Operating Systems
Jacobs University Bremen
Dr. Jürgen Schönwälder

Course: 320202
Date: 2012-03-14
Time: 10 min.

Quiz Sheet #3

Problem 3.1: *necessary deadlock conditions* (1+1+1+1 = 4 points)

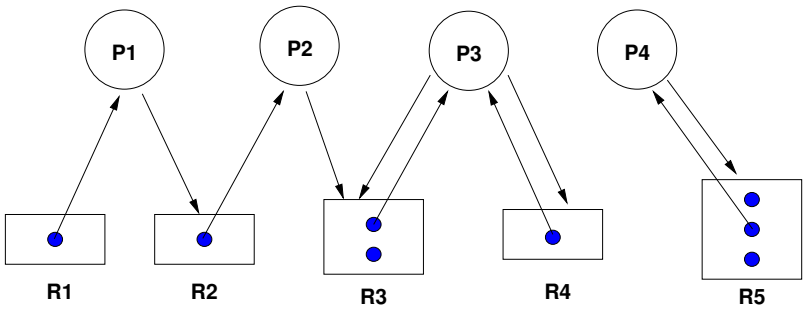
Name the four necessary conditions for deadlocks and provide a one sentence definition for each.

Solution:

1. *Mutual exclusion*:
   Resources cannot be used simultaneously by several processes
2. *Hold and wait*:
   Processes apply for a resource while holding another resource
3. *No preemption*:
   Resources cannot be preempted, only the process itself can release resources
4. *Circular wait*:
   A circular list of processes exists where every process waits for the release of a resource held by the next process

Problem 3.2: *resource allocation graphs* (1+1+1+1 = 4 points)

A system is in the state given by the following resource allocation graph. For each process, explain whether it is deadlocked or whether it can get deadlocked.



Solution:

Process P4 holds an instance of resource type R5 and requests a second instance of resource type R5. Since there are still instances of resource types R5 available, P4 is not deadlocked.

Process P3 holds an instance of resource type R3 and an instance of resource type R4. P3 at the same time requests a second instance of resource type R3 and a second instance of resource type R4. Since the system only has one instance of resource type R4, process P3 deadlocks itself.

Process P1 waits for the only instance of resource type R2, which is currently assigned to P2. P2 waits for an instance of resource type R3. Since there is still one instance of resource type R3 available, the system can assign it to P2 and thus P2 can continue and eventually P1 can continue. However, since P3 also requests the only available instance of resource type R3, P2 and P1 might also get deadlocked if the instance is assigned to P3 since P3 deadlocks on itself (see above).

Problem 3.3: *banker's algorithm* (2 points)

A system has $n = 4$ processes, $m = 2$ resource types, and a total number of resources $Total = (5, 7)$. The system is in the following state:

$$Max = \begin{pmatrix} 3 & 2 \\ 2 & 5 \\ 1 & 7 \\ 5 & 0 \end{pmatrix} \qquad Alloc = \begin{pmatrix} 0 & 2 \\ 1 & 2 \\ 0 & 0 \\ 2 & 0 \end{pmatrix}$$

Is this state safe?

Solution:

$$Need = Max - Alloc = \begin{pmatrix} 3 & 0 \\ 1 & 3 \\ 1 & 7 \\ 3 & 0 \end{pmatrix}$$

$$Avail = (2, 3) \overset{P2}{\rightarrow} (3, 5) \overset{P1}{\rightarrow} (3, 7) \overset{P3}{\rightarrow} (3, 7) \overset{P4}{\rightarrow} (5, 7)$$

The system is in a safe state since there is a resource assignment sequence in which all processes can finish while receiving their maximum number of resource requests.