Deadlock

CHAPTER # 8

Deadlock : Process are stuck waiting forever for to release resources, leading to a complete halt

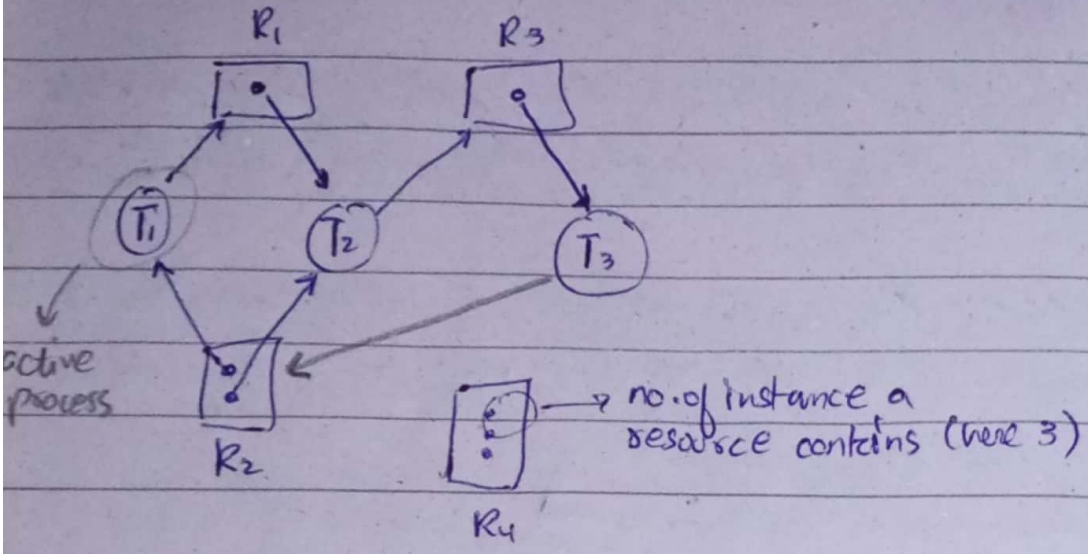
Live lock : Process are actively performing operation & changing state, but they are not progressing towards their intended outcomes

- In process in livelock are not blocked; they are continuously executing some action.
- No progress
-

Conditions for Deadlock

1. Mutual Exclusion
2. Hold & wait → A thread must be holding at least one resource and waiting to acquire additional resource currently held by another process.
3. No preemption → Resource cannot be preempted; a resource can only be released voluntarily by a thread holding it.
4. Circular wait → T0 waits for resource held by T1 & T1 ...

Resource Allocation Graph



$T_i \rightarrow R_j$ ⇒ mean T_i is waiting for Resource j.
$R_j \rightarrow T_i$ ⇒ Resource j is allocated to T_i

⇒ If there is not a cycle in the graph no deadlock occurs.

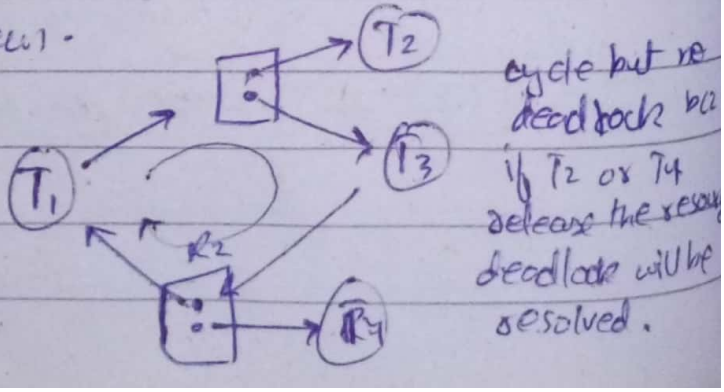
⇒ If there is a cycle then a deadlock may or may not occur.

↳ if we draw an edge $T_3 \rightarrow R_2$

↳ Now there is a cycle blw $T_1 \rightarrow R_1 \rightarrow T_2 \rightarrow R_3 \rightarrow T_3 \rightarrow R_2$. And another cycle of $T_2 \rightarrow R_3 \rightarrow R_2 \rightarrow T_2$

↳ Means a deadlock has occur.

↳ But for ex: if R_2 had 3 instances then a deadlock may not have occur.



→ Deadlock Prevention: provides a set of methods to ensure that at least one of the necessary conditions cannot hold. System create wagt kum ensure krein k deadlock na ho.

→ Deadlock Avoidance: requires that OS be given additional info in advance concerning which resource a thread will request. With this knowledge OS can decide for each request whether or not the thread should wait. We. dekhega k agr resource dene se deadlock aajyega to we resource allocate nhi krega