists of running and non-running states as described below:

- **Running:** When creating a new process, it enters the system in the running state.
- **Not Running:** Processes not currently running are placed in a queue, awaiting their turn to execute. Each queue entry points to a specific process, and the queue is implemented using a linked list. The dispatcher transfers interrupted processes to the waiting queue. If a process completes or is aborted, it is discarded. The dispatcher then selects a process from the queue to execute.
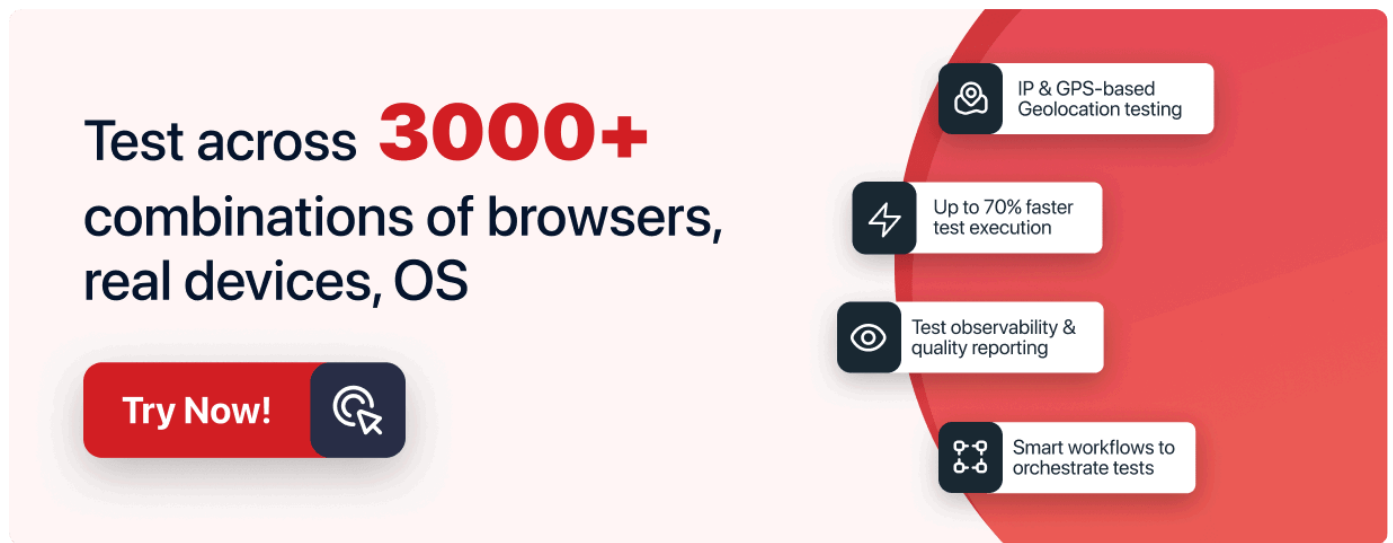
49. What Kinds of Scheduling Algorithms Exist?

Various kinds of scheduling algorithms are available.

- Processes are handled based on the order of their arrival in the First Come, First Serve (FCFS) scheduling algorithm.
- Every process is allocated a set duration of time, known as a quantum.
- Shortest Job First (SJF) is a scheduling algorithm that prioritizes processes with the shortest execution time.
- PS selects processes based on their priority value (nice value), which ranges from 0 to 99, where 0 is the highest priority and 99 is the lowest.

Scheduling algorithms are crucial for managing concurrent tasks efficiently, especially when running multiple test suites or large application simulations. However, lower RAM and older operating systems may struggle with heavy applications, causing performance lag. One solution is to use VMware or VirtualBox for local installations or a cloud-based platform like LambdaTest that offers various OS options.

This platform allows you to run heavy applications and multiple test suites concurrently without the need to maintain them locally. LambdaTest is an AI-powered test platform that lets you run manual and automated tests at scale with over 3000+ real devices, browsers, and OS combinations.

50. Explain the Function of Multiple-Level Queues Scheduling.

Multiple-level queues do not function as a standalone scheduling algorithm. They use other pre-existing algorithms to categorize and arrange tasks with common traits.

- Several queues are kept for processes with similar characteristics.

- Every queue can use its scheduling algorithm.

- Each queue is given specific priorities.

- For example, tasks that require a lot of processing power could be placed in one queue, while tasks that rely heavily on input/output operations are placed in a separate queue. The Process Scheduler switches back and forth between the queues, allocating tasks to the CPU based on the specific algorithm assigned to each queue.

51. Describe Threads at the User Level.

Regarding user-level threads, the kernel does not know they exist. The thread library contains functions for making and deleting threads, exchanging messages and data among threads, managing thread scheduling, and storing and recovering thread contexts. The application starts with only one thread.

