Blade Johnson

No synchronization

1st Test

X: 4, Y: 10000

Run 1: 27388

Run 2: 29971

Run 3: 30000

2nd Test

X: 8, Y: 100000

Run 1: 134598

Run 2: 122809

Run 3: 96617

3rd Test

X: 10, Y: 10000

Run 1: 90000

Run 2: 81829

Run 3: 75143

Locks

1st Test

X: 4, Y: 10000

Run 1: 30000

Run 2: 30000

Run 3: 30000

2nd Test

X: 8, Y: 100000

Run 1: 292941

Run 2: 278499

Run 3: 238574

3rd Test

X: 10, Y: 10000

Run 1: 90000

Run 2: 90000

Run 3: 90000

Test and Set

1st Test

X: 4, Y: 10000

Run 1: 30000

Run 2: 30000

Run 3: 30000

2nd Test

X: 8, Y: 100000

Run 1: 115066

Run 2: 118997

Run 3: 103605

3rd Test

X: 10, Y: 10000

Run 1: 90000

Run 2: 90000

Run 3: 90000

I have to preface this analysis with the fact that the tests were run on cs1, which would most likely account for why the 2nd tests were abysmal across the board. With no synchronization, there was no protection keeping the threads from constantly using the variable. This resulted in the program rarely finishing with the proper counter, and typically undershooting. Both the locks and test and set had similar results. Locks would block threads from interacting with the variable, while test and set used a boolean and a while loop to have the threads wait for their turn. By only allowing one thread at a time, the final counter hit consistent results.