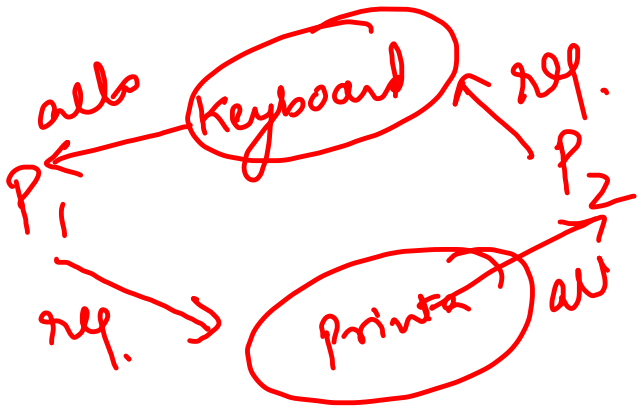
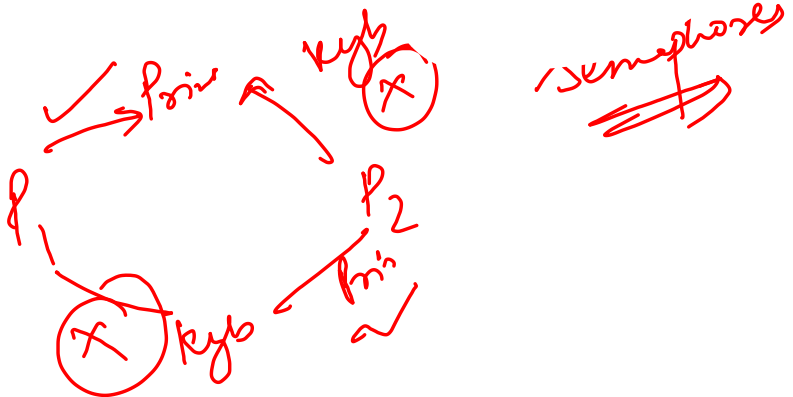
Deadlock

Condition in a system where a process cannot proceed because it needs to obtain a resource held by another process but it itself is holding a resource that the other process needs.