52. Describe Threads That Are Managed Directly by the Operating System at the Kernel Level.

The kernel manages thread management for kernel-level threads. There is no code for managing threads in the application area. The operating system directly supports kernel threads, enabling any application to be designed with multiple threads. All threads in a program are controlled in one process.

The kernel stores context data for the whole process and each thread. The kernel performs scheduling on a thread-by-thread basis. The kernel creates, schedules, and manages threads in kernel space. Creating and managing kernel threads typically have slower performance than user threads.

53. What Is the Difference Between Multithreading and Multitasking?

Below are the differences between multithreading vs multitasking in simple form.

- Multithreading

  - Multiple threads run simultaneously within the same program or different parts of it.

  - The CPU alternates between various threads.

  - It represents a lightweight process.

  - It is a characteristic of the process.

  - Multithreading involves sharing computing resources among threads of a single process.

- Multitasking

  - Multiple programs are executed at the same time.

  - The CPU alternates between different tasks and processes.

  - It represents a heavyweight process.

  - It is a characteristic of the OS.

- Multitasking involves sharing computing resources (CPU, memory, devices, etc.) among processes.

54. What Is Peterson's Approach?

Peterson's approach is a concurrent programming algorithm used to synchronize two processes to maintain mutual exclusion for shared resources. It uses two variables: a size two boolean array flag and an integer variable turn.

55. What Is Banker's Algorithm?

The Banker's algorithm is a resource allocation and deadlock avoidance algorithm. It ensures system safety by simulating resource allocation for the maximum possible amounts of all resources, performing an "s-state" check to verify potential activities before deciding whether to proceed.

56. What Is the Many-to-Many Model?

The many-to-many model allows multiple user threads to be mapped to an equal or smaller number of kernel threads. This threading model shows a scenario where six user-level threads interact with six kernel-level threads. Developers can create numerous user threads, and the corresponding kernel threads can run in parallel on a multiprocessor system. This model optimizes concurrency, allowing the kernel to schedule another thread for execution if one thread performs a blocking system call.

57. What Is the Many-to-One Model?

The many-to-one model maps several user-level threads to a single kernel-level thread. Thread management is handled in user space by the thread library. The entire process is blocked if a thread makes a blocking system call. Only one thread can interact with the kernel at any given time, preventing multiple threads from running concurrently on multiprocessors. If user-level thread libraries are implemented in an operating system that does not natively support them, kernel threads utilize the many-to-one relationship mode.

58. What Is the One-to-One Model?

The one-to-one model establishes a direct relationship between each user and kernel-level thread. This model offers greater concurrency than the many-to-one model, allowing another thread to run if one thread makes a blocking system call. It supports the execution of multiple threads in parallel on multiprocessors. However, a drawback of this model is that creating a user thread necessitates a corresponding kernel thread. Operating systems such as OS/2, Windows NT, and Windows 2000 utilize this one-to-one relationship model.

59. What Is a RAID Controller?

A RAID controller acts as a supervisor for the hard drives within a large storage system. It sits between the computer's operating system and the physical hard drives, organizing them into groups for easier management. This arrangement enhances data transfer speeds and provides protection against hard drive failures, thereby ensuring both efficiency and data integrity.

60. What Is Worst-Fit Memory Allocation in an Operating System?

In this technique, the system scans the entire memory to find the largest available space or partition and assigns the process to this largest area. This method is time-consuming as it requires checking the entire memory to identify the largest available space.

61. What Is the Best-Fit Memory Allocation?

This method organizes the list of free and occupied memory blocks by size, from smallest to largest. The system searches through the memory to find the smallest free partition that can fit the job, promoting efficient memory use. Jobs are arranged in order from the smallest to the largest.

62. What Are the Types of Segmentation in an Operating System?

Below are the two segments of an operating system.

- **Virtual Memory Segmentation:** Each process is divided into multiple segments, which may be allocated incrementally and possibly at runtime.

- **Simple Segmentation:** Each process is divided into segments loaded into memory at runtime, not necessarily in contiguous blocks.

In segmentation, there is no direct relationship between logical and physical addresses. All segment information is stored in a table called the Segment Table.

Expert Level Operating System Interview Questions

The below set of operating system interview questions is designed for experts seeking advanced understanding beyond fundamental and proficient concepts. These operating system interview questions concern process scheduling algorithms, memory management techniques, file system structures, and more. Designed to assess proficiency in managing complex OS scenarios, they aim to evaluate the ability to troubleshoot and optimize system performance effectively.

63. Explain Memory Management

Memory management is a crucial function of an operating system that manages primary memory, facilitating the transfer of processes between main memory and disk during execution. It monitors every memory location, whether allocated to a process or free. Memory management determines the amount of memory allocated to processes, decides the timing of memory allocation, and updates the status whenever memory is freed or unallocated.

64. What Are the Issues Related to Concurrency?

Below are some concurrency issues related to operating systems.

