

Operating System Interview Questions and Answers for Freshers

- By Anshika Chaudhary

1. What is an Operating System?

An Operating System (OS) is software that helps people interact with computer hardware and ensures programs run smoothly. It manages things like memory, files, processes, and input/output (like your keyboard, mouse, or printer).

Examples: Windows, macOS, Linux, Android.

2. What services does an OS provide?

The OS offers these main services:

- **User Interface:** Allows users to interact with the computer (like Windows desktop or command line).
- **Program Execution:** Runs and manages applications.
- **I/O Operations:** Manages input/output, such as files and devices.
- **File System Management:** Organizes and controls access to files.
- **Communication:** Manages data exchange between programs or computers.
- **Error Detection:** Identifies issues and alerts users.
- **Resource Allocation:** Allocates resources like memory or CPU time.
- **Security:** Protects data from unauthorized access.

3. Why do we need an Operating System?

An OS has two main purposes:

- To make the computer system work efficiently.
- To provide an environment for running and developing applications.

4. What are the key functions of an OS?

- **Memory Management:** Manages available memory and allocates it to programs.
- **Processor Management:** Determines which processes use the CPU and for how long.
- **Device Management:** Manages hardware like printers, USBs, etc.
- **File Management:** Controls how data is stored and retrieved in files.
- **Security:** Prevents unauthorized access to data and programs.

5. What types of Operating Systems are there?

- **Batch OS:** Processes tasks in groups without user interaction (like early computers).
- **Time-Sharing OS:** Shares CPU time among users for multitasking (e.g., modern personal computers).
- **Distributed OS:** Runs across multiple machines but appears as a single system (e.g., Google's server network).
- **Real-Time OS:** Processes data immediately, ideal for critical applications (e.g., in cars or medical devices).

6. What is a socket in OS?

A socket is a communication endpoint that lets two applications exchange data over a network. Think of it as a two-way street that connects devices over the internet.

7. What are the main activities an OS performs?

- **Memory Management:** Keeps track of who uses which memory and allocates/deallocates as needed.
- **Processor Management:** Schedules which processes get the CPU and for how long.
- **Device Management:** Manages communication with hardware using device drivers.
- **File Management:** Controls file storage, location, and access.
- **Security:** Blocks unauthorized access and protects data.

8. What is a Batch Operating System?

In a Batch OS, tasks with similar needs are grouped and processed together. The user doesn't interact directly with the system; instead, an operator manages the job batches.

9. Advantages and Disadvantages of Batch OS

Advantages:

- Efficient for repetitive tasks.
- Good for large volumes of similar tasks.

Disadvantages:

- Hard to debug.
- If one job fails, others are delayed.

10. What is a Time-Sharing Operating System?

A Time-Sharing OS divides CPU time between users, making multitasking possible. Each task gets a "time slice" to ensure all processes run smoothly.

11. Advantages and Disadvantages of Time-Sharing OS

Advantages:

- Fair CPU time for each user.
- Reduces idle CPU time.

Disadvantages:

- Requires good security for shared data.

12. What is a Distributed OS?

A Distributed OS connects multiple computers so they work together as one system. This allows users to access files or software on another computer within the network, enabling remote work and collaboration.

13. Advantages and Disadvantages of Distributed OS

Advantages:

- High speed and reliability.
- Easy to add more devices to the network.

Disadvantages:

- Network failure can disrupt communication.

14. What is a Real-Time Operating System?

Real-Time OS processes tasks immediately after receiving data. It's ideal for critical applications where timing is crucial, like in air traffic control.

15. Types of Real-Time OS

- **Hard Real-Time:** Strict timing for life-saving systems (e.g., airbag deployment).
- **Soft Real-Time:** Less strict timing requirements (e.g., live video streaming).

16. Advantages and Disadvantages of Real-Time OS

Advantages:

- Efficient use of resources.
- Reliable and error-free operation.

Disadvantages:

- Requires complex algorithms.

17. What is a kernel in OS?

The kernel is the core part of the OS that manages system resources and communication between hardware and software.

18. What is a monolithic kernel?

A monolithic kernel includes all OS services in a single large block, leading to faster communication within the OS but making it harder to debug.

19. Define a process.

A process is a program in execution. For example, when you open a browser, it becomes a process running on your computer.

20. What are the different states of a process?

- **New:** Process is being created.
- **Running:** Process is executing.
- **Waiting:** Process is waiting for resources.
- **Ready:** Process is ready to execute.
- **Terminated:** Process has finished execution.

21. What is the difference between a process and a program?

- A **program** is a set of instructions (like a recipe).
- A **process** is a running instance of that program (like cooking the recipe).

22. What is aging in OS?

Aging is a technique to prevent a process from "starving" (waiting endlessly) for resources by gradually increasing its priority.

23. Advantages of multithreaded programming in OS

- Faster response time.
- Resource sharing within the process.
- Efficient in multiprocessor systems.

24. What are System Calls?

System calls are functions that programs use to request services from the OS, like opening a file or accessing hardware.

25. Why use an API over System Calls?

APIs are easier to use, more portable, and make programming faster and less complex.

26. What is the System Interface's role?

It provides a bridge between application calls and system calls, simplifying access to OS services for programmers.

27. Different scheduling algorithms in OS

- **First-Come, First-Served (FCFS):** Jobs are completed in the order they arrive.
- **Shortest-Job-Next (SJN):** The shortest tasks are prioritized.
- **Priority Scheduling:** Tasks are prioritized by importance.
- **Round Robin:** Each task gets an equal "time slice."
- **Multiple-Level Queues:** Tasks are grouped into priority queues.

28. Objective of multiprogramming in OS

Multiprogramming aims to keep the CPU busy by holding multiple tasks in memory so that the CPU always has a job to execute.

29. Benefits of multithreaded programming in OS

Threads allow a program to perform multiple tasks simultaneously, improving performance and efficiency.

30. Types of System Calls in OS

- **Process Control:** Starts or stops programs.
- **File Management:** Handles files.
- **Device Management:** Manages hardware.
- **Information Maintenance:** Tracks system information.
- **Communication:** Manages data exchange between processes.