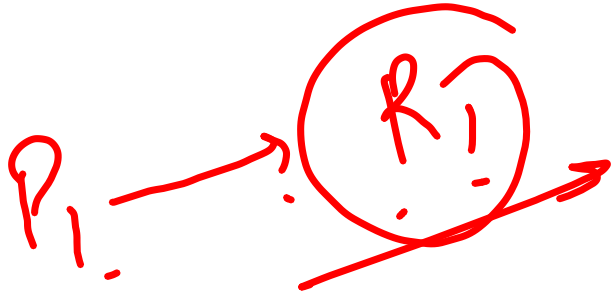




Semaphores

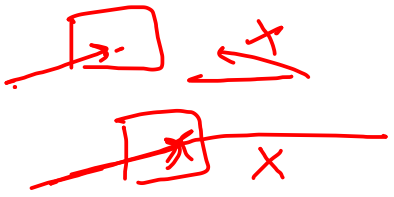
Resources are utilized as :-

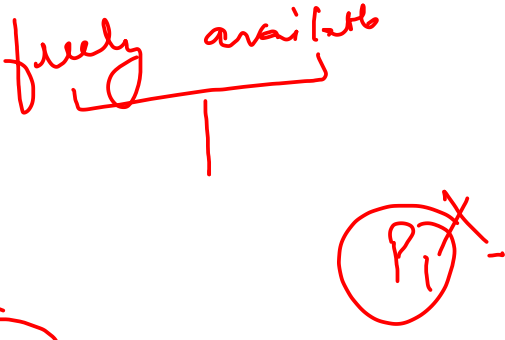
1) request → ✓
2) Use → ✓
3) Release ✓

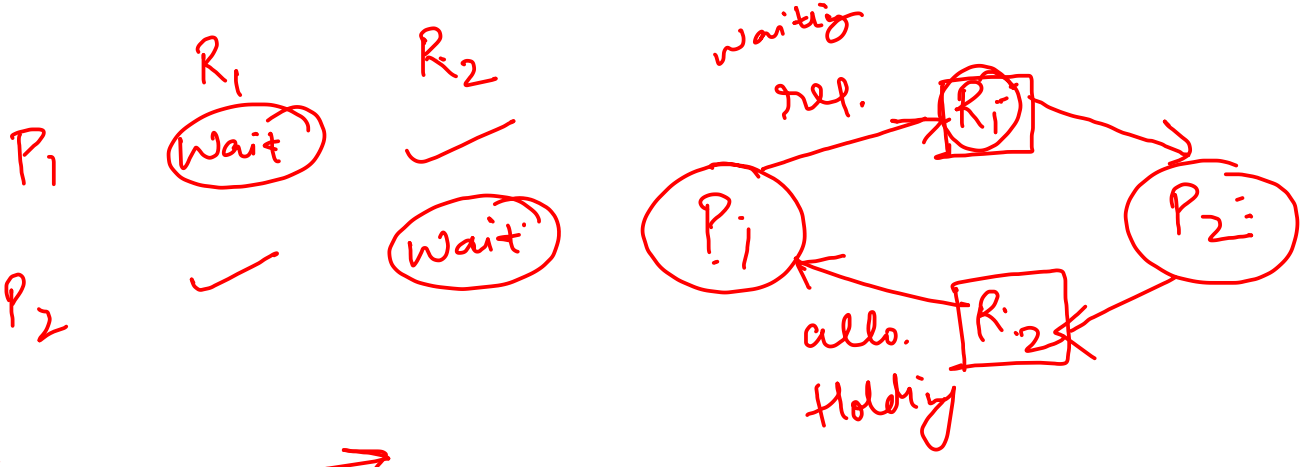


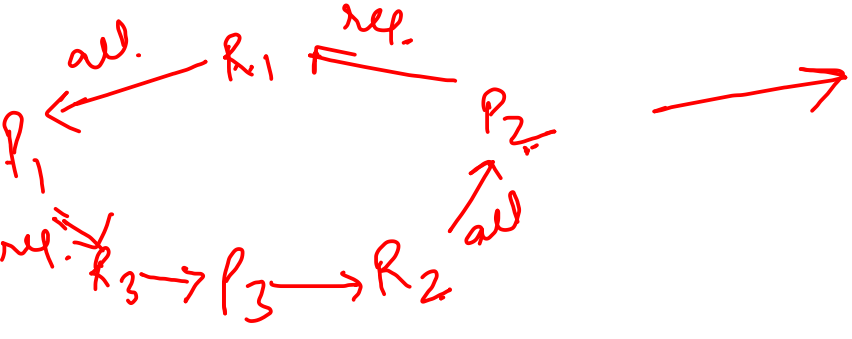
Conditions for deadlock

1. Mutual Exclusion - Resource can be allocated to one process or it is freely available
2. Hold & wait - Process is holding the resource & waiting ~~one~~ for some other resource simultaneously.
3. No pre-emption - Resource has to be voluntarily released by the process after execution.
4. Circular wait → circular waiting