- **Non-atomic Operations:** Interruptible operations by multiple processes can cause issues.

- **Race Conditions:** It occurs when the outcome depends on which process reaches a certain point first.

- **Blocking:** Processes can become blocked while waiting for resources, leading to potential long periods of inactivity, especially when waiting for terminal input, which can be problematic if the process needs to update data periodically.

- **Starvation:** This happens when a process cannot obtain the necessary resources to make progress.

- **Deadlock:** This occurs when two processes are blocked, and execution cannot proceed.

65. What Are the Drawbacks of Concurrency?

Some of the drawbacks of concurrency are mentioned below.

- It necessitates protecting multiple applications from one another.

- It requires additional mechanisms to coordinate multiple applications.

- Operating systems need extra performance overheads and complexities to switch between applications.

- Running too many applications simultaneously can lead to significantly degraded performance.

66. What Is Seek Time?

Seek time is the duration required for the disk arm to move to a specific track where the data needs to be read or written. An optimal disk scheduling algorithm minimizes the average seek time.

67. How to Calculate Performance in Virtual Memory?

The performance of a virtual memory system depends on the total number of page faults, which are influenced by "paging policies" and "frame allocation." Effective access time = (1-p) x Memory access time + p x page fault time.

68. What Is Rotational Latency?

Rotational latency is the time required for the desired disk sector to rotate into position so it can be accessed by the read/write heads. A disk scheduling algorithm that minimizes rotational latency is considered more efficient.

69. What Is the Purpose of Having Redundant Data?

Despite taking up more space, data redundancy increases the reliability of disks. In case of a disk failure, duplicating the data on another disk allows for data retrieval and continued operations. On the other hand, losing one disk could jeopardize the whole dataset if data is spread out over many disks without RAID.

70. What Are the Key Evaluation Points for a RAID System?

RAID operates transparently with the underlying system. This allows it to appear to the host system as a large single disk structured as a linear array of blocks. This seamless integration enables replacing older technologies with RAID without requiring extensive changes to existing code.

Key Evaluation Points for a RAID System:

- **Reliability:** How many disk faults can the system withstand?

- **Availability:** What proportion of total session time is the system operational (uptime)?

- **Performance:** What is the responsiveness and throughput of the system?

- **Capacity:** How much usable capacity is available to the user given N disks with B blocks each?

71. How RAID Works?

Consider how RAID operates with an analogy: Imagine you have several friends and want to safeguard your favorite book. Instead of entrusting the book to just one friend, you make copies and distribute segments to each friend. If one friend loses their segment, you can still reconstruct the book from the other segments. RAID functions similarly to hard drives by distributing data across multiple drives. This redundancy ensures that if one drive fails, the data remains intact on the others. RAID effectively safeguards your information, like spreading your favorite book among friends to keep it secure.

List the various file operations. File operations include:

- Creating

- Opening

- Reading

- Writing

- Renaming

- Deleting

- Appending

- Truncating

- Closing

72. What Methods Do Operating Systems Typically Use To Identify and Authenticate Users?

Operating Systems typically recognize and authenticate users through the following three methods:

- **Username / Password:** Users must input a registered username and password to access the system.

- **User Card/Key:** Users need to insert a card into a card slot or enter a key generated by a key generator into a specified field to log in.

- **User Attribute:** Fingerprint/Eye Retina Pattern/Signature: Users must provide a specific attribute via an input device to gain system access.

73. What Is the Goal of the Process Scheduling Algorithm?

Below are the goals for ensuring the process scheduling algorithm.

- Ensure the CPU remains utilized efficiently at all times.

- Equitably distribute CPU resources among processes.

- Increase the number of processes completing execution within a specified timeframe.

- Minimize the time it takes for a process to finish execution to reduce turnaround time.

- Reduce waiting time: Ensure processes in the ready queue promptly receive necessary resources.

- Improve response time: Decrease the duration for a process to generate its initial response.

74. What Are the Various Terms to Consider in Every CPU Scheduling Algorithm?

Some of the various terms to take into account in every CPU scheduling are:

- **Arrival Time:** The moment a process joins the ready queue.

- **Finish Time:** The moment when a process completes its execution.

- **Burst Time:** The time needed for a process to run on the CPU.

- **Turnaround Time:** The difference between the finish time and the arrival time of a process.

- **Waiting Time (W.T.):** The difference between the turnaround time and the burst time of a process.

- **Waiting Time:** The time spent waiting, which is the difference between the completion time of a task and the total time taken for its execution.

75. What Is the Need for CPU Scheduling Algorithms?

CPU scheduling selects which process will control the CPU when another process is stopped. The main objective of CPU scheduling is to ensure that the CPU is constantly in use by having the operating system choose a process from the ready queue whenever the CPU is not busy. In a multi-programming setting, if the long-term scheduler selects several I/O-bound tasks, the CPU could be inactive for long durations. A proficient scheduler seeks to optimize resource usage.

