PThreads!

By Trever Hall | March 2022

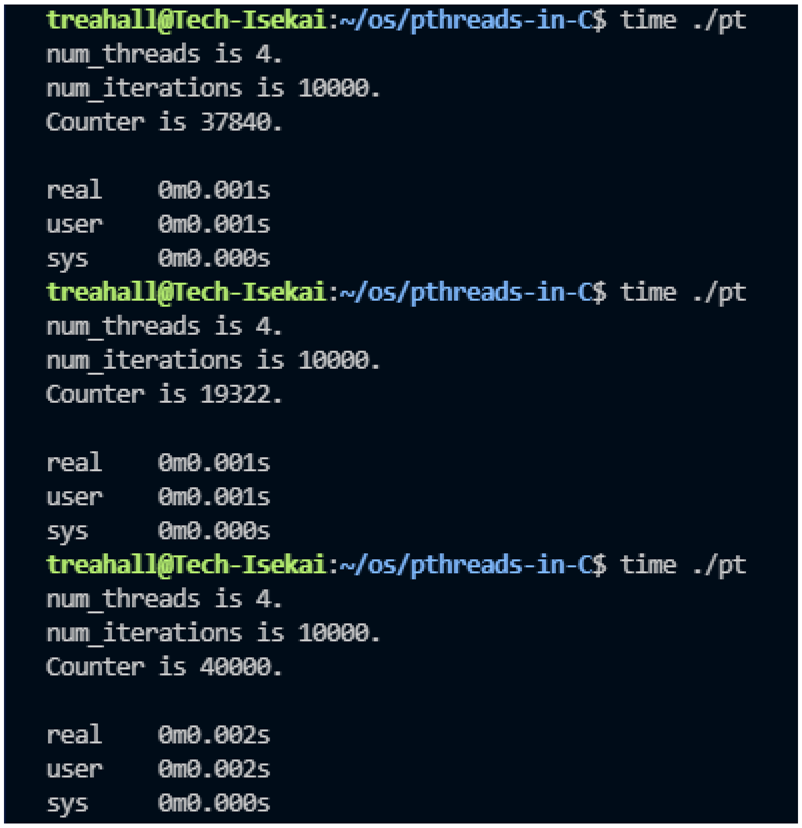
No Synchronization - Free for All

This is an implementation of pthreads that uses *pthread_join* to make sure all the threads have returned before printing what the counters value is.

Testing Default Values

The default settings are 4 threads with 10,000 each. The code would always finish executing but was rarely correct.

- Execution Time – stayed between .001 and .003 seconds in real time.
- Final Counter Values – very inconsistent and often not the expected value.

