

Operating Systems

Homework #5
Due: 2004/5/26

1. Is it possible to have a deadlock involving only one process? Explain your answer.
2. Consider a system consisting of four resources of the same type that are shared by three processes, each of which needs at most two resources. Show that the system is deadlock-free.
3. Consider the following snapshot of a system:

	<i>Allocation</i>				<i>Max</i>				<i>Available</i>			
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
P_0	0	0	1	2	0	0	1	2	1	5	2	0
P_1	1	0	0	0	1	7	5	0				
P_2	1	3	5	4	2	3	5	6				
P_3	0	6	3	2	0	6	5	2				
P_4	0	0	1	4	0	6	5	6				

Answer the following questions using the banker's algorithm:

- a. What is the content of the matrix *Need*?
 - b. Is the system in a safe state?
 - c. If a request from process P_1 arrives for $(0,4,2,0)$, can the request be granted immediately?
4. Consider the following resource-allocation policy. Requests and releases for resources are allowed at any time. If a request for resources cannot be satisfied because the resources are not available, then we check any processes that are blocked, waiting for resources. If they have the desired resources, then these resources are taken away from them and are given to the requesting process. The vector of resources for which the waiting process is waiting is increased to include the resources that were taken away.

For example, consider a system with three resource types and the vector *Available* initialized to $(4,2,2)$. If process P_0 asks for $(2,2,1)$, it gets them. If P_1 asks for $(1,0,1)$, it gets them. Then, if process P_0 asks for $(0,0,1)$, it is blocked (resource not available). If P_2 now asks for $(2,0,0)$, it gets the available one $(1,0,0)$ and one that was allocated to P_0 (since P_0 is blocked). P_0 's *Allocation* vector goes down to $(1,2,1)$, and its *Need* vector goes up to $(1,0,1)$.

- a. Can deadlock occur? If so, give an example. If not, which necessary condition cannot occur?
- b. Can indefinite blocking occur?