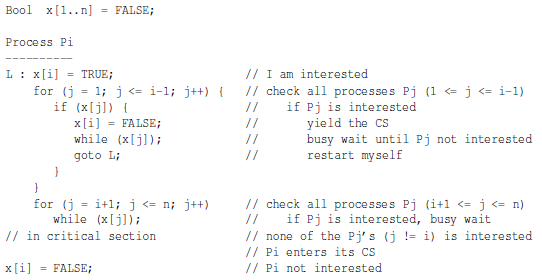
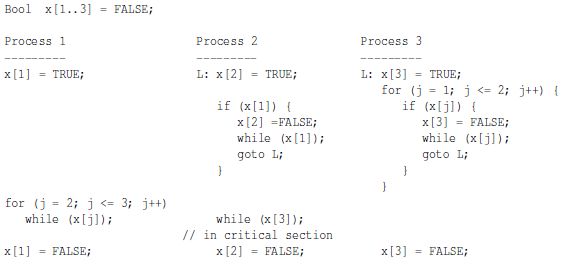
Fall 2013 Final Exam

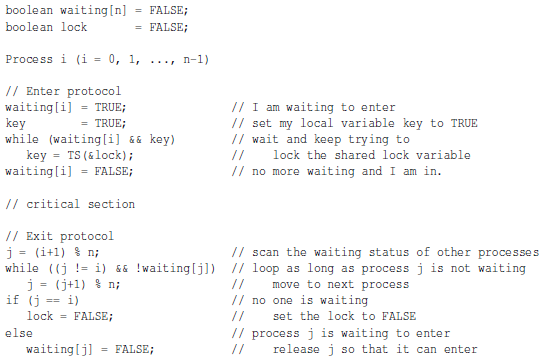
1. Define the meaning of a race condition? Use an execution sequence with argument.
2. The following is a solution to the critical section problem for n processes where is a known integer. The general code for process is below.



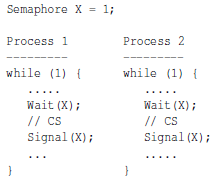
Convert the above code to a 3-processes version for . Prove rigorously that the above algorithm for the tree processes satisfies the mutual exclusion requirement.



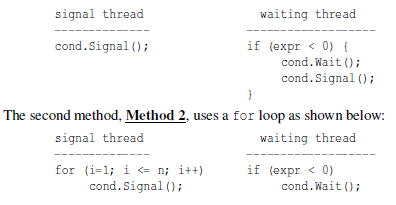
1. Consider the following solution to the critical section problem using the atomic TS instruction. This solution works for processes, where is a know integer. It was shown that this solution satisfies mutual exclusion. Show that this solution also satisfies progress and bounded waiting.



1. Consider the following implementation of mutual exclusion with a semaphore X. show rigorously that the above implementation satisfies mutual exclusion.



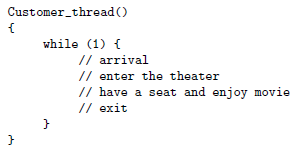
1. Suppose in a Hoare type monitor a thread waits on condition variable cond if the value of an expression expr is negative. The value of expr my be modified by other threads, and the thread that makes expr non-negative signals cond to release a waiting thread. There are two different ways to signal all threads that are waiting on cond. Suppose we know that the total number of waiting threads is n. The first method, Method 1, does not have to know the value of n and uses cascading signaling as follows. In this way, a released thread can signal cond immediately to release another.



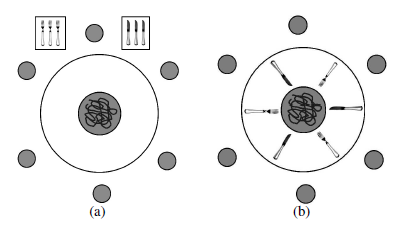
Study both methods and answer this question: In a Hoare type monitor, which method or methods can guarantee that the expression exp is non-negative when a waiting thread is released?