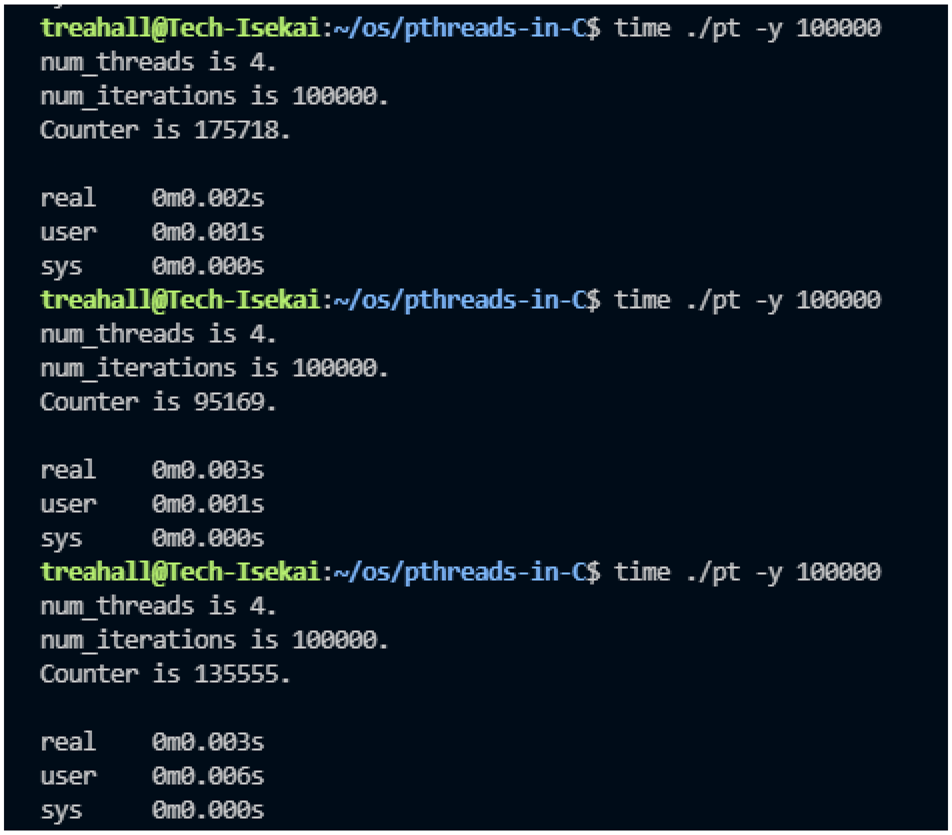


```
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt
num_threads is 4.
num_iterations is 10000.
Counter is 37840.

real    0m0.001s
user    0m0.001s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt
num_threads is 4.
num_iterations is 10000.
Counter is 19322.

real    0m0.001s
user    0m0.001s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt
num_threads is 4.
num_iterations is 10000.
Counter is 40000.

real    0m0.002s
user    0m0.002s
sys     0m0.000s
```

Testing With more Iterations

For this test I ran the code with 4 threads and 100,000 iterations. This increased the error rate considerably, since through all testing I was unable to get it to return the expected result of 400,000.

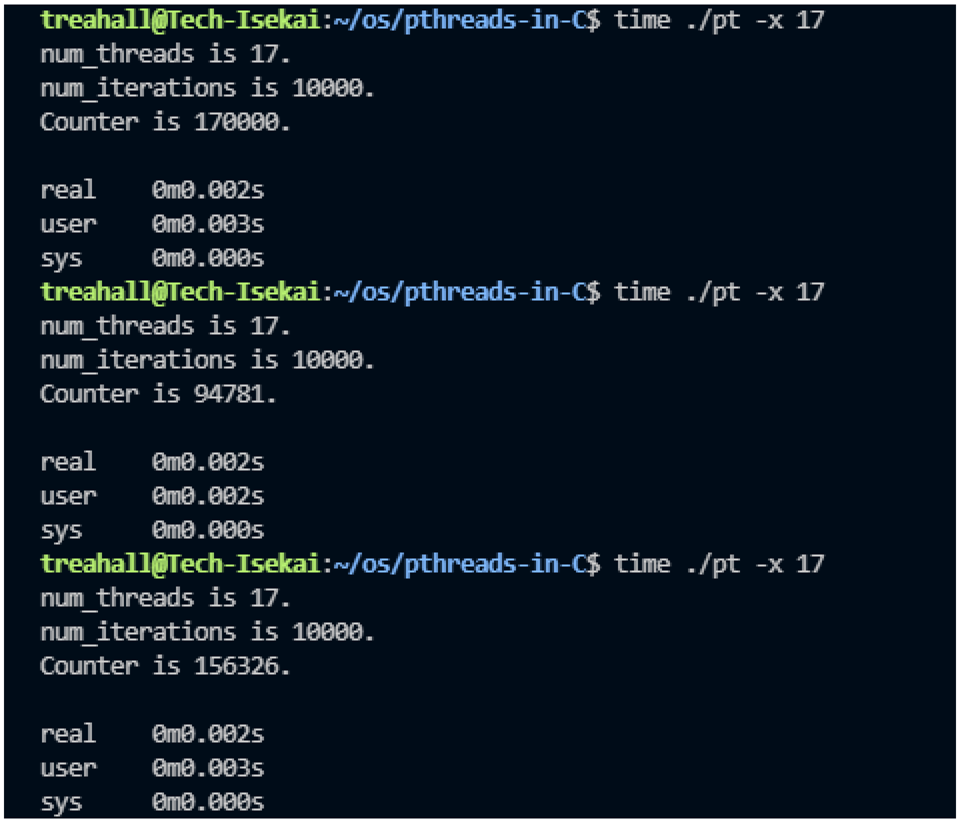
- Execution Time – between .002 and .003 seconds in real time.
- Final Counter Values – even more variation and inaccuracy.


```
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt -y 100000
num_threads is 4.
num_iterations is 100000.
Counter is 175718.

real    0m0.002s
user    0m0.001s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt -y 100000
num_threads is 4.
num_iterations is 100000.
Counter is 95169.

real    0m0.003s
user    0m0.001s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt -y 100000
num_threads is 4.
num_iterations is 100000.
Counter is 135555.

real    0m0.003s
user    0m0.006s
sys     0m0.000s
```

