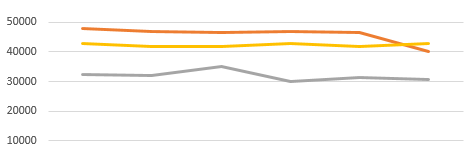
OS Project 2

# Semaphores

Upper limit = 5000

|  |  |  |  |
| --- | --- | --- | --- |
| Buffer Size | (1,1) | (1,5) | (5,1) |
| 10 | 47864 | 32522 | 42881 |
| 20 | 47003 | 31915 | 41733 |
| 30 | 46455 | 35189 | 41819 |
| 40 | 46971 | 29991 | 42935 |
| 50 | 46439 | 31198 | 41795 |
| 100 | 40299 | 30587 | 42811 |



The orange line is (1,1)

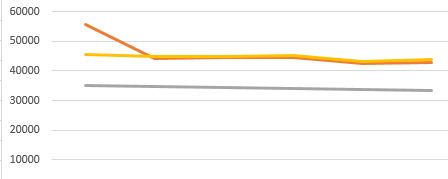
The gray line is (1,5)

The yellow line is (5,1)

# Spinlocks

Upper limit = 5000

|  |  |  |  |
| --- | --- | --- | --- |
| Buffer Size | (1,1) | (1,5) | (5,1) |
| 10 | 49434 | 35157 | 45516 |
| 20 | 44188 | 34737 | 44947 |
| 30 | 44464 | 34472 | 44866 |
| 40 | 44656 | 34008 | 45127 |
| 50 | 42659 | 33690 | 43290 |
| 100 | 42717 | 33433 | 43876 |



The orange line is (1,1)

The gray line is (1,5)

The yellow line is (5,1)

It was not possible to extend the critical section dramatically. Every time this was attempted, Hera2 would run very slow, and freeze. Because of this, the difference between the spinlock and semaphore efficiency is not easily noticeable. However, the spinlock is slightly faster than the semaphore. This is true because when the critical section’s length is short because the CPU usage of the spinlock is not as costly as the context switch of the semaphore. If the critical section was very large, the semaphore would have faster results, as the context switch does not take as many resources as the busy waiting of the spinlock when the critical section is long.

Both programs were observed to run faster with a larger number of consumers than producers. This could indicate that the process of removing the element from the buffer and printing it is consistently larger than the producer’s task of inserting into the buffer and incrementing the shared variable. It may also be due to the producer’s having an extra semaphore to guard the variable shared among them.