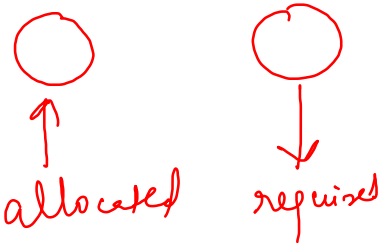
Resource allocation graph

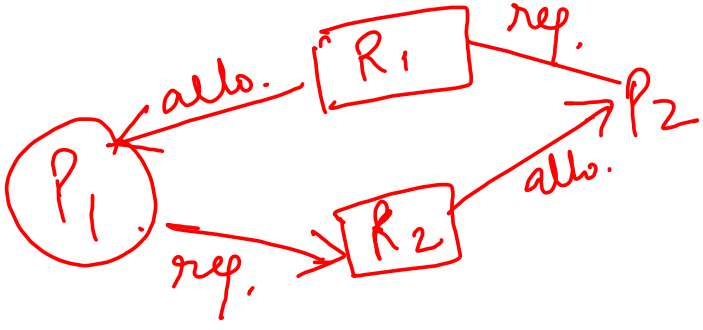
Vertex → process, resources

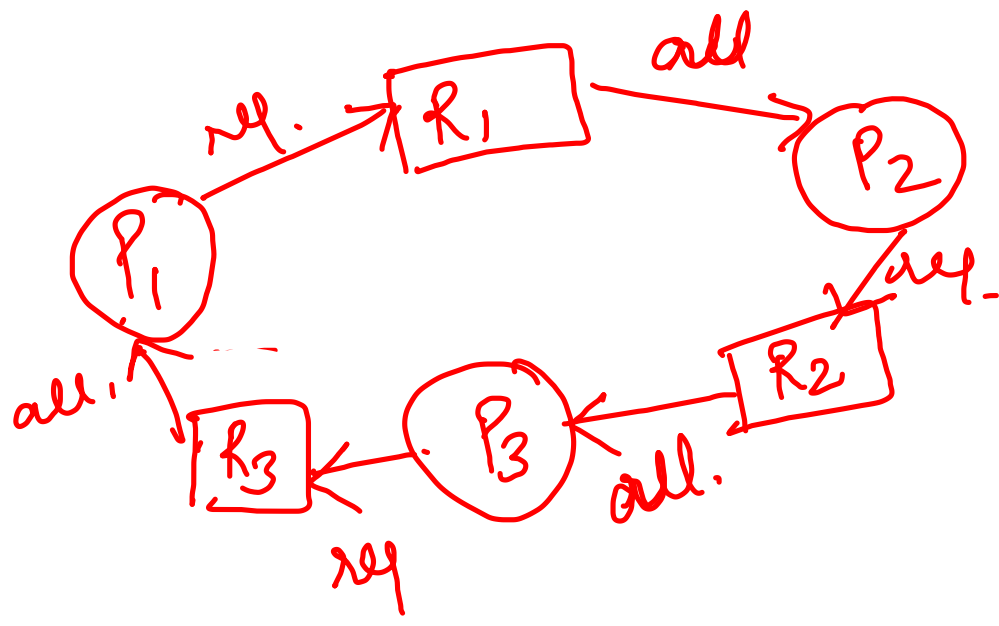
edges. → arrows

Process → Circle

Res → 







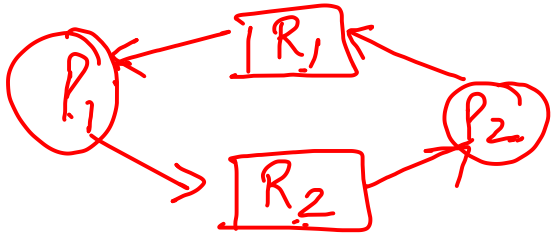
Deadlock

1) Deadlock ignorance
2) Deadlock prevention & avoidance
3) Detection & Recovery

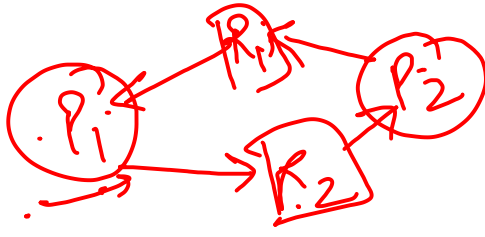
Deadlock ignorance

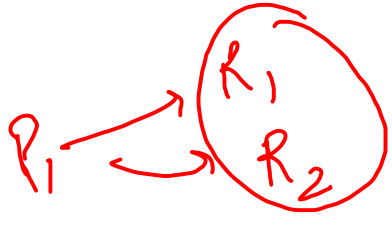
(Ostrich method)

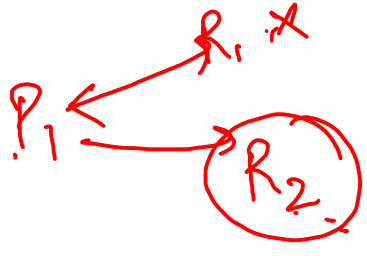
Deadlock Prevention



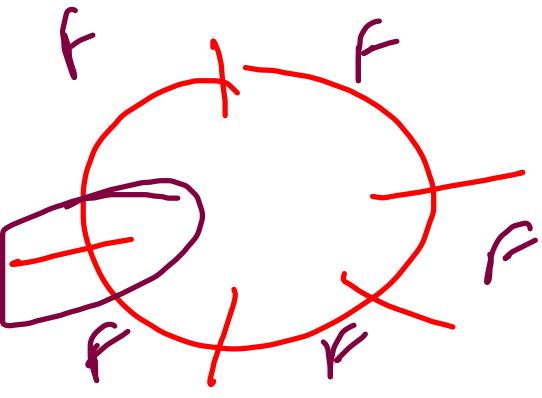
1. Mutual Exclusion - unshared resource → M.E.
2. No preemption - (Time quantum)
3. Hold & wait
4. Circular wait