More Threads Than CPUs

Lscpu reported my computer has 16 CPUs, so I ran this test using 17 threads at default iterations.

- Execution Time – consistently .002 seconds in real time.
- Final Counter Values – not much different then running with fewer threads than CPUs.



```
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt -x 17
num_threads is 17.
num_iterations is 10000.
Counter is 170000.

real    0m0.002s
user    0m0.003s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt -x 17
num_threads is 17.
num_iterations is 10000.
Counter is 94781.

real    0m0.002s
user    0m0.002s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt -x 17
num_threads is 17.
num_iterations is 10000.
Counter is 156326.

real    0m0.002s
user    0m0.003s
sys     0m0.000s
```

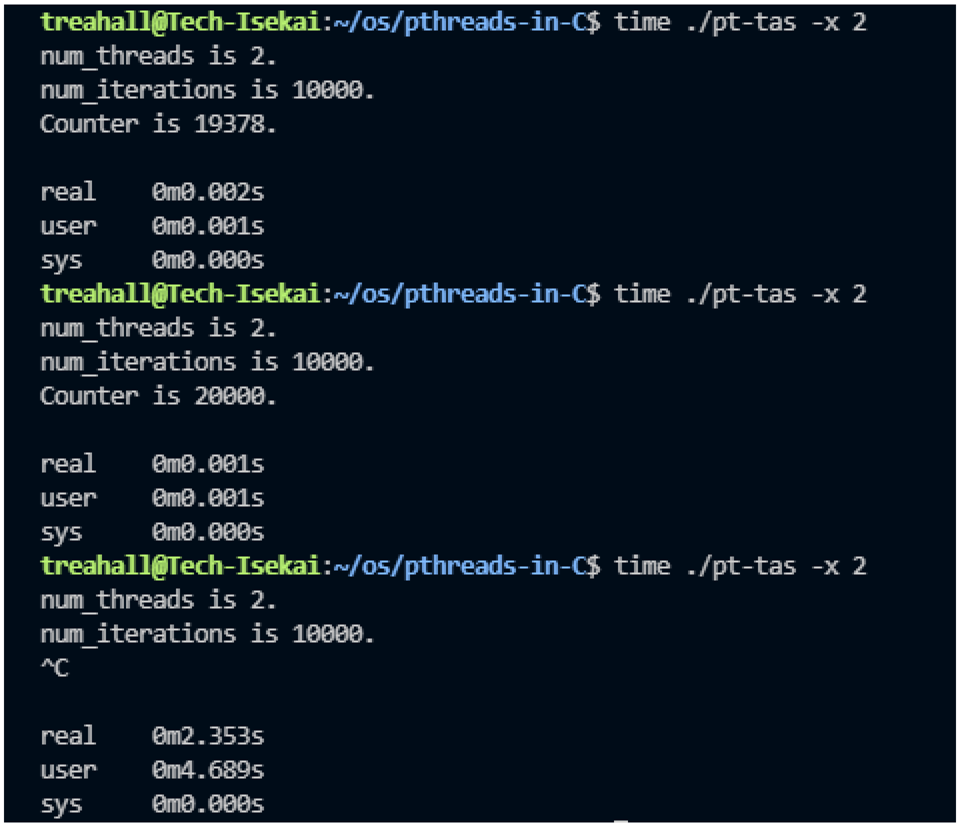
Test And Set (TAS) Locks

This is an implementation of pthreads using a spin-lock called Test and Set.

Testing Default Values

