Group Project 2: Synchronization

Date: 11:55 pm, October 13th (Sunday), 2024

Member’s name:

Member 1: Nhi Nguyen ID:030818149

Member 2: An Tran ID:030578689

In this project using POSIX threads, mutex lock and condition variables. We created five threads representing for 5 philosopher, each philosopher have to action: thinking or eating by pthread\_create() function, and each philosopher thread is passed its own unique philosopher number (0 to 4). There are three status for each philosopher : THINKING, NEED\_TO\_EAT and EATING and these status stored in status[] array. The mutex lock in the code help ensure that only one philosopher can attempt to pick up (pthread\_mutex\_lock()) or return chopsticks at a time (pthread\_mutex\_unlock()).

Here is some key functions we applied to our code:

1. **test(int philosopher\_number)**: check philosopher’s neighboring philosophers status in status[] array.
2. **take\_forks(int philosopher\_number)**: check if philosophers available to pick up the chopstick or have to put in a waiting status until the chopsticks are available by pthread\_cond\_wait().
3. **return\_forks(int philosopher\_number)**: return the chopsticks and notice philosopher’s neighboring philosophers the chopsticks are available.
4. **start\_dining(void\* num)**: the main loop for each philosopher thread, which alternates between thinking and eating in an infinite loop.

Contribution:

| project\_2.cpp | Nhi, An |
| --- | --- |
| Readme | An |
| Report | Nhi |
| Video | An |
| Submit + Organize | Nhi |

Nhi: 100%

An: 100%