Conditions



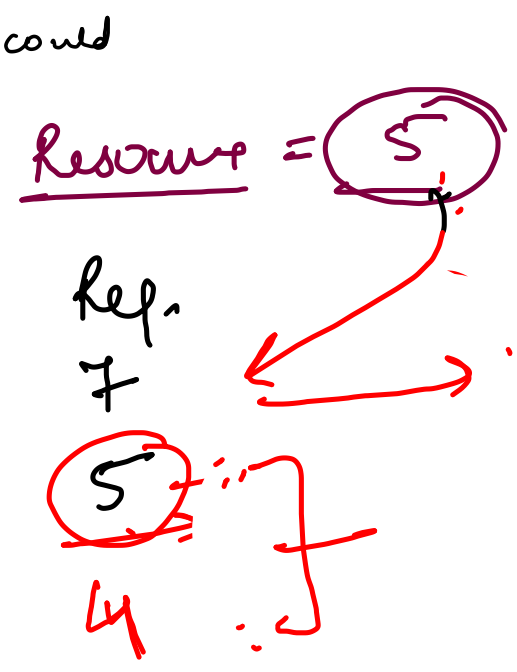
circular wait ⇒ Hold & wait

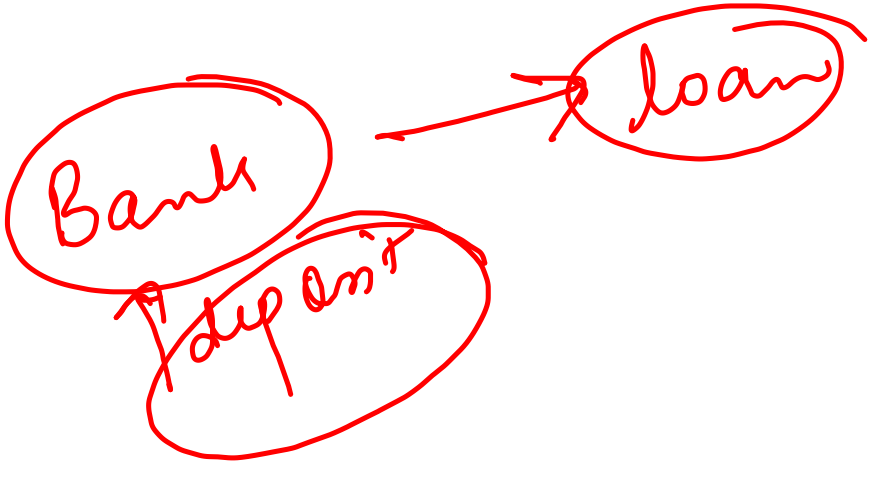
Deadlock avoidance

Never allocate resources in a way, that could lead to deadlock.

Bankers algo.

P	Need	Allo-
A	12	5
B	7	2
C	6	2

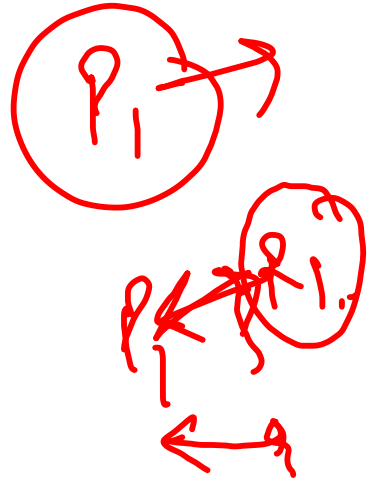




Detection & recovery

1) Detect → RAG

2) Kill all deadlock processes
   - → Process Termination
   - → Resource preemption
   - → Ostrich method



1) Ignorance (Ostrich method)
2) Prevention ⇒ 4.
3) Avoidance → Bankers algo.
4) Detection & Recovery → Process Term
→ Resou Preem
→ Rollback