The default settings are 4 threads with 10,000 each. The code would not execute due to blocking problems cause by lock escalation. To mitigate this, I had to lower the number of threads to 2.

- Execution Time – .001 to .002 seconds in real time, and sometimes would block.
    - Note that this has half the threads as the other experiments, which impacts time.
- Final Counter Values – When it did not block, it nearly always returned the correct value. Curiously, there were a few runs that did not return the correct value.

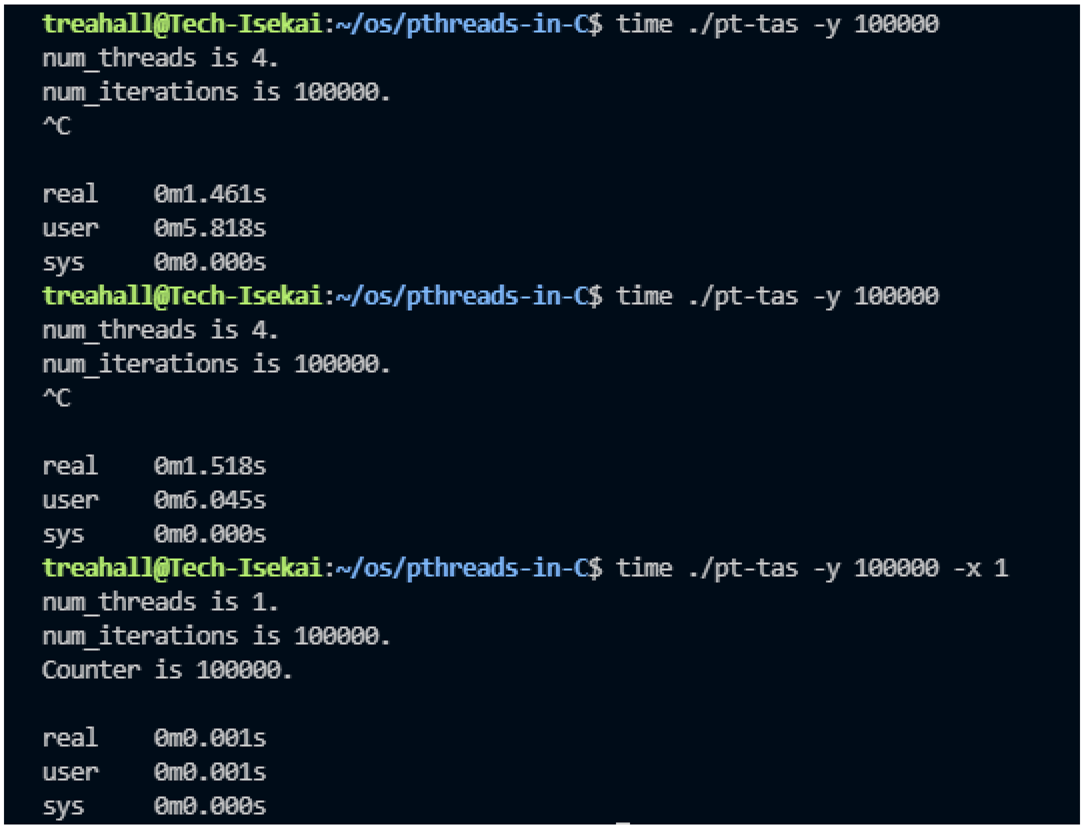


```
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-tas -x 2
num_threads is 2.
num_iterations is 10000.
Counter is 19378.

real    0m0.002s
user    0m0.001s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-tas -x 2
num_threads is 2.
num_iterations is 10000.
Counter is 20000.

real    0m0.001s
user    0m0.001s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-tas -x 2
num_threads is 2.
num_iterations is 10000.
^C

real    0m2.353s
user    0m4.689s
sys     0m0.000s
```

