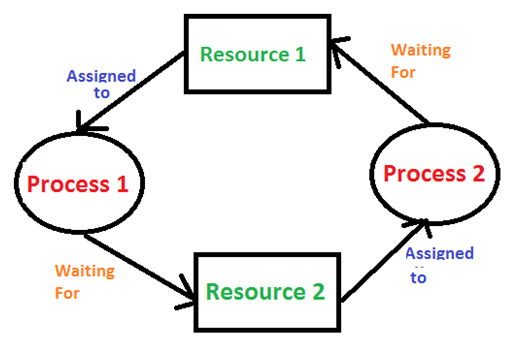
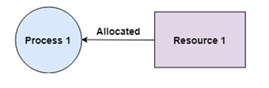
Deadlock

* ***Deadlock***is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process.
* For example, in the below diagram, Process 1 is holding Resource 1 and waiting for resource 2 which is acquired by process 2, and process 2 is waiting for resource 1.
* A process in operating systems uses different resources and uses resources in the following way.   
  1) Requests a resource   
  2) Use the resource   
  2) Releases the resource

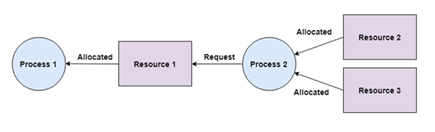


Deadlock Characterization

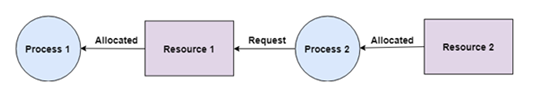
* In a deadlock, processes never finish executing and system resources are tied up preventing other jobs from starting.
* A deadlock occurs if the four Coffman conditions hold true. But these conditions are not mutually exclusive. They are given as follows −
* Mutual exclusion
* Hold and wait
* No pre-emption
* Circular Wait
* **Mutual Exclusion :** A resource can only be shared in mutually exclusive manner. It implies, if two process cannot use the same resource at the same time.



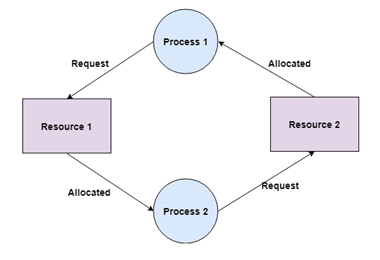
**2. Hold and Wait:** A process waits for some resources while holding another resource at the same time. In the diagram given below, Process 2 holds Resource 2 and Resource 3 and is requesting the Resource 1 which is held by Process 1.



**3. No pre-emption:** The process which once scheduled will be executed till the completion. No other process can be scheduled by the scheduler meanwhile. In the diagram below, Process 2 cannot pre-empt Resource 1 from Process 1. It will only be released when Process 1 relinquishes it voluntarily after its execution is complete.



**4. Circular Wait:** All the processes must be waiting for the resources in a cyclic manner so that the last process is waiting for the resource which is being held by the first process. For example: Process 1 is allocated Resource2 and it is requesting Resource 1. Similarly, Process 2 is allocated Resource 1 and it is requesting Resource 2. This forms a circular wait loop.



Resource Allocation Graph

* Deadlocks can be described in terms of a directed graph called a system resource allocation graph.
* This graph consist of set of vertices and set of edges E.
* A set of resource categories, { R1, R2, R3, . . ., RN }, which appear as square nodes on the graph. Dots inside the resource nodes indicate specific instances of the resource.
* A set of processes, { P1, P2, P3, . . ., PN }
* **Request Edges -**A set of directed arcs from Pi to Rj, indicating that process Pi has requested Rj, and is currently waiting for that resource to become available.
* **Assignment Edges -**A set of directed arcs from Rj to Pi indicating that resource Rj has been allocated to process Pi, and that Pi is currently holding resource Rj.