Deadlocks Part-1

Content :-

DeadLock :- It is a condition , wherein a set of processes are waiting forever for the resources held by each other. None of them is able to proceed with its execution.

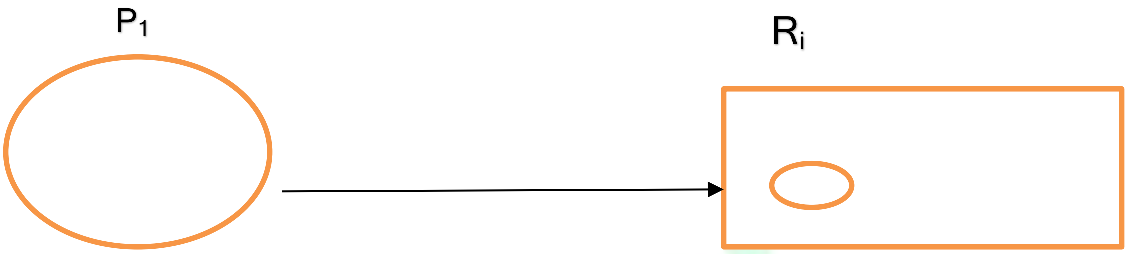
Necessary Conditions For DeadLock To Occur :-

1. **Mutual exclusion** :- Some processes must hold some resources in a non shareable mode or mutually exclusive mode.
2. **Hold and wait :-** Some processes must be holding some resources in a non-shareable mode and at at the same time must be waiting to acquire some more resources, which are currently held by the other processes in a non-shareable mode .
3. **No Preemption :-** the resources held by processes cannot be preemted forcibly. A process releases the resource voluntarily when finished with it.
4. **Cycle Wait** :- A set of process, say [P_0,P_1,P_2,P_3……P_{N-1}] must wait in a cycle fashion for resources held by each other .

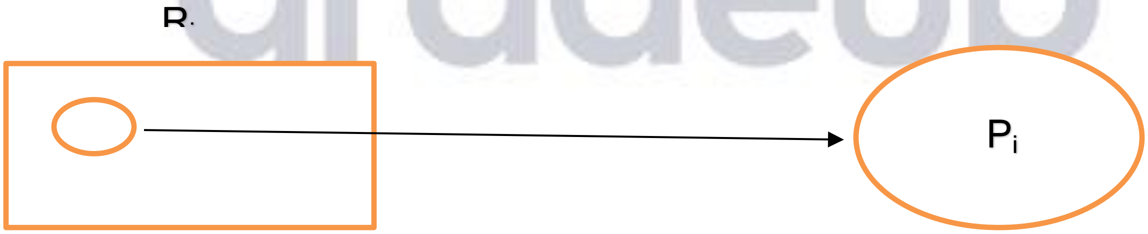
Note : THIS ARE NECCERAY CONDITION BUT NOT SUFFICIENT.

Resource allocation graph :

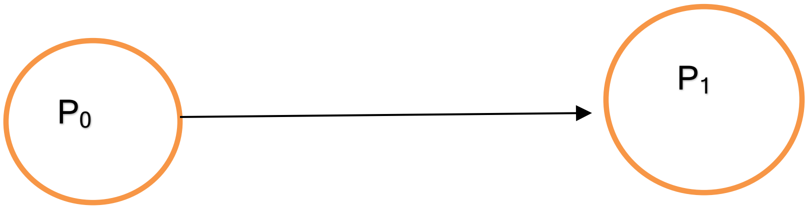
An edge of process P_1 to resource R_1 represent a request edge .



An edge from resource R_1 to process P_1 represent an assignment edge.



Process Wait for Graph (PWFG):- An edge from P_0 to P_1 indicates that process P_0 is waiting for a resource, currently held by P_1.



Methods of Deadlocking Handling :-

Deadlock can be handle any of the following methods:

1. **Deadlock Prevention** :- Deadlock can be prevented from occurring by preventing one of the necessary four conditions (i.e. mutual exclusion, hold and wait, no preemtion and cycle wait) from holding. If one of the conditions can be prevented from holding then deadlock will not occur.
2. **Deadlock avoidance** :- Deadlock can be avoided by maintaining the system always in safe state. The system is said to be in safe state , if all the pending processes can be successfully executed in some sequence, while satisfying their resource requirement fully. The sequence in which pending process can be executes is called as safe sequence.
3. **Deadlock detection and recovery** :- The above approaches consider overheads. A less costly approach , when system requirement is not very critical it can be :
    a. Allow deadlock to occur
    b. Detect deadlock , when it occurs
    c. Recovery from the deadlock

4. **Allow deadlock to occur :-** Restart when the system stops functioning. This is least costly approach, while cab be employed when application is not at all critical. When deadlock occurs, system will finally stop functioning. Just reboot the system.

https://bit.ly/2ZamfsH

Score Better.
Get Started with Gradeup!
Sign in with Google
Continue
Select your exam
Upcoming
Banking
SSC
Teaching
JEE
GATE
Medical
UPSC
Defence
NET
SBI Clerk
JEE Main 2020
CTET
RRB NTPC
NEET 2020
NIELIT
SSC CPO
CLAT
UPPSC PCS
SSC CGL
KVS
CDS
SSC CHSL
UPTET
SEBI Grade A