Testing With more Iterations

For this test I ran the code with 4 threads and 100,000 iterations. The code was incapable of a single successful run. It would only work if there was a single thread.

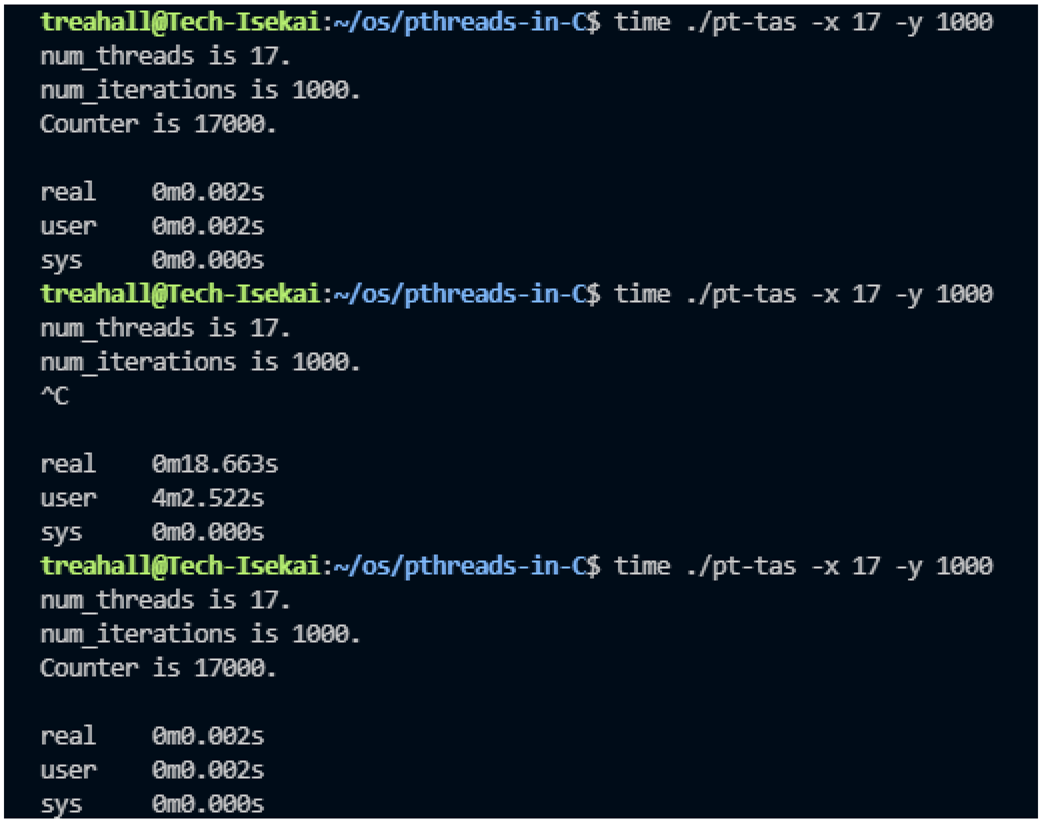
- Execution Time – Never completed, had to be killed.
- Final Counter Values – Never printed the counter.


```
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-tas -y 100000
num_threads is 4.
num_iterations is 100000.
^C

real    0m1.461s
user    0m5.818s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-tas -y 100000
num_threads is 4.
num_iterations is 100000.
^C

real    0m1.518s
user    0m6.045s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-tas -y 100000 -x 1
num_threads is 1.
num_iterations is 100000.
Counter is 100000.

real    0m0.001s
user    0m0.001s
sys     0m0.000s
```

More Threads Than CPUs

Lscpu reported my computer has 16 CPUs, so I ran this test using 17 threads. I was unable to use the default iterations due to blocking, so I lowered it to 1000 iterations. It completed execution most of the time but blocked for a few of the runs.

