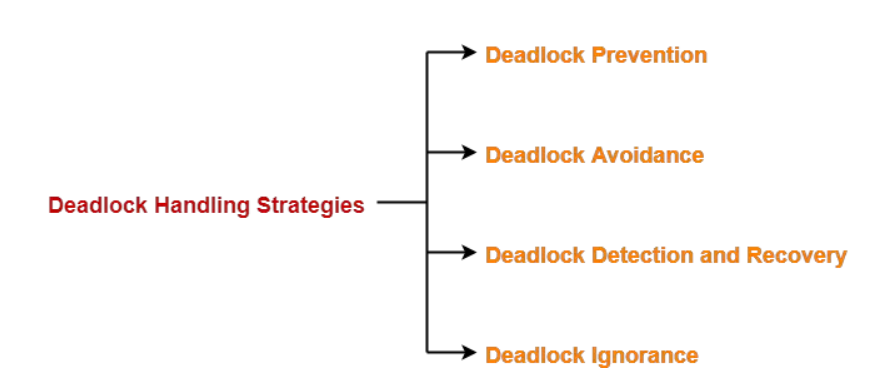
**METHODS FOR HANDLING DEADLOCKS**

There are mainly four methods for handling deadlock.

1. Deadlock Prevention
2. Deadlock Avoidance
3. Deadlock Detection and Recovery
4. Deadlock Ignorance

****

**1)Deadlock Prevention:**

In this method, the system will prevent any deadlock condition to happen i.e. the system will make sure that at least one of the four conditions of the deadlock will be violated. Since we are preventing any one of four conditions to happen by applying some techniques.

**2)Deadlock Avoidance:**

In the deadlock avoidance technique, we try to avoid deadlock to happen in our system. Here, the system wants to be in a safe state always. So, the system maintains a set of data and using that data it decides whether a new request should be entertained or not. If the system is going into the bad state by taking that new request, then the system will avoid those kinds of request and will ignore the request. So, if a request is made for a resource, from a system, then that request should only be approved if the resulting state of the system is safe i.e. not going into deadlock.

**3)Deadlock detection and Recovery:**

If the Operating System does not employ either Deadlock Prevention or Deadlock avoidance then we can assume that deadlock will occur. In those cases, It is easily possible to detect deadlock because the resource scheduler in the Operating System has all the information about which process has locked which resources and requested which resources.

After detecting deadlock exists, there are several options to break the deadlock and recover the system using 2 approaches.

1)Process termination

2)Resource pre-emption.

**4)Deadlock Ignorance (Ostrich Method):**

This is the most widely used methods of deadlock handling which is used in most operating systems like Windows and Linux. This approach is best suitable for a single end user system where User uses the system only for browsing and all other normal stuff. Since in many systems, deadlocks occur infrequently (say, once per year), the extra expense of the other methods may not seem worthwhile. Ignoring the possibility of deadlocks is cheaper than the other approaches. so, the Operating System assumes that the deadlock is never going to happen. It simply ignores the deadlock. Performance of the system decreases if the deadlock is handled always. If in the worst-case deadlock occurs, it may be necessary to shut down the system or manually kill some process.