CHAPTER - 04 DEADLOCK

* Deadlock : It is a situation in which two or more processes are unable to proceed because each is waiting for the other to release resources.
A Process in an Operating System uses resources in following way:
1. Request 2. Use 3. Release 4. Wait

* Resource Allocation Graph It shows which resource is held by which process and which process is waiting for a resource of a specific kind.

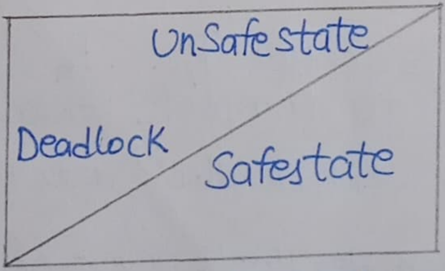
Nodes/Vertices → Process $\textcircled{P_1}$
→ Resources $\square R_1 \Rightarrow \boxed{\cdot\;\cdot\;\cdot} R_1$ (3 instance of resource R_1)

Edges → Allocation : from resource instance to process.
→ Request (wait) : from process resource.

* Necessary Conditions for Deadlock
Deadlock occurs when all the 4 condition is true.
1. Mutual Exclusion : A resource can never be used by more than one process simultaneously.
2. Hold and Wait : When a process holds a resource and waiting for some other resources.
3. No Preemption : A resource cannot be taken from a process unless the process releases the resource.
4. Circular Wait : A set of processes waiting for each other in circular form.

* Deadlock Prevention
To prevent from deadlock we have to violate one of the four necessary conditions to occur.

* Deadlock Avoidance
The request for any resource will be granted if the resulting state of the system doesn't cause deadlock in the system.



OS tries to keep system in safe state.
To check if system is in safe state or not & if a request of a process must be granted or not for this purpose it uses Banker's Algorithm.

Banker's Algorithm: It is a resource allocation and deadlock avoidance algorithm that tests for safety. It's named so because bankers also use similar algorithm.

1. Safety Algo: It checks whether system is in safe state or not.
2. Resource Request Algo: Every process must announced the max number of intances of each resources before the process starts execution.
    1. If request $\leq$ Need then go to step 2
       else
       invalid request.
    2. If request $\leq$ Available then go to step 3 process will wait.
    3. Available = Available − Request
       Allocation = Allocation + Request
       Need = Need − Request
    4. Run safety algo if safe then grant request else reject.

* Deadlock Detection: The OS doesn't apply any mechanism to avoid or prevent the deadlocks. Therefore system considers that the deadlock will definitely occur. In order to get rid of deadlocks, the OS periodically checks the system for any deadlock. In case, it finds any deadlock then the OS will recover the system using some recovery techniques.

Detection
- Single Instanced
    - Detect Cycle (Wait for Graph)
- Multiple Instanced
    - Safety Algorithm (BA)

Assume n number of processes $P_1 \ldots P_n$ each process P_i needs R_i number of resources to complete execution.

Min. no. of resources for which deadlock never occurs $= \sum_{i=1}^{n}(R_i - 1) + 1$

No. of resources for which deadlock never occurs $\geq \sum_{i=1}^{n}(R_i - 1) + 1$

Max no. of resources for which deadlock can occur $= \sum_{i=1}^{n}(R_i - 1)$