

EECS 343: Homework 3

Threads and Synchronization

Fall 2014

Important Dates

Out: November 7, 2014.

Due: November 14, 2014, 11:59PM CDT.

Submitting your assignment: Please use the course submission site. There is a link to it from the class site.
Submit only ASCII text files.

Problems

1. In class we listed the register set as a per-thread rather than per-process item. Why? After all, the machine has only one set of registers.
2. If a multithreaded process forks, a problem occurs if the child gets copies of all the parent's threads. Suppose that one of the original threads was waiting for keyboard input. Now two threads are waiting for keyboard input, one in each process. Does this problem ever occur in single-threaded processes?
3. Can two threads in the same process synchronize using a kernel semaphore if the threads are implemented by the kernel? What if they are implemented in user space? Assume that no threads in any other processes have access to the semaphores.
4. Show how counting semaphores can be implemented using only binary semaphores and ordinary machine instructions.
5. Consider a system in which threads are implemented entirely in user space, with the run-time system getting a clock interrupt once a second. Suppose that a clock interrupt occurs while some thread is executing in the run-time system. What problem might occur? Can you suggest a way to solve it?