

Project 4: Synchronization, Extra Credit

CS 311

Introduction

After pushing the corrections to your Project 4 programs, you can select one of the following problems to solve using the Pthreads library. Your corrections and your extra credit program are to be pushed to your Project 4 GitHub repository.

Dining Philosophers Variation, 20 pts.

The “obvious” solution to the Dining Philosophers problem suffers from deadlock:

```
while (1) {
    think();
    wait(left_chopstick);
    wait(right_chopstick);
    eat();
    signal(left_chopstick);
    signal(right_chopstick);
}
```

The chopsticks are modeled using binary semaphores. Eliminate the circular wait property necessary for a deadlock condition by using a counting semaphore to ensure that no more than four philosophers are trying to eat at any time.

Pascal Pizza Party Protocol, 40 pts.

The MaCS Department is throwing its annual Pascal Pizza Party, with a twist. A pizza maker brings out a single 42 slice pizza at a time, and then goes into a back room to continue devising evil programming projects for his operating systems students. The students at the party alternate between talking and eating. 40 of the students take a single slice of pizza at a time when they eat. The 41st student takes two slices at a time and the 42nd student takes three slices at a time. If either of these two latter students don't find enough available slices when they want to eat, they wait for the next pizza to appear. The student who takes the last slice of pizza notifies the pizza maker to produce the next pizza.

Write a program to solve this problem. You may use Pthreads semaphores, mutexes, and condition variables as you see fit. Hint: You'll probably want a single condition variable, used by all of the student threads. See `pthread_cond_broadcast()`.