CST 334: Operating Systems

Dr. Glenn Bruns

# Implementing a readers-writers lock

Important: please remember the class honor code. For example, do not use the web to see how to implement a readers-writers lock.

**Instructions**. In class we covered Professor Mike Dahlin’s method for writing pthreads code. Read the short note [Basic Threads Programming](https://drive.google.com/file/d/1RKTRORxgWYHMow8bs_HPqGaDeCV8Uoxo/view?usp=sharing) by Mike Dahlin.

Here’s a reminder of the basic method:

1. add a lock to the shared object
2. add code to each method to acquire the lock at the beginning of the method and release it at the end
3. add zero or more condition variables to the object
4. add wait calls on condition variables within loops
5. finally, add signal calls

In this programming project you will start with some C code I provide (see below), and finish it by adding code where you see a comment that says ‘YOUR CODE HERE’.

This purpose of the code is to implement a “readers-writers lock”. The idea of a readers-writers lock is that we have multiple programs wanting to concurrently access data. It is okay for multiple programs to access the data at the same time as long as they are only reading. But, if some program is modifying the data, then no other programs can access the data at the same time.

More precisely, with a reader-lock, multiple threads can access the data at the same time, but no 'writing thread' can access the data while either a reading thread or another writing thread is accessing the data. If a writing thread is accessing the data, it must be the only thread accessing the data.

Copy the file

/home/CLASSES/brunsglenn/cst334/hw/hw10/rwlock-skeleton.c

to a directory of your own, and rename it to ‘rwlock.c’. Read the file to understand it. The RWLOCK structure and the functions ‘rwlock()’, ‘start\_read()’, ‘done\_read()’, ‘start\_write()’, and ‘done\_write()’ are part of the rwlock “class” (C has no classes but this code is similar to a class definition). The purpose of the ‘reader()’, ‘writer()’, and ‘main()’ functions is to exercise the rwlock class.

You should be able to compile the code like this:

gcc -pthread -lm -o rwlock rwlock.c

Notice that the reader() function prints “sr” and “dr”, which are short for “start read” and “done read”. Similarly, the writer() function prints “sw” and “dw”.  **The code you submit must produce as output only lines with either ‘sr’, ‘dr’, ‘sw’, and ‘dw’**.

**Testing your code**. In another part of this assignment you will write an awk script to test your code.

**Submitting**. Submit your rwlock.c on iLearn.

**Grading**. I will run 4 tests on your code, each of which is worth 15 points.