- Execution Time – When it did not block, it took .002 seconds in real time.
- Final Counter Values – for runs that completed, they returned the expected value.



```
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-tas -x 17 -y 1000
num_threads is 17.
num_iterations is 1000.
Counter is 17000.

real    0m0.002s
user    0m0.002s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-tas -x 17 -y 1000
num_threads is 17.
num_iterations is 1000.
^C

real    0m18.663s
user    4m2.522s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-tas -x 17 -y 1000
num_threads is 17.
num_iterations is 1000.
Counter is 17000.

real    0m0.002s
user    0m0.002s
sys     0m0.000s
```

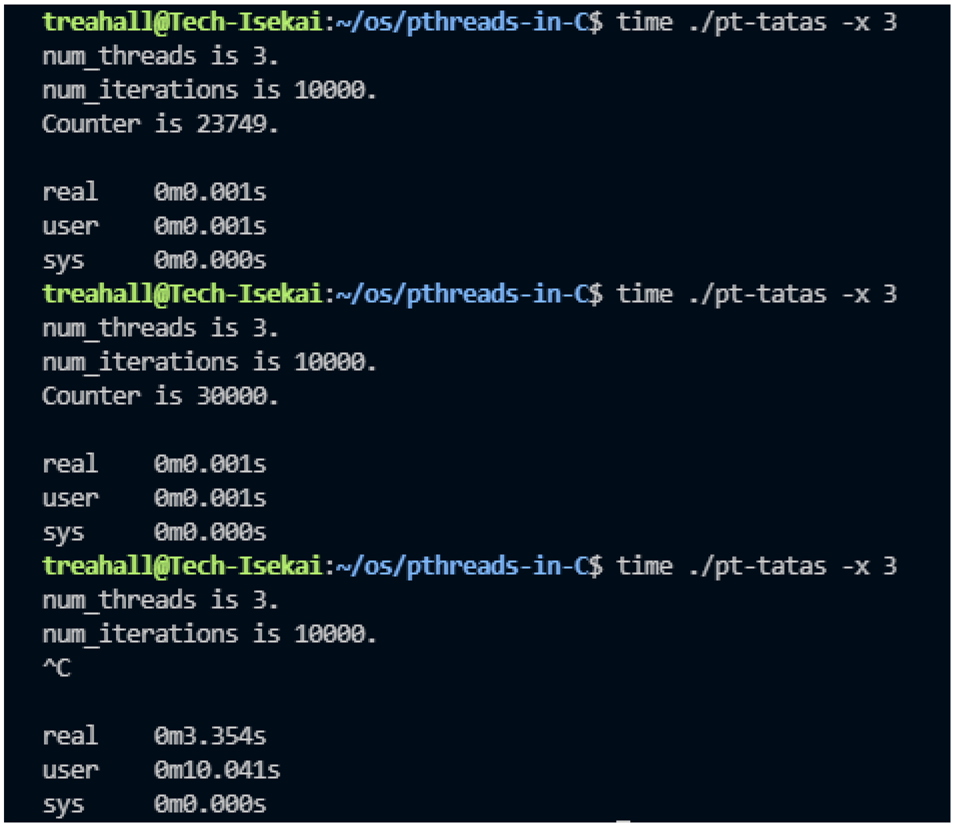
Test And Test and Set (TATAS) locks

This is an implementation of a spinlock for pthreads called Test and Test and Set. From experimenting, I found that it has many of the same problems as TAS locks but can handle slightly more lock escalation.