76. What Are Starvation and Aging in an OS?

Below is a clear explanation of starvation and aging in OS.

- **Starvation:** This is a resource management issue where a process does not receive the resources it needs for a long time because those resources are allocated to other processes.

- **Aging:** Aging is a technique to prevent starvation by gradually increasing the priority of waiting processes. As time progresses, the priority of the waiting request increases, ensuring it eventually becomes the highest priority.

77. What Is Cycle Stealing?

Cycle stealing is where computer memory (RAM) or the bus is accessed without interfering with the CPU. It is similar to direct memory access (DMA), allowing I/O controllers to read or write RAM without the CPU's intervention.

78. How Are One-Time Passwords Implemented?

In a One-Time Password (OTP) system, a unique password is required to log in each time. Once used, the OTP cannot be reused. OTPs are implemented through various methods:

- **Random Numbers:** Users are given cards with numbers and corresponding alphabets. The system requests numbers that match randomly selected alphabets.
- **Secret Key:** Users receive a hardware device that generates a secret ID linked to their user ID. The system asks for this secret ID each time they log in.
- **Network Password:** Some commercial applications send OTPs to the user's registered mobile or email, which must be entered before logging in.

79. What Are the Different Types of CPU Scheduling Algorithms Available?

There are primarily two kinds of scheduling techniques.

- Preemptive scheduling is applied when a process transitions from a running or waiting state to a ready state.
- Non-preemptive scheduling is used when a process finishes execution or moves from running to a waiting state.

80. What Are the Names of Synchronization Techniques?

Below are the names of some synchronization techniques.

- Mutexes
- Condition variables
- Semaphores
- File locks

81. Define the Term Bounded Waiting

A system adheres to bounded waiting conditions if a process that wants to enter a critical section is ensured to be allowed to do so within a finite amount of time.

82. What Are Some Examples of Program Threats?

- **Trojan Horse:** This program captures user login credentials and stores them, sending them to a malicious user who can access system resources.

- **Trap Door:** A program that functions as expected but contains a security flaw, allowing unauthorized actions without the user's knowledge.

- **Logic Bomb:** A situation where a program behaves maliciously under certain conditions but functions normally otherwise, making it difficult to detect.

- **Virus:** Viruses can replicate themselves on a computer system, potentially modifying or deleting user files and crashing systems. They are usually small codes integrated into programs that spread as the program is used, eventually making the system unusable.

83. What Are the Types of Process Schedulers?

There are three types of process schedulers:

- **Long-Term or Job Scheduler:** This scheduler introduces new processes into the 'Ready State.' It manages the degree of multiprogramming by controlling the number of processes in the ready state at any given time. It's crucial for the long-term scheduler to carefully choose a mix of I/O-bound and CPU-bound processes.

- **Short-term or CPU Scheduler:** This scheduler selects a process from the ready state to be scheduled in the running state. Note that the short-term scheduler only selects the process but does not load it into the running state. This is where various scheduling algorithms are applied. The CPU scheduler ensures no processes starve due to high burst time demands.

- **Medium-Term Scheduler:** This scheduler is responsible for suspending and resuming processes and handles swapping (moving processes between main memory and disk). Swapping might be required to optimize the process mix or to free up memory when memory requirements exceed availability. It helps maintain a balance between I/O-bound and CPU-bound processes.

84. Can You Explain What a Zombie Process Is?

A zombie process is a process that has finished running but remains in the process table to relay its status to the parent process. Once a child process completes its execution, it transforms into a zombie state until its parent process retrieves its exit status, causing the child process entry to be eliminated from the process table.

85. What Do Orphan Processes Refer To?

An orphan process occurs when its parent process ends before the child process, resulting in the child process being without a parent.

86. What Is a Trap and a Trapdoor?

Below are the clear definitions of a trap and trapdoor in OS.

- **Trap:** A software interrupt known as a trap typically arises from an error situation and holds the highest priority as a non-maskable interrupt.
- **Trapdoor:** An undisclosed entryway in a program enables access without the need for typical authentication methods.

87. What Is the Function of Compaction Within an Operating System?

The operating system may discover enough space when assigning memory to a process. Still, it is separated into fragmented sections that are not contiguous, leading

to the inability to fulfill the process's memory needs. The problem of external fragmentation can be solved by using compaction as a technique.

88. What Does Fixed Partitioning Include?

Static or fixed partitioning involves dividing physical memory into partitions of a set size. Every partition is given to a particular process or user when the system starts up and stays assigned to that process until it ends or gives up the partition.

89. What Causes Internal Fragmentation in Fixed Partitioning?

Internal fragmentation occurs when a process is smaller than the allocated partition, leading to unused memory within the partition and inefficient memory utilization.

