

EECS 338 Assignment #6: Posix Thread programming

Due: Thursday, April 12th, 2007

Spring 2007, G. Ozsoyoglu

In this assignment you will use posix threads to implement the threaded version the Fair Readers-Writers Problem from assignment 5.

Fair Readers-Writers Problem. This is a variant of the Fair Readers-Writers Problem **where NOBODY STARVES**: When there are readers and writers waiting to read and write (due to a writer or reader in its CS), the service of the system becomes first-come-first-serve (FIFO) with the provision that consecutive FIFO readers are still allowed to read concurrently. You may assume that you know the total number of readers and writers in the system.

Your solution consists of twelve threads, eight for readers, and four for writers. Arrange two execution scenarios, one in which readers and writers arrive randomly to read and write, respectively, and another one in a carefully controlled manner to demonstrate the correctness of your implementation. For each action, print a message to the screen containing the thread-id of the involved thread, the action, the current time (and, if you use a queue, the state of the queue).

Remember to error-check all system calls: check return values for success. Also, do not forget to conform to the assignment grading policy requirements listed at <http://art.case.edu/338.S05/grpolicy.html>

Run your program in the script environment, just like in the previous assignments.

On the due date (April 12th),

1. Type and submit the Fair Readers-Writers algorithm that you have used to implement your thread-based solution. Your algorithm can use thread-based synchronization primitives, such as `mutex_trylock`, as well as `mutex_lock` and `mutex_unlock`.
2. Submit a hardcopy printout of your code and output in the class, and
3. Upload your assignment (typed algorithm, source code, and the printout) to the blackboard.

Remember in your implementation to error-check all system calls: check return values for success, and use *perror* when possible on failure.