

Deadlock: Resource Allocation Graph (RAG)



System Develop

Resources

Deadlock

can be :

- ✓ physical (eg. Printers)
- ✓ logical (eg. A record in a database)

a process may utilize a resource in only the following order :

- ❖ **Request** : if the request cannot be granted immediately, then the requesting process must wait until it can acquire the resource.
- ❖ **Use** : the process can operate on the resource
- ❖ **Release** : the process releases the resource

partitioned into several types.

$R_1, R_2, \dots, R_m$

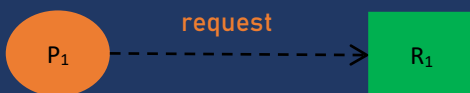
each consisting of some number of identical instances.

a state when every process in the set of processes is waiting for an event that can be caused only by another process in the set.

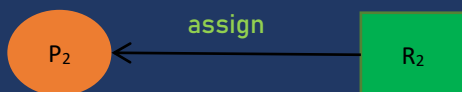


Deadlock Characterization

Types of Edges, E



Request edge:
directed edge $P_i \rightarrow R_j$



Assignment edge:
directed edge $R_j \rightarrow P_i$

can arise if four conditions hold simultaneously:

- ✓ **Mutual exclusion** : only one process at a time can use a resource
- ✓ **Hold and wait** : a process holding at least one resource is waiting to acquire additional resources held by other processes
- ✓ **No preemption** : a resource can be released only voluntarily by the process holding it, after that process has completed its task
- ✓ **Circular wait** : there exists a set $\{P_0, P_1, \dots, P_n\}$ of waiting processes such that P_0 is waiting for a resource that is held by P_1 , P_1 is waiting for a resource that is held by P_2 , ..., P_{n-1} is waiting for a resource that is held by P_n , and P_n is waiting for a resource that is held by P_0 .