Mastery Level Operating System Interview Questions

In the operating system interview questions below, you are expected to learn extensively about handling complex OS scenarios, troubleshooting system-level issues, and implementing efficient solutions. These operating system interview questions assess deep understanding and proficiency in advanced OS concepts and principles.

90. What Is the Need for a Page Replacement Algorithm?

In operating systems that use paging for managing memory, page replacement algorithms are crucial for deciding which page to replace when a new page is loaded. If a new page is requested but is not in memory, a page fault happens, which causes the operating system to swap out one of the current pages for the needed new page. Different page replacement algorithms provide distinct approaches for determining which page to replace, all aimed at reducing the occurrences of page faults.

91. What Do You Understand by Belady's Anomaly?

Increasing the number of frames allocated to a process's virtual memory speeds up execution by reducing the number of page faults. However, occasionally, the opposite occurs—more page faults happen as more frames are allocated. This unexpected result is known as Belady's Anomaly. Belady's Anomaly refers to the counterintuitive situation where increasing the number of page frames leads to increased page faults for a given memory access pattern.

92. Why Do Stack-Based Algorithms Not Suffer From Belady's Anomaly?

Stack-based algorithms avoid Belady's Anomaly because they assign a replacement priority to pages independent of the number of page frames. Some algorithms like Optimal, LRU (Least Recently Used), and LFU (Least Frequently Used) are good examples.

These algorithms can also calculate the miss (or hit) ratio for any number of page frames in just one pass through the reference string. In the LRU algorithm, a page is relocated to the top of the stack whenever a page is accessed. Therefore, the top n pages in the stack represent the n pages that have been used most recently. The top of the stack will always hold the n+1 most recently used pages, even when the number of frames is increased to n+1.

93. How Can Belady's Anomaly Be Eliminated?

A stack-based approach can be employed to eliminate Belady's Anomaly. Examples of such algorithms include:

- Optimal Page Replacement Algorithm
- Least Recently Used (LRU) Algorithm

These algorithms operate on the principle that if a page remains inactive for a long period, it is not frequently used. Therefore, replacing this page, improving memory

management, and eliminating Belady's Anomaly is best.

94. What Happens if a Non-Recursive Mutex Is Locked More Than Once?

Deadlock occurs. If a thread that has already locked a mutex attempts to lock it again, it will enter the mutex's waiting list, resulting in a deadlock. This happens because no other thread can unlock the mutex. To prevent this, an operating system implementer can ensure that the mutex's owner is identified, and if the same thread tries to lock it again, it can return the mutex to avoid deadlocks.

95. How To Recover From a Deadlock?

Deadlock recovery can be achieved through the following methods process termination:

- Abort all deadlocked processes.

- Abort one process at a time until the deadlock is resolved.

- Resource preemption.

- Rollback.

- Selecting a victim.

96. What Are File Allocation Methods and Their Main Purpose?

File allocation methods define how files are stored in disk blocks. The three main disk space or file allocation methods are:

- Contiguous Allocation.

- Linked Allocation.

- Indexed Allocation.

The primary goals of these methods are:

- Efficient disk space utilization.

- Fast access to file blocks.

97. How Can the Issue of a Single Index Block Being Unable To Hold All the Pointers for Very Large Files Be Resolved?

For very large files where a single index block cannot hold all the pointers, the following mechanisms can be used:

- **Linked scheme:** The linked scheme links two or more index blocks together to hold the pointers. Each index block contains a pointer or address to the next index block.
- **Multilevel index:** In this approach, a first-level index block points to second-level index blocks, which point to the disk blocks occupied by the file. This can be extended to three or more levels depending on the maximum file size.
- **Combined scheme:** This uses a special block called the Inode (information node), which contains all information about the file, such as name, size, and permissions. The remaining space in the Inode stores disk block addresses containing the actual file data. The first few pointers in the Inode point to direct blocks containing the file data.

The next few pointers point to indirect blocks, which may be single, double, or triple indirect. Single indirect blocks contain the addresses of blocks with the file data. Double indirect blocks contain addresses containing the addresses of the file data blocks.

98. Describe Contiguous Allocation

In contiguous allocation, every file takes up a consecutive series of blocks on the disk. If a file requires n blocks and starts at block b, the file will be allocated blocks in this sequence: b, b+1, b+2, ..., b+n-1. Therefore, by having the initial block address and the file size (in blocks), we can figure out which blocks the file uses.

99. What Is the Method for Implementing the Free Space List?

The system keeps a list of free space to monitor disk blocks that are not assigned to any file or directory. This list can primarily be put into action in the following ways:

- **Bitmap or Bit Vector:** A group of bits where each bit symbolizes a disk block. The bit can have a value of 0 or 1: 0, which signifies that the block is in use, while one signifies that the block is available.

- **Linked List:** In this method, free disk blocks are connected through links, each with a pointer to the next free block. The initial available disk block's block number is saved both on disk and stored in memory for quick access.

