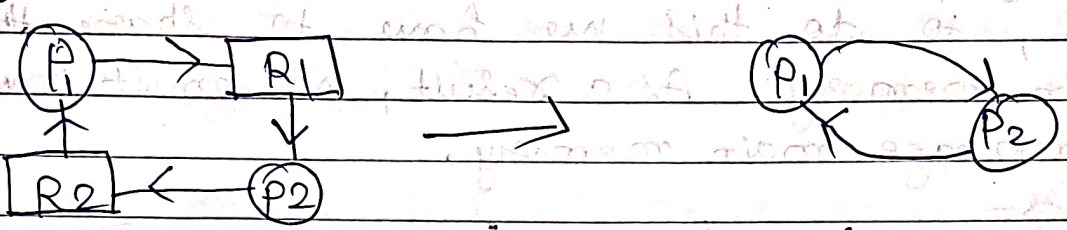
L:- 22

* Deadlock Part :- 2 (DL)

- **Deadlock avoidance** :- Idea is, the Kernel be given in advance info concerning which resources will use in lifetime.
- Schedule process and its resources in such a way that DL never occur.
- **Safe State** :- A State where System can allocate resources to each process (upto its max) in some order and still avoid DL.
- **Unsafe State** :- OS cannot prevent processes from requesting resources in such a way that a deadlock occurs.
- To find whether the System is in Safe State or not we use algorithm called "Banker Algorithm".
- **Banker Algorithm** :- When a process request a set of resources, the system must ensure that allocating these resources will leave the System in a safe state, if yes, allocate it, if not, then process must wait till other processes release enough resources.
  Same as how Bank manages its funds to avoid cashout issue.

- **Deadlock detection** :- System doesn't implemented any deadlock-prevention or avoidance technique, so now we detect the deadlock.

(a) **Single-Instance** :- Each resource is single instance, a deadlock exist in the system if and only if there is a cycle in the wait-for graph.



(RAG) (Wait For Graph)

(b) **Multiple-Instance for each resource** :- Use Banker algorithm.

- **Recovery from deadlock**

(a) **Process termination** :- (i) Abort all processes which are involved in deadlock.
(ii) Second way :- Abort one process at a time, until deadlock is eliminated.

(b) **Resource preemption** :- OS preempt resources from lower priority to higher priority jobs in order to eliminate Deadlock cycle.