

CS/SE 4348: Operating Systems Concepts

Section 0U1

Programming Project 3

Instructor: Neeraj Mittal

Assigned on: Monday July 18, 2022

Due date for pseudocode: Monday July 25, 2022 (at midnight)

Due date for program: Wednesday, August 3, 2022 (at midnight)

This is a group assignment. A group can consist of *two students*. (You must work with at least one other student on this project.) Each group is expected to work independently and submit only their own work.

Copying or using work not your own will result in disciplinary action and the suspected incident will be referred to the Office of Community Standards and Conduct for investigation!

1 Project Description

Design and implement a **barrier** using monitors. Your barrier should support one method, namely `arriveAndWait()`. A thread invoking the method is blocked until all threads have arrived at the barrier (in other words, invoked the `arriveAndWait()` method of the barrier) at which point all threads are released from the barrier and can resume their respective executions. The barrier should be **reusable**, that is, threads can use the same barrier object multiple times to synchronize with each other. Ensure that your design and implementation of a barrier satisfies the following two properties: (a) *safety*: no thread is released from the barrier prematurely, and (b) *liveness*: once all threads have arrived at the barrier, all threads are released from the barrier eventually.

1.1 Testing your Barrier

To test your barrier, you will be provided with a program that solves the prefix sum problem using Hillis and Steele's parallel algorithm. The algorithm executes in phases. All threads must complete the current phase before any thread can advance to the next phase, which is achieved using a barrier.

1.2 Monitors

Write your program in Java. For an example of using monitors in Java, please see <https://docs.oracle.com/en/java/javase/18/docs/api/java.base/java/util/concurrent/locks/Condition.html>. The example shows how to solve the producer consumer (bounded buffer) problem using monitors in Java.

2 Grading Criteria

As such, projects will be graded with these criteria in mind:

- Solutions must adequately address the problem at hand. Specifically:
 - The solution represents a good-faith attempt to actually address the requirements for the assignment.
 - The program complies and executes.
 - The program runs correctly.
- The solution constitutes a high quality product expected of a professional. Specifically:
 - The program is easy to read and to understand, that is, it is well commented. In addition, method and variable names are meaningful, all potentially confusing/complex code is well documented.
 - The general design of the program is clear and reasonable.
 - All procedure and function headers include comments explaining what the method is supposed to do (not how it does it) and the purpose of each formal parameter. Be as precise and careful as you can be.
 - The program is robust and handles important errors and exceptions properly.

3 Submission Information

You have to submit your project through eLearning. The submission will consists of two steps:

- (i) Submit the pseudocode of your solution by Monday July 25th.
- (ii) Submit all the source files by Wednesday August 3rd. Along with all the source files, submit the following: (a) a Makefile to compile the program and (b) a README file that contains the names of all the group members and instructions for running the compiled program.

Points will be deducted if you fail to submit either a Makefile or README file. Only one team member needs to submit the files.