**Operating Systems Design** :

Tutorial 05 - Deadlocks

***Question 1.*** *Describe how deadlocks can be prevented.*

> A deadlock is obtained when one of the these conditions is verified : No pre-emption, mutual exclusion, resource holding and circular wait. In order to prevent a deadlock, we must eliminate one of these four conditions : The no pre-emption could be bypassed by allowing Operating System to deallocate resources from the jobs ; Mutual exclusion is necessary and exceptions include devices where mutual exclusion can be bypassed by spooling ; Resource holding can be avoided by forcing each job to request at the creation time every resource it will need ; Finally, circular wait can be bypassed if the Operating System prevents the formation of a circle (using scheme of hierarchical ordering).

***Question 2.*** *Suppose the current system resource allocation is described by the following table. A request is received by job 1 for 2 resources. Determine whether or not the request should be served in order to avoid a deadlock. Explain how the decision was reached.*

*Une image contenant capture d’écran

Description générée automatiquement*

> In view of the elements in this following table, 7 devices are currently allocated out of a total of 10 and 3 units are still available. For job 1, the request can be served because job 1 requires 2 to complete its execution. For job 2 and job 3, 3 units are available but job 2 requires 5 to complete its execution and job 3 requires 6 to complete its execution, so request can’t be served and we have an unsafe case.

***Question 3.*** *In order to recover from a deadlock it may be necessary to select a victim process to terminate. Explain factors to be considered to select such a process.*

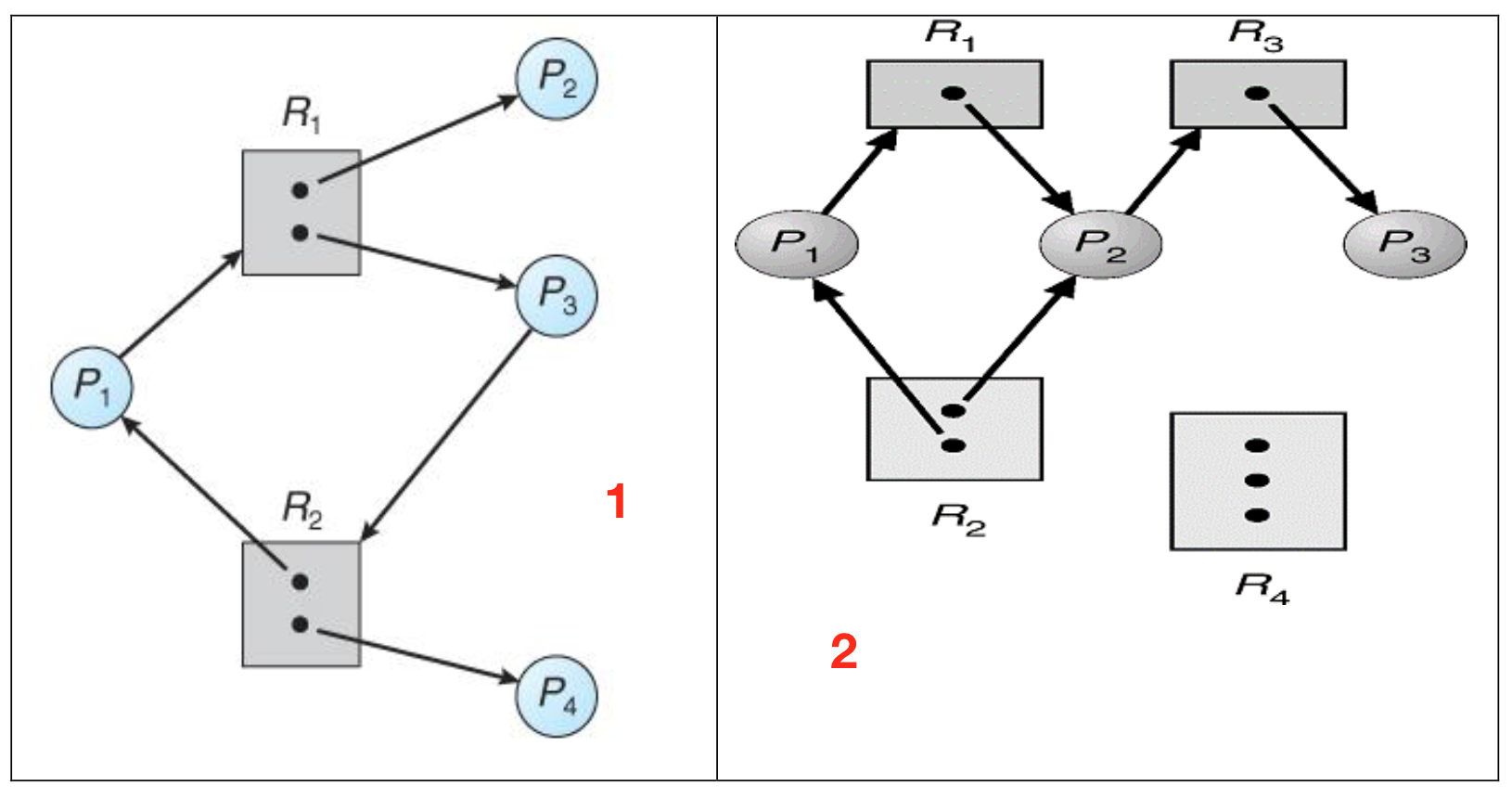
> The recovery process requires a “victim”, which must have the least negative effects on the entire system. An "ideal" victim would be a victim where high-priority jobs are usually untouched (priority of job under consideration), jobs close to completion are usually left alone (CPU time used by a job). The number of others jobs that would be affected if this job were selected as the “victim” and jobs that are modifying data shouldn’t be selected for termination, are also factors.

***Question 4.*** *Write the conditions required for a deadlock to occur and explain how to eliminate two of them from a system.*

> There are currently four conditions for an impasse to occur : The no pre-emption is a lack of temporary reallocation of resources ; The mutual exclusion is an act of allowing only one process to have access to a dedicated resource ; The resource holding is an act of holding a resource and not releasing (“hold & wait”) and waiting for the other job to retreat ; Finally, the circular wait is when each process involved in impasse is waiting for another to voluntarily release the resource so that at least one will be able to continue. If we want to eliminate, for example the circular wait, we have to make sure that the Operating System prevents the formation of a circle (using scheme of hierarchical ordering). For no pre-emption, we can allow the Operating System to deallocate resources from jobs.

***Question 5.*** *Explain the deadlock detection algorithm using directed resource graphs. Implement the algorithm on each of the two directed resource graphs below and determine the deadlocked processes if any.*

> The deadlock detection algorithm can detect a deadlock by building directed resource graphs and looking for cycles, in 3 steps : First, we must remove a process that is currently using a resource and not waiting for one. Secondly, we must remove a process that is waiting only for resource classes that aren’t fully allocated. Finally, go back to the first step and continue with first and secondly steps until all connecting lines have been removed.



1. P2 doesn't wait for a resource, P2 is removed.  
   P4 doesn’t wait for a resource, P4 is removed.  
   R1 isn’t fully allocated, P1 is removed.

P3 is removed.  
There’s no deadlock.

1. P3 doesn’t wait for a resource, P3 is removed.  
   R3 is free, P2 is removed.  
   Then, P1 can be removed.  
   There’s no deadlock.