

Operating Systems 2 – Spring 2023

Theory Assignment 3: Deadlock Exercises

Deadline: 2nd April 2023, 9:00 pm

1. 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 $A \cdots E$, 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 $A \cdots E$, it must first acquire the lock for object F . This solution is known as containment: the locks for objects $A \cdots E$ are contained within the lock for object F . Is there any drawback of this scheme?

2. Question 8.8 on page 345 of the book.

3. Question 8.22 on page EX-29 of the book.

4. Variant of 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. (a) Describe a simple rule for determining whether a particular request can be satisfied without causing deadlock given the current allocation of chopsticks to philosophers. (b) Explain the practical difficulty in implementing this rule.

5. Question 8.28 on page EX-30 of the book: