

Programming Assignment 1

Problem 1 (50 pts):

Your non-technical manager assigns you the task to find all primes between 1 and 10^5 . The assumption is that your company is going to use a parallel machine that supports five concurrent threads. Thus, in your design you should plan to spawn 5 concurrent threads that will perform the necessary computation. Your boss does not have a strong technical background but she is a reasonable person. Therefore, she expects to see that the work is distributed among the threads in a fair way. Finally, you need to provide a brief summary of your approach and an informal statement reasoning about the correctness and efficiency of your design. Remember, that your company cannot afford a supercomputer and rents a machine by the minute, so the longer your program takes, the more it costs. Feel free to use any programming language of your choice as soon as you provide a ReadMe file with instructions for your manager explaining how to compile and run your program. Using a makefile is highly recommended.

Problem 2 (50 pts):

The Dining Philosophers problem was invented by E. W. Dijkstra, a concurrency pioneer, to clarify the notions of deadlock and starvation freedom. Imagine five philosophers who spend their lives just thinking and feasting. They sit around a circular table with five chairs. The table has a big plate of rice. However, there are only five chopsticks (in the original formulation forks) available (see Figure 1 of Chapter 1, Exercise 1 from our textbook). Each philosopher thinks. When he gets hungry, he sits down and picks up the two chopsticks that are closest to him. If a philosopher can pick up both chopsticks, he can eat for a while. After a philosopher finishes eating, he puts down the chopsticks and again starts to think.

1. Write a program to simulate the behavior of the philosophers, where each philosopher is a thread and chopsticks are shared objects. Notice that you must prevent a situation where two philosophers hold the same chopstick at the same time.
2. Amend your program so that it never reaches a state where philosophers are deadlocked, that is, it is never the case that each philosopher holds one chopstick and is stuck waiting for another to get the second chopstick.
3. Amend your program so that no philosopher ever starves.
4. Write a program to provide a starvation-free solution for any number of philosophers N .

Grading policy:

General program design and correctness: 50%

Efficiency: 30%

Documentation including statements and proof of correctness and efficiency: 20%

Additional Instructions:

The assignment is individual. You can discuss the problem with your fellow classmates but you are supposed to write your own solution. Cheating in any form will not be tolerated.

Please, submit your work via webcourses.

Submissions by e-mail will not be accepted.

Due date: Wednesday, Feb 22nd by 11:59 PM

Best of luck!!