CSE231: Operating Systems

Assignment 4

Name: Mihir Chaturvedi
Roll no.: 2019061
Section: B
Branch: CSE

In this assignment, we attempt to solve a modified version of the Dining Philosophers problem using a custom implementation of a counting semaphore using mutex primitives.

Blocking

A blocking semaphore blocks the thread until it receives a signal to continue.
In the my_semaphore_wait function, we first lock the semaphore's mutex. We then wait in a while loop using conditional waiting until the semaphore's value is greater than zero. We decrement the value of the semaphore and release the mutex.
In the my_semaphore_signal function, we lock the semaphore's mutex and increment the value of the semaphore. We signal the conditional variable if the counting semaphore has free slots.

Non-blocking

A non blocking semaphore does a single check to see whether the semaphore is available, and if not, returns.
In the my_semaphore_wait function, we check to see if the value of the semaphore is greater than zero, and if it is, decrement it and return positively. Else, we return 0.
In the my_semaphore_signal function, we lock the semaphore's mutex and increment the value of the semaphore.
The semaphore's wait is wrapped around a while loop that only continues when the wait function returns positively.