Ques> What is deadlock? How to deadlock detect and recover?

ANS:-A process in operating system uses different resources and uses resources in following way.

1. Requests and resource.

2. Use the resource.

3. Release the resource.

Deadlock is a situation where a set of process are blocked because each process is holding a resource and waiting for another.

Consider an example when two trains are coming toward each other on same track and there is only one track, none of the trains can move once they are in front of each other. Similar situation occurs in operating systems when there are two or more process hold some resources and wait for resources held by others. 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.

Deadlock can arise if following four conditions hold simultaneously (necessary conditions) mutual exclusion. One or more than one resource are non-sharable (only one process can use at a time) Hold and wait: A process is holding at least one resource and waiting for resources. NO pre-emption: A resource cannot be taken from a process unless the process release the resource.

Circular wait: A set of processes are waiting for each other in circular form.

METHODS FOR HANDLING DEADLOCK:-

There are three ways to handle deadlock:-

1. Deadlock prevention or avoidance: The idea is not be the system into deadlock state.
2. Deadlock detection and recover: Let deadlock occur, then do pre-emption to handle it happen and reboot the system.
3. Ignore the problem all together: If deadlock is very rare, then let it happen and reboot the system. This is the approach that both windows and UNIX take.

How to deadlock detect and recover:-

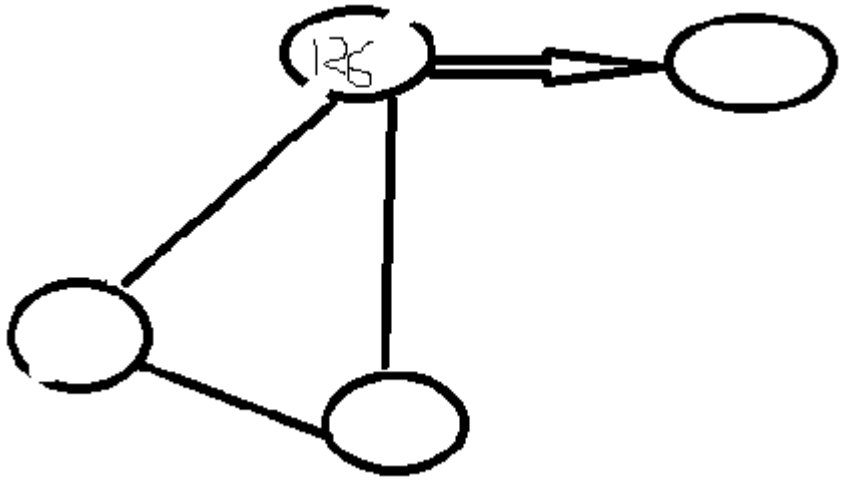
If is a system does not employ some protocol that ensures deadlock freedom, that a detection and recovery scheme must be used. An algorithm that examines the state of the system is invoked periodically to determine whether a deadlock has occurred. If one has, then the system must attempt to recover from the deadlock. To do so, The system must.

a. Maintain information about the current allocation of data item to transactions.
b. Provide and algorithm that to determine whether the system has entered a deadlock state.
c. Recover from the deadlock when the detection algorithm determines that a deadlock exists.

Deadlock Detection:-

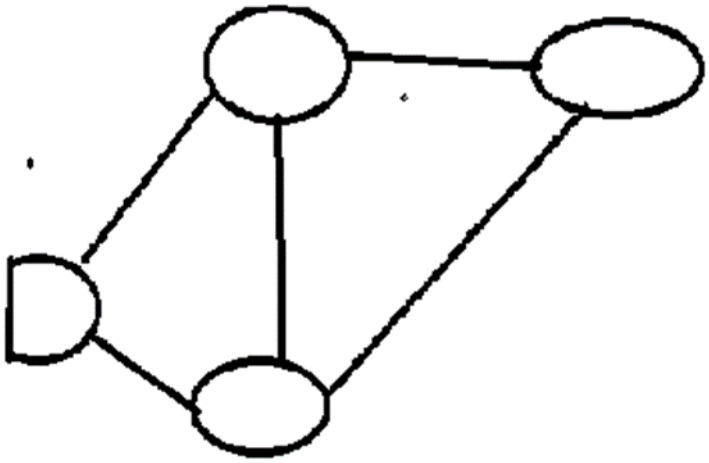
A simple way to detect a state of deadlock is with the help of wait for graph. This graph is constructed and maintained by the system. One node is created in the wait for graph for each transaction that is currently executing. Whenever a transaction TI is waiting to lock an item X that is currently locked by a transaction TJ, a directed edge (TI->TJ) is created in the wait for graph. When TJ releases the lock(s) on the items that TI was waiting for, the directed edge is dropped from the wait for graph. We have a state of deadlock if find only if the wait for graph has a cycle. Then each transaction involved in the cycle is said to be deadlocked. To detect deadlocks, the system needs to maintain the wait for graph, and periodically to invoke an algorithm that searches for a cycle in the graph.

To illustrate these concepts, consider the following wait for graph in figure. Here:
Transaction T_{25} **is** waiting for transaction T_{26} and T_{27}.
Transaction T27 is waiting for transaction T26.
Transaction T26 is waiting for transaction T28.



REPRESTING NO DEADLOCK STATE

This wait for graph contains the cycle, so there is no deadlock state.
Suppose now that transaction T28 is requesting an item held by t27.
Then the edge T28 ----------→T27 is added to the wait for graph, resulting in a new system state as shown in figure.



REPRESENTING A DEADLOCK SATE

This time the graph contains the cycle.

T26----→T28-→T27-→----T26

It means that transaction T26, T27 and T28 are all deadlock.

Invoking the deadlock detection algorithm

The invoking of deadlock detection algorithm depends on two factors:

A. How often does a deadlock occur?
B. How many transactions will be affected by the deadlock?

If deadlocks occur frequently, then the detection algorithm should be invoked more frequently than usual. Data items allocated to deadlock transaction will be unavailable to other transactions until the deadlock can be broken. In the worst case, we would invoked the detection algorithm every time a request for allocation could not be granted immediately.

RECOVERING FROM DEADLOCK:-

When a detection algorithm determines that a deadlock exists, the system recover from the deadlock. The most common solution is to roll back one or more transaction to break the deadlock .Choosing which transaction to abort is known as victim selection.

ROLLBACK:-

Once we have decided that a particular transaction must be rolled back, we must determine how far this transaction should be roll back. The simplest solutions is total rollback; Abort the transaction and then restart it. However it is more effective to roll back the transaction only as far necessary to break the deadlock.