Operating Systems 2 – Spring 2022
Theory Assignment 2: Deadlock Exercises

Deadline: 11th March 2022, 9:00 am (before the class)

Variant of Question 8.4:

A possible method for preventing deadlocks is to have a single, higher-order resource that must be requested before any other resource. For example, if multiple threads attempt to access the synchronization objects A ⋯ E, deadlock is possible. (Such synchronization objects may include mutexes, semaphores, condition variables, and the like.) We can prevent deadlock by adding a sixth object F. Whenever a thread wants to acquire the synchronization lock for any object A ⋯ E, it must first acquire the lock for object F. This solution is known as containment: the locks for objects A ⋯ E are contained within the lock for object F. Is there any drawback of this scheme?

Question 8.6:

Consider a computer system that runs 5,000 jobs per month and has no deadlock-prevention or deadlock-avoidance scheme. Deadlocks occur about twice per month, and the operator must terminate and rerun about ten jobs per deadlock. Each job is worth about two dollars (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 of about 10 percent in the average execution time per job. 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.

A. What are the arguments for installing the deadlock-avoidance algorithm?
B. What are the arguments against installing the deadlock-avoidance algorithm?

Question 8.22:

Consider a system consisting of four resources of the same type that are shared by three threads, each of which needs at most two resources. Show that the system is deadlock free.

Question 8.24:

Consider the version of the dining-philosophers problem in which the chopsticks are placed at the center of the table and any two of them can be used by a philosopher. Assume that requests for chopsticks are made one at a time. Describe a simple rule for determining whether a a particular request can be satisfied without causing deadlock given the current allocation of chopsticks to philosophers.

Question 8.28:

Consider the following snapshot of a system:

	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D

T0	3	1	4	1	6	4	7	7	2	2	2	4
T1	2	1	0	2	4	2	3	2				
T2	2	4	1	3	2	5	3	3				
T3	4	1	1	0	6	3	3	2				
T4	2	2	2	1	5	6	7	5				

Answer the following questions using the banker's algorithm:

A. Illustrate that the system is in a safe state by demonstrating an order in which the threads may complete.
B. If a request from thread T4 arrives for (2, 2, 2, 4), can the request be granted immediately?
C. If a request from thread T2 arrives for (0, 1, 1, 0), can the request be granted immediately?
D. If a request from thread T3 arrives for (2, 2, 1, 2), can the request be granted immediately

3 3 3 6
2 1 3 0
0 1 2 0
2 2 2 2
3 4 5 4