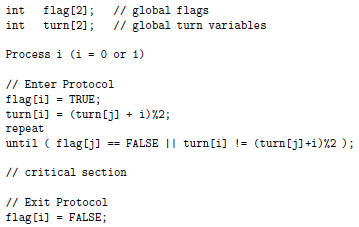
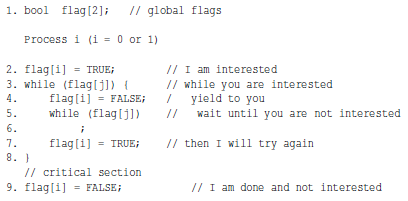
Exam 2 Fall 2018

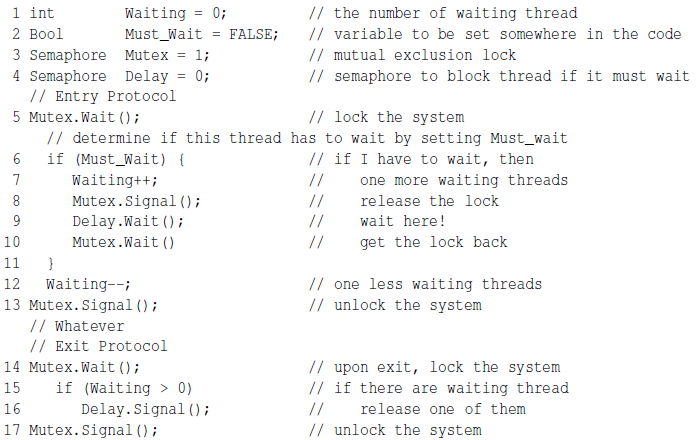
1. Consider the following solution to the mutual exclusion problem for two processes where flag is a Boolean array of two elements and turn is an int array, each of its two elements can only hold 0 or 1. Note that flag and turn are global variables shared by both processes. Prove that this solution satisfies mutual exclusion.



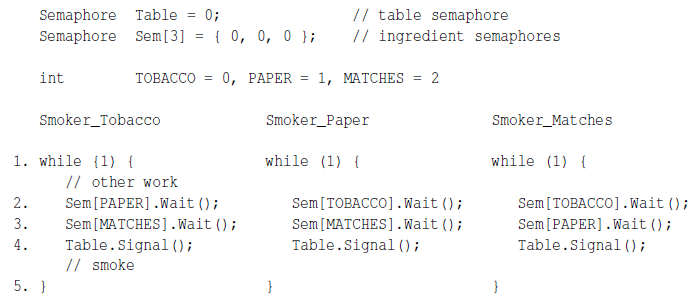
1. Consider the following solution to the mutual exclusion problem for two processes where flag is a global Boolean array of two elements. Explain whether the progress and bounded waiting conditions are met. For each condition, either provide a proof showing the condition is met or provide an execution sequence showing it is violated.



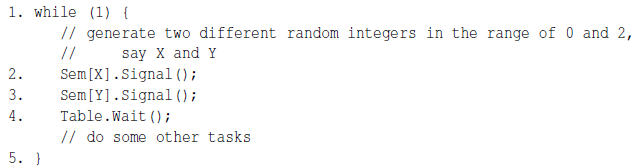
1. Define the meaning of a race condition. Use an execution sequence.
2. The semaphore methods wait and signal must be atomic to ensure mutual exclusion. Use execution sequences to show if wait is not atomic then mutual exclusion cannot be maintained.
3. As discussed, it’s typical that a thread must do something (entry protocol) before doing its core task and do something else upon exit (exit protocol). The entry and exit protocols must perform locking and unlocking activities in order to access shared data items to prevent race conditions. However, too many locking and unlocking activities is inefficient, and the passing the baton technique may be used to cut some redundant locking and unlocking activities. The following shows an entry and exit protocol without the use of passing the baton. Modify it to use passing the baton. Just directly edit the code and indicate clearly how the baton is passed.



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. That is, one has tobacco, the second has paper, and the third has matches. The following solution uses three semaphores, each of which represents an ingredient, and a fourth 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.

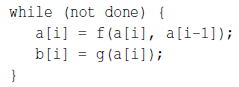


The agent adds two randomly selected different ingredients on the table and signals the corresponding semaphores. This process loops indefinitely.



Show using execution sequences that this solution can have deadlock.

1. A multithreaded program has two global arrays, a and b, and a number of threads that execute concurrently. Thread Ti runs the following code, where thread Ti passes the value of a[i-1] and the value of a[i] to f() to compute the new value for a[i] which is passed to function g() to compute b[i]. Declare semaphores with initial values and add wait() and signal() calls to thread Ti so it will compute the result correctly.



1. A main highway cuts through a rural road as shown below. East-bound cars are on the highway, while south-bound cars are on the rural road. To avoid delays on the highway, the following traffic regulations are implemented. Write code for the east-bound and south-bound and add semaphores and variables as needed.
   1. As long as there are east-bound cars, they don’t have to stop. In this case, south-bound cars must stop.
   2. If there is a south-bound car crossing, all east-bound vehicles must stop.
   3. To prevent south-bound cars from blocking the highway, only one south-bound car can enter the intersection. However, multiple east-bound cars may cross the intersection at the same time.
   4. If east-bound cars and south-bound cars approach the intersection at the same time, only one can proceed and it can be either one.

