1. The four conditions that create deadlock:

Mutual Exclusion: At least one resource must be held in a non-shareable mode, meaning only one process can use the resource at any time.

Hold and Wait: A process holding at least one resource is waiting to acquire additional resources held by other processes.

No Preemption: Resources cannot be forcibly removed from processes holding them; they must be released voluntarily.

Circular Wait: A circular chain of processes exists where each process holds a resource the next process in the chain needs.

2. How can the circular wait condition be prevented?

Circular wait can be prevented by imposing a total ordering on resources and requiring processes to request resources in an increasing order of enumeration.

3. Differences among deadlock avoidance, detection, and prevention:

Deadlock Avoidance: Ensures that the system only allocates resources in a way that avoids entering an unsafe state by using algorithms like the Banker's Algorithm.

Deadlock Detection: Allows deadlocks to occur but uses detection algorithms to find them, enabling recovery actions afterward.

Deadlock Prevention: Proactively prevents deadlocks by structurally negating one of the deadlock conditions, such as by using resource ordering to avoid circular wait.

4. Banker's Algorithm – Safe and Unsafe States

Safe State: A system is in a safe state if it can allocate resources to each client in some order, ensuring that all requests can be safely met without leading to deadlock.

Unsafe State: An unsafe state means the system may not be able to meet all requests in some sequence, leading to a potential deadlock.

Requests Analysis:

Starting with 10 million dollars:

Starting with 10 million dollars:

Request 1: Client B requests 3 million (Remaining: 7 million)

Request 2: Client C requests 2 million (Remaining: 5 million)

Request 3: Client D requests 1 million (Remaining: 4 million)

Request 4: Client C requests 1 million (Remaining: 3 million)

Request 5: Client D requests 2 million (Remaining: 1 million)

Request 6: Client B requests 2 million (Remaining: -1 million)

Result: Request 6 will lead to an unsafe state because it leaves the banker with insufficient funds (negative balance), creating a risk of deadlock.