

COP 4520 Spring 2020

Project Assignment 1

Notes:

Please, submit your work via Webcourses.

Submissions by e-mail will not be accepted.

Due date: Wednesday, March 18th by 11:59 PM

Late submissions are not accepted.

This is a team assignment.

You can use a programming language of your choice for this assignment.

If you do not have a preference for a programming language, I would recommend C++.

Grading policy:

General program design and correctness: 50%

Efficiency: 30%

Documentation including statements and proof of correctness, efficiency, and experimental evaluation: 20%

Additional Instructions:

Cheating in any form will not be tolerated.

In addition to being parallel, your design should also be efficient in terms of execution time and memory use.

The goal of this assignment is to gain familiarity with the concurrent data structures described in your main topic paper and implement a version of it based on locks. Here are the required steps for Project Assignment 1:

1. Read the main topic paper and focus on the main operations supported by the data structure or algorithm described in it.
2. Implement a sequential version of this data structure. In this sequential version, of course, you do not have to worry about any of the synchronization mechanisms described in the paper. The goal is to implement a straightforward single-threaded version of the container.
3. Use the MRLOCK algorithms to convert your sequential implementation in a multi-threaded implementation. One possible strategy to do so is to map each shared variable as a resource.

You can find a C++ implementation of MRLOCK, here:

<http://cse.eecs.ucf.edu/gitlab/deli/mrlock>

Here is a link to the paper describing MRLOCK and how it works:

https://link.springer.com/chapter/10.1007/978-3-319-03850-6_19

Do not buy the pdf file, it is free if you access the digital library from a campus network.

I will also post the paper on Webcourses.

If you are using a language other than C++, it is your responsibility to re-implement MRLOCK for that language or platform.

Do not worry about garbage collection and make sure to pre-allocate all elements that you might need in your tests.

4. Write a main program where you will spawn 4 threads and each will execute a random mix of operations on your multi-threaded MRLOCK version of the data structure.
5. Write a brief report that will include: a) a quick summary of the data structure and its key methods and properties, b) discuss the technical details of your implementation including the use of MRLOCK and the key algorithms of your implementation, and c) include a list of references including a citation to the MRLOCK algorithm, your main topic paper, and other relevant approaches.
6. Publish your report according to the publication instructions and submit your work via Webcourses.