

LEC-22: Deadlock Part-2



1. **Deadlock Avoidance:** Idea is, the kernel be given in advance info concerning which resources will use in its lifetime.

By this, system can decide for each request whether the process should wait.

To decide whether the current request can be satisfied or delayed, the system must consider the resources currently available, resources currently allocated to each process in the system and the future requests and releases of each process.

- a. Schedule process and its resources allocation in such a way that the DL never occur.
- b. Safe state: A state is safe if the system can allocate resources to each process (up to its max.) in some order and still avoid DL.
A system is in safe state only if there exists a safe sequence.
- c. In an Unsafe state, the operating system cannot prevent processes from requesting resources in such a way that any deadlock occurs. It is not necessary that all unsafe states are deadlocks; an unsafe state may lead to a deadlock.
- d. The main key of the deadlock avoidance method is whenever the request is made for resources then the request must only be approved only in the case if the resulting state is a safe state.
- e. In a case, if the system is unable to fulfill the request of all processes, then the state of the system is called unsafe.
- f. Scheduling algorithm using which DL can be avoided by finding safe state. (Banker Algorithm)

2. Banker Algorithm

- a. When a process requests a set of resources, the system must determine whether allocating these resources will leave the system in a safe state. If yes, then the resources may be allocated to the process. If not, then the process must wait till other processes release enough resources.

3. **Deadlock Detection:** Systems haven't implemented deadlock-prevention or a deadlock avoidance technique, then they may employ DL detection then, recovery technique.

- a. Single Instance of Each resource type (**wait-for graph method**)
 - i. A deadlock exists in the system if and only if there is a cycle in the wait-for graph. In order to detect the deadlock, the system needs to maintain the wait-for graph and periodically system invokes an algorithm that searches for the cycle in the wait-for graph.
- b. Multiple instances for each resource type
 - i. Banker Algorithm

4. Recovery from Deadlock

- a. Process termination
 - i. Abort all DL processes
 - ii. Abort one process at a time until DL cycle is eliminated.
- b. Resource preemption
 - i. To eliminate DL, we successively preempt some resources from processes and give these resources to other processes until DL cycle is broken.