#	Question	Suggested Answer	What Interviewers Look For
1	Explain the difference between a process and a thread.	A process is an independent executing program; a thread is a lightweight sub-process. Threads share the same memory space.	Conceptual clarity, real-world relevance, and OS fundamentals understanding.
2	What are the four necessary conditions for a deadlock?	Mutual exclusion, hold and wait, no preemption, and circular wait.	Understanding of Coffman conditions and deadlock prerequisites.
3	How can deadlock prevention be achieved?	By violating at least one of the four necessary conditions (e.g., eliminating circular wait).	Knowledge of prevention strategies and trade-offs.
4	What is deadlock avoidance?	The OS dynamically checks if allocating resources would lead to an unsafe state (e.g., Banker's Algorithm).	Familiarity with proactive avoidance mechanisms.
5	What is the Banker's Algorithm?	A deadlock avoidance algorithm that checks resource allocation against available resources to ensure a safe state.	Grasp of algorithmic solutions to deadlock.
6	What is a resource allocation graph (RAG)?	A directed graph showing processes, resources, and allocations/requests. A cycle in RAG may indicate deadlock.	Ability to interpret graphs for deadlock analysis.
7	How does the operating system detect deadlocks?	The OS employs techniques such as resource allocation graphs and wait-for graphs to detect cycles, which may indicate potential deadlocks.	Awareness of cycle detection techniques and graph analysis.
8	What is the role of the wait-for graph in deadlock detection?	A wait-for graph is derived by removing resource nodes from a resource allocation graph. A cycle in this graph signals a deadlock.	Understanding Graph Simplification and Its Implications in Detection.
9	What happens when a deadlock is detected?	The OS can break the deadlock by terminating processes, preempting resources, or rolling back processes to a safe state.	Knowledge of recovery strategies and their consequences.
10	What is deadlock recovery?	Deadlock recovery is the process of resolving a deadlock after it has been detected by terminating processes or preempting resources.	Familiarity with system recovery strategies.
11	What are the methods for deadlock recovery?	1. Process termination 2. Resource preemption 3. Rollback to previous safe states	Clarity on recovery mechanisms and trade-offs.

12	What is resource preemption in the context of deadlock recovery?	It involves temporarily taking a resource from a process to resolve a deadlock, often requiring the process to be rolled back.	Insight into aggressive recovery approaches.
13	How does process termination help in deadlock recovery?	By terminating one or more processes in the cycle, resources are released, and the deadlock is broken.	Understanding the impact and criteria of termination.
14	What are the disadvantages of deadlock prevention?	It often leads to reduced resource utilization and system performance due to the constraints it imposes.	Awareness of performance trade-offs.
15	Can starvation occur while handling deadlocks?	Yes, especially in preemption-based systems or when using timeouts, a process may be perpetually denied access.	Understanding the relationship between deadlock and starvation.
16	What is livelock, and how is it different from deadlock?	Livelock is when processes keep changing states without doing useful work, unlike deadlock, where they are blocked.	Conceptual differentiation of livelock, deadlock, and starvation.
17	Is it possible to have a deadlock with a single process?	Yes, if a process waits for a resource it already holds, or due to faulty logic.	Understanding edge cases and programming issues.
18	Can deadlocks occur in single-processor systems?	Yes, deadlocks can happen even on a single CPU due to poor synchronization and resource allocation.	Clarity that deadlocks are not limited to multiprocessor systems.
19	What is a safe state in the context of deadlock avoidance?	A safe state means the system can allocate resources in a way that avoids deadlock.	Grasp of the concept of system safety.
20	How does the system enter an unsafe State?	By allocating resources in a way that cannot guarantee all processes will complete, leading to a potential deadlock.	Understanding risk in dynamic resource allocation.
21	What is the difference between a safe and an unsafe state?	A safe state ensures deadlock-free execution; an unsafe state may or may not lead to deadlock.	A nuanced understanding of system state classification.

22	How do real-time systems handle deadlocks?	They avoid complex locking or use static resource allocation to prevent timing violations.	Awareness of special considerations in real-time environments.
23	Why is deadlock detection not always used?	Because it can be resource-intensive and may disrupt system performance.	Real-world implications of choosing detection strategies.
24	What is the impact of deadlock on system performance?	Processes stop making progress, causing system slowdown and wasted resources.	Clear recognition of practical consequences.
25	What are circular wait conditions in deadlocks?	A condition where each process is waiting for a resource held by another in a circular chain.	Understanding of cycle formation and its implications.
26	What is the simplest way to break the circular wait?	By defining a fixed order in which resources must be acquired.	Knowledge of deadlock prevention through ordering.
27	How does multithreading affect deadlocks?	It increases the likelihood of deadlocks due to more shared resources and concurrent execution.	Understanding deadlocks in concurrent programming
28	How can timeouts help prevent deadlocks?	They force the release of resources or abort the process after waiting too long.	Familiarity with timeout mechanisms.
29	What are atomic operations and their role in deadlock?	They execute without interruption, helping prevent inconsistent states and deadlocks.	Grasp of atomicity in concurrency control.
30	How do semaphores relate to deadlocks?	Misuse of semaphores (e.g., not releasing them) can lead to deadlocks.	Understanding of synchronization primitives.