- **Grouping:** This method stores the addresses of free blocks in the first free block. The first free block holds the addresses of several free blocks (say n blocks). The first n-1 blocks are free, and the last block contains the address of the next n-free blocks. This makes it easy to find the addresses of a group of free blocks.

- **Counting:** This approach stores the address of the first free disk block and the number of free contiguous blocks that follow it. Each entry in the list includes:

  - The address of the first free disk block.

  - A number indicating how many contiguous free blocks follow.

100. What Is the Importance of Free Space Management Techniques in Operating Systems?

Some of the important free space management techniques in OS are mentioned below.

- **Efficient Use of Storage Space:** These techniques help optimize the use of storage space on hard disks or other secondary storage devices.

- **Easy To Implement:** Some methods, like linked allocation, are easy to implement and require minimal processing and memory resources.

- **Faster Access to Files:** Techniques such as contiguous allocation help reduce disk fragmentation and improve file access times.

101. What Is a Disk Scheduling Algorithm, and What Is Its Importance?

Operating systems manage disk scheduling to schedule I/O requests for the disk. It is also known as I/O scheduling.

Importance of Disk Scheduling in Operating Systems.

- Multiple I/O requests can arrive from different processes, but the disk controller can only handle one request at a time. Therefore, other requests must wait in the queue and be scheduled.
- Requests may be far apart on the disk, causing more disk arm movement.
- Hard drives are among the slowest computer system components, necessitating efficient access.

102. What Is the Disk Response Time?

Response time is the average time a request waits for its I/O operation. The average response time refers to all requests' response times. Variance response time measures how individual requests are serviced relative to the average response time. A disk scheduling algorithm that minimizes variance response time is preferable.

103. What Is the SCAN Algorithm?

The SCAN algorithm moves the disk arm in a specific direction, servicing requests along its path. Once it reaches the disk's end, it reverses direction and services requests again. This algorithm operates like an elevator, also known as the elevator algorithm. Consequently, mid-range requests are serviced more frequently, while those arriving behind the disk arm must wait.

104. What Is the Advantage or Limitation of a Hashed-Page Table?

Some of the advantages and limitations of a Hashed-Page table are mentioned below.

Advantages:

- **Synchronization:** Hashed-page tables can efficiently handle synchronization, making them suitable for scenarios where quick data access and updates are required.

- **Efficiency:** In many instances, hash tables outperform search trees and other table lookup structures. This efficiency makes them widely used in various computer software applications, especially for associative arrays, database indexing, caches, and sets.

Limitation:

- **Hash Collisions:** Collisions are nearly inevitable when hashing a random subset of a large set of potential keys. This can lead to inefficiencies in data retrieval.

- **Inefficiency with Many Collisions:** Hash tables become significantly less efficient when collisions are frequent.

- **No Null Values:** Hash tables, unlike hash maps, do not permit null values.

105. What Is "Locality of Reference"?

Locality of reference refers to the tendency of a computer program to repeatedly access the same set of memory locations over a specific period. Essentially, it means that a program often accesses instructions whose addresses are close to one another.

106. What Is the Advantage of Dynamic Allocation Algorithms?

Some of the advantages of dynamic allocating algorithms are.

- When the exact amount of memory required for a program is unknown beforehand.

- When creating data structures that don't have a fixed upper memory limit.

- To use memory space more efficiently.

- Insertion and deletion in dynamically created lists can be easily managed by manipulating addresses, unlike statically allocated memory, which requires more movement and can lead to memory wastage.

- Dynamic memory allocation is essential when using structures and linked lists in programming.

107. What Are the Components of the Linux Operating System?

The Linux operating system is composed of three primary components:

- **Kernel:** The kernel is the core component of Linux, handling all major functions of the OS. It contains various modules and interfaces directly with the hardware. The kernel provides an abstraction to hide the low-level hardware details from system or application programs.

- **System Library:** These libraries are special functions or programs that allow application programs or system utilities to access the kernel's features. They implement most of the operating system's functionalities without needing kernel module access rights.

- **System Utility:** System utility programs are responsible for performing specialized, individual tasks.

108. What Are the Approaches to Implementing Mutual Exclusion?

Some of the approaches to implementing mutual exclusion in OS are mentioned below.

- **Software Method:** This approach relies on processes themselves to manage mutual exclusion. These methods are typically prone to errors and have high overheads.

- **Hardware Method:** This method uses special-purpose machine instructions to access shared resources. While faster, it doesn't provide a complete solution as it can't ensure the absence of deadlock and starvation.

- **Programming Language Method:** This approach offers support through the operating system or the programming language.

109. Which Elements Decide if a Deadlock Avoidance System Must Employ a Detection

Algorithm?

The frequency of deadlock occurrence when implementing this algorithm is a deciding factor. The second issue concerns the number of processes impacted by deadlock when implementing this algorithm.

