* Deadlock:

A deadlock can be defined as the permanent blocking of a set of processes that either compete for system resources, or communicate with each other. Or deadlock involves conflicting needs for resources by two or more processes.

eg: In a multiprogramming system, suppose two processes are there & each want to print large tape files. Process A request permission to use the printer & is granted. Process B then requests permission to use the tape drive & is also granted. Now A asks for the tape drive but the requests are denied until B release it. Instead of realeasing the tape drive, B asks for the printer. At this point both processes are blocked & will remain so forever. This situation is deadlock.

* Conditions of Deadlock:

a) Mutual exclusion:

The resources which require only mutually exclusive access, may give rise to a deadlock. These resource types cannot allow multiple processes to access it at the same time. For eg, a memory location, if allocated to a process, cannot be allowed to some other process. A printer cannot print the output of two processes at the same time.

b) Hold and wait:

There must exist a process that is holding at least one resource & is waiting to acquire additional resources that are currently being held by another process. Process currently holding resources granted earlier can request new resources.

c) No preemption:

Resources cannot be preempted, that is a resource can only be released voluntarily by the process holding it, after the process has completed its task. Resources previously granted cannot be forcibly taken away from a process. They must be explicitly released by the process holding them.

d) Circular Wait:

There exist a set (P_0 - P_n) of waiting processes such that P_0 is waiting for a resource which is held by P_1, P_1 is waiting for a resource which is held by P_2, P_{n-1} is waiting for a resource which is held by P_n, & P_n is waiting for a resource which is held by P_0. Thus there must be a circular chain of two or more processes, each of which is waiting for a resource held by the next member of the chain.

→ Resource Allocation Graph (RAG):

Deadlock can be described more precisely in terms of a directed graph called system resource allocation graph. The RAG follows the following notations & rules.

(P) → Process

□ R → Resource

[• •] → Number of instances of that resource

(P) → [• •] R → process has requested this resource but it has not been allocated. Known as request edge.

[• •] R → (P) → one instance of this resource has been allocated to the process. Known as assignment node.