CSI3131 – Operating Systems
Tutorial 6 – Deadlocks

1. Is it possible to have a deadlock involving only one single process? Justify your answer.

2. Consider a computer system that runs 5,000 jobs per month with no deadlock-prevention or deadlock-avoidance scheme. Deadlocks occur about twice per month, and the operator must terminate and rerun about 10 jobs per deadlock. Each job is worth about $2 (in CPU time), and the jobs terminated tend to be about half-done when they are aborted.

   A systems programmer has estimated that a deadlock-avoidance algorithm (like the banker's algorithm) could be installed in the system with an increase in the average execution time per job of about 10 percent. Since the machine currently has 30-percent idle time, all 5,000 jobs per month could still be run, although turnaround time would increase by about 20 percent on average.
   - What are the arguments for installing the deadlock-avoidance algorithm?
   - What are the arguments against installing the deadlock-avoidance algorithm?

3. Recall the following example in class:
   - 5 processes are running in the system; *P*0 through *P*4.
   - They are using 3 resource types *A* (with 10 instances), *B* (with 5 instances), and *C* (with 7 instances).
   - At time T0, the snapshot of data structures maintained by the OS are as follows:

Process	**Allocation Matrix**	**Max Matrix**	**Need Matrix**	**Available Vector**
	A B C	A B C	A B C	A B C
*P*0	0 1 0	7 5 3	7 4 3	3 3 2
P1	2 0 0	3 2 2	1 2 2	
P2	3 0 2	9 0 2	6 0 0	
P3	2 1 1	2 2 2	0 1 1	
P4	0 0 2	4 3 3	4 3 1	

   (a) Can request vector for (3,3,0) by P4 be granted?

   (b) Can request vector for (0,2,0) by P0 be granted?

4. Given the following system state that defines how 4 types of resources are allocated to 5 running processes.

Available = {2, 1, 0, 0 }

$$\textbf{Allocation} = \begin{array}{c} \\ P0 \\ P1 \\ P2 \\ P3 \\ P4 \end{array} \begin{array}{c} \begin{array}{cccc} r0 & r1 & r2 & r3 \end{array} \\ \left\{ \begin{array}{cccc} 0 & 0 & 1 & 2 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 4 \\ 2 & 3 & 5 & 4 \\ 0 & 3 & 3 & 2 \end{array} \right\} \end{array} \quad \textbf{Max} = \left\{ \begin{array}{cccc} 0 & 0 & 1 & 2 \\ 2 & 7 & 5 & 0 \\ 6 & 6 & 5 & 6 \\ 4 & 3 & 5 & 6 \\ 0 & 6 & 5 & 2 \end{array} \right\}$$

a) Complete the Need matrix :

$$\textbf{Need} = \left\{ \begin{array}{cccc} \_\_\_ & \_\_\_ & \_\_\_ & \_\_\_ \\ \_\_\_ & \_\_\_ & \_\_\_ & \_\_\_ \\ \_\_\_ & \_\_\_ & \_\_\_ & \_\_\_ \\ \_\_\_ & \_\_\_ & \_\_\_ & \_\_\_ \\ \_\_\_ & \_\_\_ & \_\_\_ & \_\_\_ \end{array} \right\}$$

b) Is the system in a safe state? Why or why not? If in a safe state, give a safe sequence.

c) For each of the following requests:
   P0: *Request* = {0, 1, 0, 0}
   P1: *Request* = {0, 1, 0, 0}
   P2: *Request* = {0, 1, 0, 0}
   P3: *Request* = {0, 0, 0, 1}
   determine if the request should be granted. If it can be granted, show that the system is in a safe and give a safe sequence.

5. Deadlock Detection: Given the following system resource allocation state:

Available = { 2, 1, 0, 0 }

$$\textbf{Allocation} = \left\{ \begin{array}{cccc} 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 \end{array} \right\} \qquad \textbf{Request} = \begin{array}{c} P0 \\ P1 \\ P2 \end{array} \left\{ \begin{array}{cccc} 2 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \end{array} \right\}$$

a) Determine if a deadlock exists.

b) Illustrate the system with a resource allocation graph.