Deadlock in Operating System

Deadlock is the situation in which two processes are each waiting for the other in order to complete the present process before going to the other one. The result is that both the processes are hanging. The situation of Deadlock generally occurred in multitasking and client-server environment. A deadlock is also called a deadly embrace. This term is most commonly used in the Country Europe. Here lecture notes in computer science on Deadlocks are given including their necessary conditions, Deadlock handling, prevention, and avoidance.

Deadlock in Operating System:

In an operating system, a deadlock is the occurs when a process enters into a waiting state because a resource request is being made by the other waiting process, which in turn become a waiting status for the other resource. When the process is unable to change its state indefinitely because the resources requested by it are being used by another waiting process, then the system in a deadlock state.

So this was a general description about deadlock, technically we can define deadlock as a state where certain process rather we can say the inter related process wait for the resources which is used by the other process so the state where every process waits for the resources is called as the deadlock.

So there are certain conditions due to which in a system deadlock occur:

Mutual Exclusion:

In this condition, at least two resources are must be non-sharable and only one process can use the resource any given instant of time.

Hold and Wait:

This is also known as Resource holding, a process in which at least one resource is in holding state and can request more resources which are being held by other processes.

No preemption:

In this condition the other process is not able to request the resources until the previous process is not complete. These resources must be released by the holding process voluntarily.

Circular Wait:

This is the last but not the least condition in which one process waits for the resource held other process which is waiting for the nest process to release the resources.

These are the four necessary conditions for Deadlock which is named as Coffman conditions. These conditions are enough to preclude a deadlock from occurring.

Deadlock handling

Most of the operating systems are not able to prevent the deadlocks to occur. When a condition of deadlock occurs then different operating systems respond to them in different and non-standard way. Most approaches work by preventing one of the four Coffman conditions from occurring, especially for the Current Wait. Major approaches are

Ignoring Deadlock

This approach assumed that a deadlock will never occur which is one of the most important application of Ostrich algorithm. The Ignoring deadlock is commonly used by MINIX and UNIX. When the time interval between the occurrences of deadlocks are seems to be large in addition to the increment in data loss incurred each time is tolerable then in such case Ignoring deadlock is used.

Detection

In this approach, deadlocks are allowed to occur. Then the state of the system is examined to detect that a deadlock has occurred and subsequently it is corrected. An algorithm is employed that tracks resource allocation and process states, it rolls back and restarts one or more of the processes in order to remove the detected deadlock.

Avoidance

Deadlock can be avoided if some of the information about the processes is well known Operating System before the allocation of resources starts. For every resource request, the system sees whether granting the request will mean that the system will enter an unsafe state, meaning a state that could result in deadlock. The system then only grants requests that will lead to safe states.