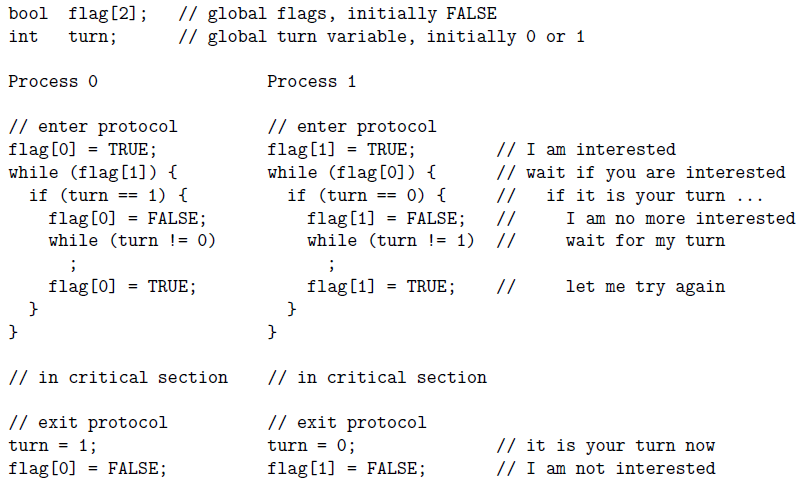
Fall 2013 Exam 1

1. Explain interrupts and traps, and provide a detailed account of the procedurethat an operating system handles an interrupt.
2. Explain what the CPU modes are and their uses. How does the CPU know what mode it is in?
3. What is a context? Provide a detail description of all activities of a context switch.
4. Draw the state diagram of a process from its creation to termination, including all transitions. Make sure you will elaborate **every state** and **every transition** in the diagram.
5. Enumerate the major differences between kernel-supported threads and user-level threads.
6. Define the meaning of a race condition? Answer the question first and use an execution sequence with a clear and convincing argument to illustrate your answer.
7. Explain the progress and bounded waiting conditions and enumerate their differences.
8. Jason and John are sharing an apartment, and they also share the responsibility of buying milk. To avoid the possibility of both buying, they come up with the following “algorithm”: As soon as you arrive home, you leave a signed note on the fridge’s door. Only then you check, and if you find that there is no milk and there is no note (other than yours), then you go and buy milk, put the milk in the fridge, and remove your note. Their pseudo-code looks like the following:

|  |  |
| --- | --- |
| **Jason** | **John** |
| **Jason leaves note**  **If (no note from John) then**  **If (no milk) then**  **Buy milk**  **End if**  **End if**  **Remove Jason’s note** | **John leaves note**  **If (not note from Jason) then**  **If (no milk) then**  **Buy milk**  **End if**  **End if**  **Remove John’s note** |

Jason and John both suspect that they may end up no milk at all! Use a step-by-step execution of the above “algorithm” to show that Jason and John **can** end up with no milk at all. Note that Jason and John cannot see and talk to each other.

1. Consider the following solution to the critical section problem for two processes.



Show that this solution satisfies the mutual exclusion condition.