1. A movie theater with m seats shows movies non-stop. Customer can come and go at any time, but the theater cannot seat more than m customers. For some unknown reason, this theater has a design flaw. The entrance door allows customers to get in one-by-one without affecting movie showing in any way. But, opening the exit door would cause too much light into the theater. Therefore, the owner of this theater made the following rules:
   1. Customers can get into the theater one at a time
   2. Customers leaving the theater must be in a group of n so that the exit door will not be opened too often
   3. The entrance door cannot be used for exiting and only allows customers to enter one-by-one
   4. The exit door is exit only and cannot be used to enter the theater

Each customer is represented as a thread. Complete the above code segment with semaphores to implement the theater’s policy. You must prevent a fast exiter who may come back into the theater, join the current exiting group, and exit again. In other words, members of the current exiting group must exit completely before admitting the next m exiting customers.



1. Each thread in a system has a unique ID, which is a positive integer. The system also has a shared file that can be accessed by multiple threads simultaneously as long as the sum of the ID’s of all threads that are currently accessing the file is less than a predefined value MAXIMUM. Design a Hoare monitor Strange and monitor procedures Access(id) and Release(id) where id is the ID of the calling thread. Monitor procedure Access(id) allows the caller to access the file if the sum of all the ID”s and id is less than MAXIMUM. In this case, Access(id) returns. Otherwise, the caller is blocked until the condition will be met in the future. On the other hand, when a thread finishes accessing the shared file, it calls monitor procedure Release(id) to release the file. Hint: Watch out the way of releasing threads.
2. Use semaphores to design an asynchronous Mailboc class of capacity and its two methods Send() and Receive(). The Send(int x) take an int as its argument and sends the int to the mailbox, and the Receive( int \*x) receives an int from the mailbox. You may use simplified syntax for the semaphore declaration and operations. For example, Sem X = 0 declares a semaphore X with initial value 0, and Signal(X) and Wait(X) signals and waits on semaphore X.
3. Enumerate and elaborate all major differences between a semaphore wait/signal and a condition variable wait/signal.
4. Why is calling a monitor procedure from within another monitor (i.e. nested monitor call) not a good programming practice?
5. Three processes share four resource units that can be acquired and released only one at a time. Each processes needs maximum of two units at any time. We also know that these resource units must be used in a mutually exclusive way and that they are non-preemptable. Show that this is a deadlock free system. Hint: Think about the necessary condition of a deadlock.
6. There are six philosophers seated at a circular table. There are three knives and three forks available. The philosophers are quite hungry but require both a fork and a knife to eat.

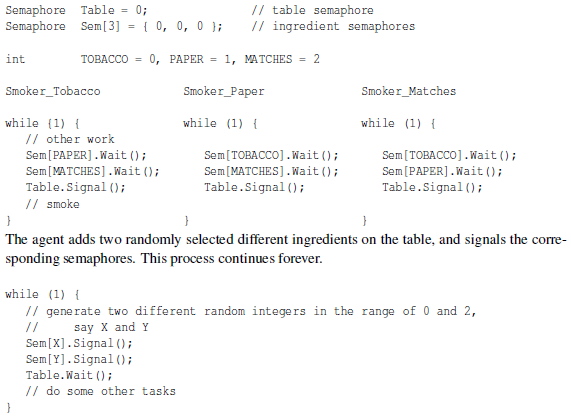


Do the following problems:

* 1. Each philosophers goes to the tray (Figure (a) above) and grabs any knife until he is successful. He then grabs any fork until he is successful. Once he has a fork and a knife, he eats and finally returns the fork and knife.
  2. If the forks and knives are arranged in an alternating way as shown in Figure (b), each philosopher flips a coin to determine if he is going to first try for a fork or for a knife. He grabs his choice until he is successful and grabs the other type of utensil until he is successful. Then he eats and returns the utensils after eating.

For each problem, answer if a deadlock is possible. If not, clearly explain why. If it can, use an execution sequence to show the existence of a deadlock.

1. Three ingredients are needed to make a cigarette: tobacco, paper, and matches. An agent has an infinite supply of all three. Each of the three smokers has an infinite supply of one ingredient only. The following solution uses three semaphores, each of which represents an ingredient, and a fourth one to control the table. A smoker waits for the needed ingredients on the corresponding semaphores signals the table semaphore to tell the agent that the table has been cleared, and smokes for a while.



Show using execution sequences that this solution can have a deadlock.

1. What is rendezvous in message passing? You must provide the context of a rendezvous and a clear explanation.