

Project 2 - Tasos Lilis 031295951

Video: https://youtube.com/shorts/p4wrj_L1nmo

For project 2 I decided to go with C++ since it is a language that I feel comfortable with, and also had good instructions provided. In this program, I designed and implemented a program that simulates the dining-philosophers problem using POSIX mutex locks and condition variables. There are five philosophers, each of them have their own separate running thread. They alternate between two states, thinking and eating. Each thread sleeps for a random period of time between one to three seconds. I completed this report individually.

I first made a constant integer variable set at 5 (cannot be altered) for the total number of philosophers, then I made two states (thinking and eating) using an enumerator. The next variable is the one called state that takes in the number of philosophers as an argument. I then created three pthread mutex variables and conditional variables for synchronization. Also one to help maintain a concise output that doesn't overlap. I then make two functions to get both the index of the left and right philosophers from the current position. Then we go to another function that picks up the forks and eats when the forks on both sides are available, we use the functions to synchronize the access. It then prints it out to the console without any overlapping or corrupted outputs. We then make a new function to return the forks after eating. It now calls the philosopher to think. I then made a new function to help philosophers alternate between thinking and eating at random. I made this to prevent the main from being too overloaded with functions and such. Lastly we go to the main that manages all the threads and philosophers. We make ID's, initializers, threads, and more to help the main run. We then

destroy the mutexes and conditionals and return 0. That is the very overall simple design of my project.