Assignment 2: Introduction

ETH Zurich

1 Dining Philosophers: Synchronization

Imagine \( n \) philosophers who spend their lives just thinking and eating. They sit around a table with \( n \) chairs. The table has a big plate of rice. However, there are only \( n \) chopsticks available. The chopsticks are arranged in a way that there is exactly one chopstick between every pair of neighboring philosophers. A single chopstick can be picked by one philosopher at most. 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 resumes thinking.

1.1 Task

Write a program in the programming language of your choice which simulates the behavior of the philosophers. Each philosopher must be a thread and the chopsticks must be shared objects. Notice that you must prevent a deadlock situation where each philosopher holds one chopstick and is stuck waiting for a second chopstick. Optional: Make sure your solution is starvation free.

2 Amdahl’s Law

Use Amdahl’s Law to resolve the following questions.

1. Suppose a computer program has a method \( M \) that cannot be parallelized, and that this method accounts for 40% of the program’s execution time. What is the limit for the overall speedup that can be achieved by running the program on an \( n \)-processor multiprocessor machine.

2. Suppose the method \( M \) accounts for 30% of the program’s computation time. What should be the speedup of \( M \) so that the overall execution time improves by a factor of 2?

3. Suppose the method \( M \) can be sped up three-fold. What fraction of the overall execution time must \( M \) account for in order to double the overall speedup of the program?