Deadlock

A deadlock is a situation in an operating system where two or more processes are unable to proceed because each is waiting for the other to release a resources Essentially, it's a standstill where no process can move forward.

#Conditions for Deadlock:

1. Mutual Exclusion: Only one process can use a resource at any given time.

2. Hold and wait: A process holding at least one resources is waiting to acquire additional resources that are currently being held by other process.

3.No Perception: A resource can only be released voluntarily by the process holding it.

4.Circular Wait: A set of processes must exist such that each process is waiting for a resource held by the next process in the set.

# Fixing Deadlocks:

There are several strategies to handle deadlocks:

1 Deadlock Prevention: Modify the system to eliminate one of the four conditions that lead to deadlocks.

For example, you can ensure that a process requests all required resources at once, preventing hold and wait1.

2 Deadlock Avoidance: Use algorithms to ensure the system will never enter an unsafe state. The Banker’s algorithm is a famous example of deadlock avoidance.

3 Deadlock Detection and Recovery: Allow deadlocks to occur, detect them, and recover from them. This involves periodically checking for deadlocks and taking action to resolve them, such as terminating a process or forcibly releasing resources.

4 Resource Allocation Graphs: Use graphs to represent resource allocation and detect cycles, which indicate deadlocks.