Deadlock Prevention | Operating System – M04 P04

This is a multipart blog article series, and in this series I am going to explain you the concepts of operating system. This article series is divided into multiple modules and this is the forth module which consists of 8 articles.

In this article we are going to see that what are the ways by which we can prevent deadlock and various methods to handle deadlock in an operating system.

**Various methods to handle deadlock.**

1. Deadlock ignorance (ostrich method)
2. Deadlock prevention.
3. Deadlock avoidance (banker’s algorithm)
4. Deadlock detection and recovery.

**Deadlock ignorance:**

* In this method when deadlock occur in the system then operating system just ignore the deadlock.
* This method is used because, if we add one more algorithm/ patch of code, in our operating system then the performances of the system get decreased.
* And also that the deadlock condition occur very rarely.
* Restart the system that is deadlock ignorance.
* This is called ostrich method, because when sand storm comes ostrich put its head in ground, and think that there is no sand storm.

**Deadlock prevention:**

* This method means that we try to find a way that we do not get any deadlock scenario, before the deadlock occurs.
* It can be followed by removing the following conditions (any one)
  + Mutual exclusion.
  + No preemption
  + Hold and wait
  + Circular wait

**Deadlock avoidance**

* In this method when we give resource to a process than we check that if it is safe or not to give that resource to that process.
* This is done by “Banker’s algorithm”

**Deadlock detection and recovery**

* In this method we try to detect that is there any deadlock present or not in the system. If the deadlock is present than we try to recover our system form it.
* We can do recovery by
  + Kill the processes or process (But this will decrease the performance of the system)
  + Resource preemption from a process.

So this was all about deadlock handling methods and deadlock prevention. Hope you liked it and learned something new from it.

If you have any doubt, question, quires related to this topic or just want to share something with me, than please feel free to contact me.