Conclusion

An operating system is crucial for computer software and software development, providing a common interface for managing essential computer operations. Without it, programs would require their interfaces and code to perform tasks such as disk storage and network connections, making software development impractical. System software facilitates communication between applications and hardware, ensuring consistent support for various applications and allowing users to interact with system hardware through a familiar interface.

Comprehensive knowledge of operating systems is vital for numerous IT careers. Familiarity with potential operating system interview questions can help candidates prepare effective answers in advance. This tutorial covers over 100+ operating system interview questions and example responses, aiding professionals in enhancing their understanding and readiness for job opportunities in software development.

Frequently asked questions

General →

Can an individual create an OS entirely from scratch?

What is OS fingerprinting in cybersecurity?

Can an OS be run from a USB drive?

Can an operating system support multiple desktop environments?

Author's Profile
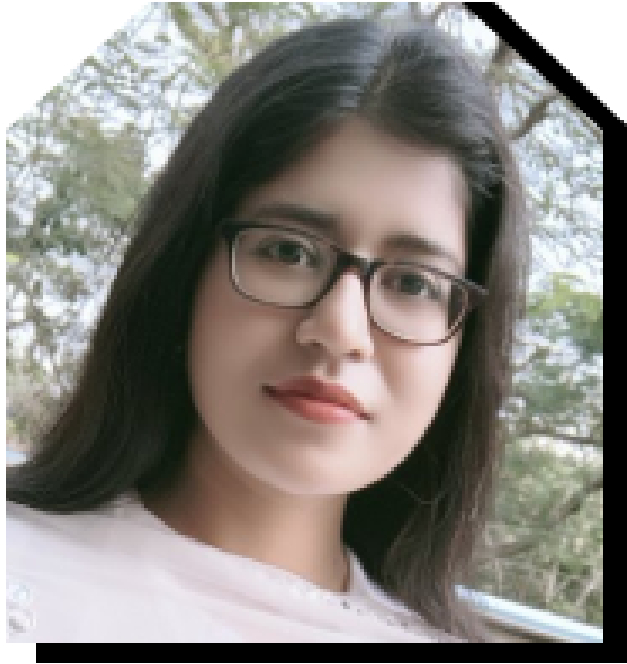
Nazneen Ahmad

Nazneen Ahmad is an experienced technical writer with over five years of experience in the software development and testing field. As a freelancer, she has worked on various projects to create technical documentation, user manuals, training materials, and other SEO-optimized content in various domains, including IT, healthcare, finance, and education. You can also follow her on Twitter.

Hubs: 60

Reviewer's Profile

Saniya Gazala

Saniya Gazala, a Computer Science graduate, brings two years of hands-on experience in manual testing and delving into technical writing, showcasing her detail-oriented and systematic approach. As a technical reviewer, her key responsibilities involve spotting inconsistencies and factual errors in content. Leveraging her in-depth knowledge, she ensures the accuracy and coherence of the material under review.

Did you find this page helpful?

More Hubs
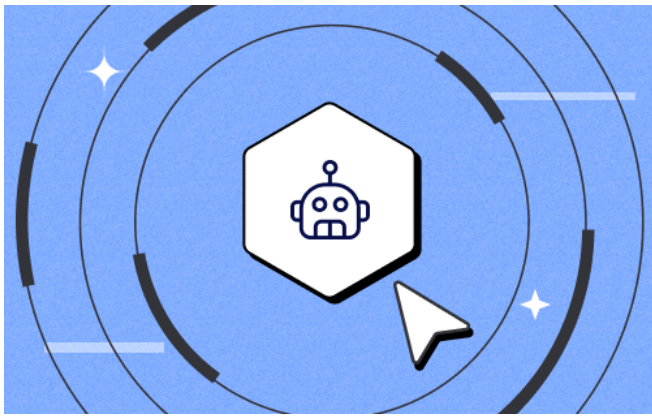
14 Chapters

Automation Testing

The quintessential guide to Automation Testing. This is an in-depth resource covering all aspects of testing.
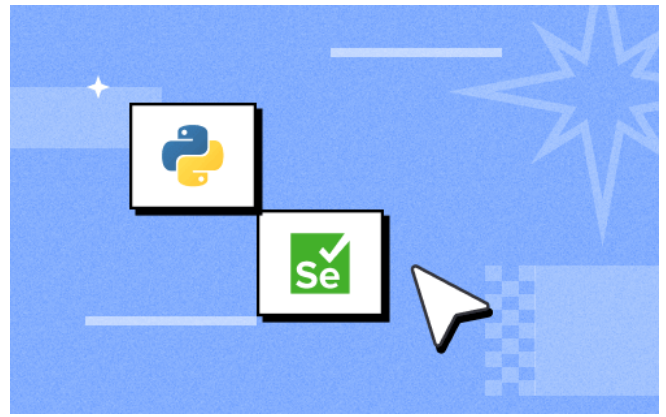
27 Chapters

Selenium Python

Want to learn automation testing with Python? Check out our step-by-step tutorial on Python with examples!

