Due April 7th, 2011

Homework 6

Objective:

Designing and implementing the banker algorithm for deadlock detection for several instance of a resource type.

Introduction:

For this implementation you need several data structure;

- **Available:** A vector or array of length m indicates the number of available resources of each type.
- **Max:** A n x m matrix defines the maximum demand of each process. If Max[i][j] equals k, then process P_i may request at most k instances of resource R_j
- **Allocation:** An n x m matrix defines the number of resources of each type currently allocated to process.
- **Need:** an n x m matrix indicates the current need of each process. If Need[i][j] equals k, then process P_i is requesting k more instances of resource type R_j. (*Need =Max-Allocation*).

More Details:

To illustrate the banker algorithm, consider a system with R resources and P processes such as each resource type R_j has a W number of instances. For Example, a system with 6 resources (j=6) and 7 processes (P=7) is shown in Figure 1.

For your implementation, you have to generate all this numbers by using your programming language's integer generator function.

Write a program to do the followings;

a) Generate numbers for R and P such as 5 < R < 15, 10 < P < 30.

b) Generate numbers for resource instances such as 1 < W < 7 . For example, If R is 6, your program must generate 6 integers for each resource type such as R_1 has 6 instances, R_2 has 7 insurances, R_3 has 3 instances, R_4 has 6 instances, R_5 has 2 instance, and R_6 has 2 instance.

c) Fill out the Allocation and Max Matrices by generating random integers.

Hint: if X € **Max** than 0 < X <= W. If Y € **Allocation** than 0 < Y <= X

d) Find the Available vector and Need Matrix.

e) Detect if the system is in a safe state or unsafe state. If the system is currently in a safe state. Print the sequence that satisfied the safety, e.g. P2, P3, P4, P0, P4, P5, P6. If the system is in an unsafe state it must print “unsafe state”.

Allocation

	R_1	R_2	R_3	R_4	R_5	R_6
P_0	Y					
P_1						
P_2						
P_3						
P_4						
P_5						
P_6						

Max

	R_1	R_2	R_3	R_4	R_5	R_6
P_0	X					
P_1						
P_2						
P_3						
P_4						
P_5						
P_6						

Available

Figure 1