LAB 12

CS 361: Systems Programming / Spring 2023

Description

In this lab session, you will explore how to use mutexes and semaphores to synchronize thread execution by implementing a barrier as seen in class.

Guide

1. Accept the invitation for Lab 12 on Github classroom: https://classroom.github.com/a/ar8mM4R_
2. Import the Github repository created to your machine using vscode, as explained in Assignment 0
3. Make sure that you can launch a terminal inside vscode via menus: Terminal > New Terminal
4. Read this guide and answer the questions as they appear. You should answer a total of 7 questions.

Barriers

Please open the slides for Lecture 21 about how to implement barriers. It is also helpful to open your solution to Labs 10 and 11, as you will be creating threads and semaphores/mutexes.

Question 1: Inspect file lab12-1.c and list all variables that function thread uses. Which are shared, and which are private?

Question 2: Describe, in your own words, what function thread does.

Question 3: Change line 48 to launch N threads, and line 52 to wait for all N threads.

Question 4: Compile (using make) and run lab12-1 (using ./lab12-1) with 10 threads and 10, 100, and 1000 rounds. Is there a single thread that wins all the races? If so, which?

Question 5: Inspect file lab12-2.c. What is different on function thread between lab12-1.c ?

Question 6: Implement a barrier on file lab12-2.c as seen in class. Add all the data you need to struct cs361_barrier_t, and implement functions cs361_barrier_init (which should initialize a barrier to wait for the given number of threads) and cs361_barrier_wait (which should only release threads when all threads reach the barrier). Change lines 73 and 77 as you did in Q3.

Question 7: Compile (using make) and run lab12-2 (using ./lab12-2) with 10 threads and 10, 100, and 1000 rounds. The results should be very different from your answer to Q4. How different are they?

Extra / Optional

Can you explain why lab12-1 behaves as it does, and why adding a barrier in lab12-2 changes that?

Grading

Show your UIC card to the TA when you enter the lab, or type your UIN on the chat when joining remotely. Stay in the session until you show your work, or until the TA announces that the lab is over.

- You have to remain present for the whole lab to get attendance, which you can then use to resubmit Assignment 5.
- You can leave early after showing your work to the TA (answers to all questions). In this case, you will get a 5% bonus in Assignment 5.