

CS333 - Lab7

Samarth Mishra
130260018

Pradyot Prakash
130050008

February 29, 2016

1 Part A

1.1

Average value = 91152.67

Average time = 907 us

1.2

For the 3 runs, there was a singular run where the value of count turned out to be $100000 = N * K$. This was probably because the thread completed execution before it could be context switched out. This can be attributed to the small value of 'K'. We tried out the experiment with slightly higher values of 'K', and we observed that the values were not equal to $N * K$ sampled over a few runs.

For those runs where the values were different from the expected ones, the problem was because of line 13. Something similar to this could have been caused: The value of 'count' was loaded onto the register and then the particular thread could have been context switched out. The other thread of the same process could have performed some operations and when the first thread would have been scheduled, then it would have resumed its operation overwriting the work done by the other threads.

2 Part B

2.1

Average value = 89029.33

Average time = 1225 us

2.2

A similar explanation as the preceding part holds here, with the added problem of race conditions on the 'locked' variable. If some thread has gone past the while loop waiting for lock to be free, it may then get context switched out and another thread (may be more than 1) could also acquire the lock (change locked to 1), defeating the purpose of the software lock.

The time taken in this process was higher because of the added instructions because of the while loop and updates to the 'locked' variable.

3 Part C

3.1

Average value = 100000
Average time = 22340 us

3.2

There were no race conditions and all the runs gave the value of $\text{count} = N * K = 100000$. The mutex variable took care of using the proper hardware instructions to lock and unlock the variables atomically.

The time spent in waiting was orders of magnitude higher compared to the previous two cases.