12 Chapters

Selenium Locators

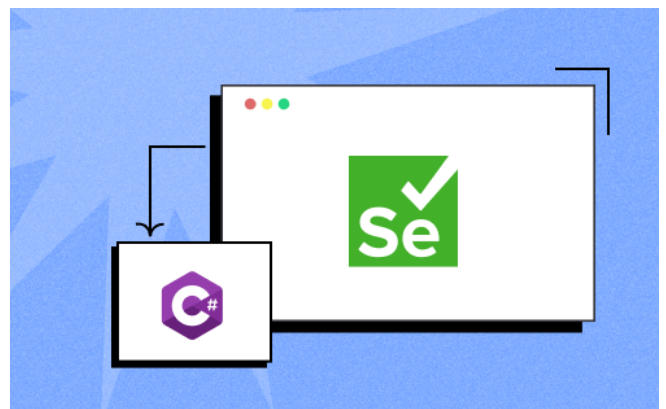Here we explore different types of Selenium locators and learn how they are used with different automation testing.
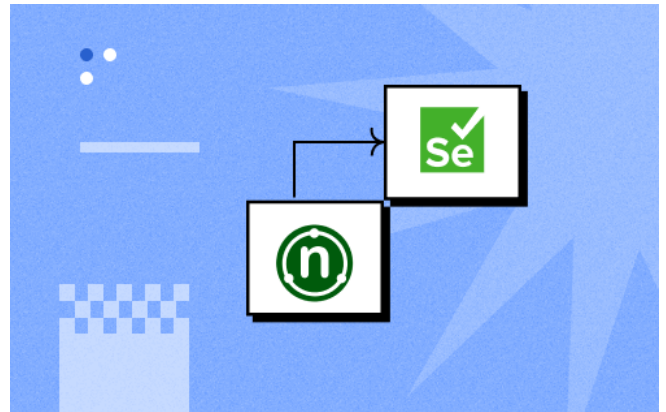
9 Chapters

Selenium NUnit

Isn't it better to use an effective test framework like NUnit on a powerful and scalable Selenium Grid?

10 Chapters

8 Chapters

Selenium WebdriverIO

Through this guide, we will learn how to use WebdriverIO, a next-gen test automation framework for Node.js.

Selenium C#

This tutorial will teach how to master Selenium, making your test automation more streamlined and efficient.

Try LambdaTest Now !!

Get 100 minutes of automation test minutes FREE!!

G Start free with Google

Start free with Email

Products & Features

Automation Testing Cloud

Kane AI - Testing Agent 🔥

Cross Browser Testing

Real Device Cloud

Test Manager

Mobile App Testing

AI-Powered Testing

HyperExecute

LT Browser

LT Debug

Local Page Testing

Automated Screenshots

Geo-Location Testing

Accessibility Testing

Responsive Testing

Localization Testing

Visual Regression Testing

Integrations

Test Analytics

Professional Services New

Test on

iPhone 15

List of Browsers

Internet Explorer

Firefox

Chrome

Safari Browser Online

Microsoft Edge

Opera

Yandex

Mac OS

Mobile Devices

iOS Simulator

Android Emulator

Browser Emulator

Browser Automation

Selenium Testing

Selenium Grid

Cypress Testing

Playwright Testing

Puppeteer Testing

Taiko Testing

Mobile App Automation

Appium Testing

Espresso Testing

XCUITest Testing

Resources

TestMu Conf 2025 🔥

Blogs

Community

Certifications

Product Updates

Newsletter

Webinars

Videos

FAQ

Web Technologies Compatibility

Automation Testing Advisor

Software Testing [Glossary]

Free Online Tools

Mobile Testing Advisor

Sitemap

Status

Company

About Us

Careers 💼

Customers

Press

Achievements 🏆

Reviews

Community & Support

Partners

Open Source

Content Editorial Policy

Write for Us

Become an Affiliate

Terms of Service

Privacy Policy

Trust

Team

Contact Us

Learning Hub

Selenium Tutorial

Cypress Tutorial

Playwright Tutorial

Appium Tutorial

Jest Tutorial

More Learning Hubs

What's New

Changelog

Future of QA Survey Report

Test on iPhone 15

Test on Samsung Galaxy S24 Series

July'24 Updates

Coding Jag - Issue 206

Kayak [Case Study]

File Upload In Selenium [Blog]

Python Assert [Hub]

Appium 101 [Certification]

LAMBDATEST

Deliver unparalleled digital experience with our Next-Gen, AI-powered testing cloud platform. Ensure exceptional user experience across all devices and browsers.

LambdaTest is #1 choice for SMBs and Enterprises across the globe.

LambdaTest has formal standards certification and comply in line with acts and regulations across the globe.