Testing Default Values

The default settings are 4 threads with 10,000 each. It was not capable of running 4 threads at default iterations, but it could run 3 threads at default iterations. There were still problems with occasional inaccuracy and blocking at that lock escalation.

- Execution Time – When it would not block, it took between .001 and .002 seconds in real time.
- Final Counter Values – at this level of lock escalation, it was mostly inaccurate answers. Lowering the number of iterations drastically improves rate of accuracy.

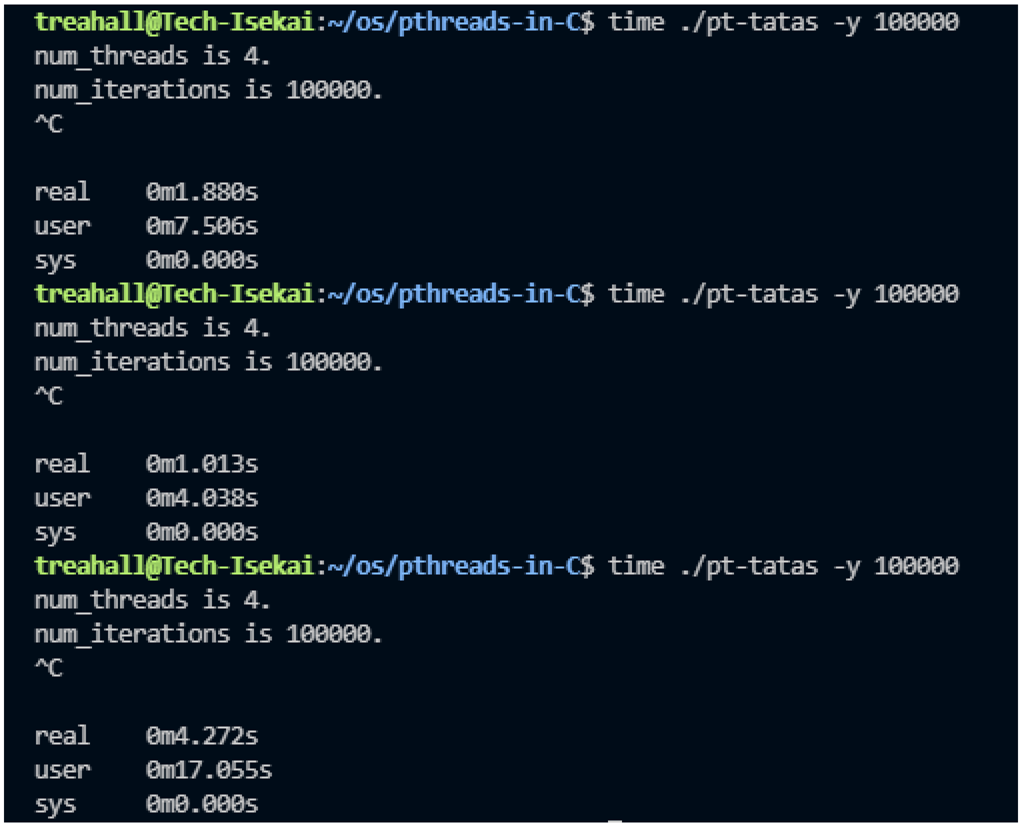


```
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-tatas -x 3
num_threads is 3.
num_iterations is 10000.
Counter is 23749.

real    0m0.001s
user    0m0.001s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-tatas -x 3
num_threads is 3.
num_iterations is 10000.
Counter is 30000.

real    0m0.001s
user    0m0.001s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-tatas -x 3
num_threads is 3.
num_iterations is 10000.
^C

real    0m3.354s
user    0m10.041s
sys     0m0.000s
```

Testing With more Iterations

For this test I ran the code with 4 threads and 100,000 iterations. Code could not run with more iterations.

- Execution Time – no data due to blocking.
- Final Counter Values – no data due to blocking.

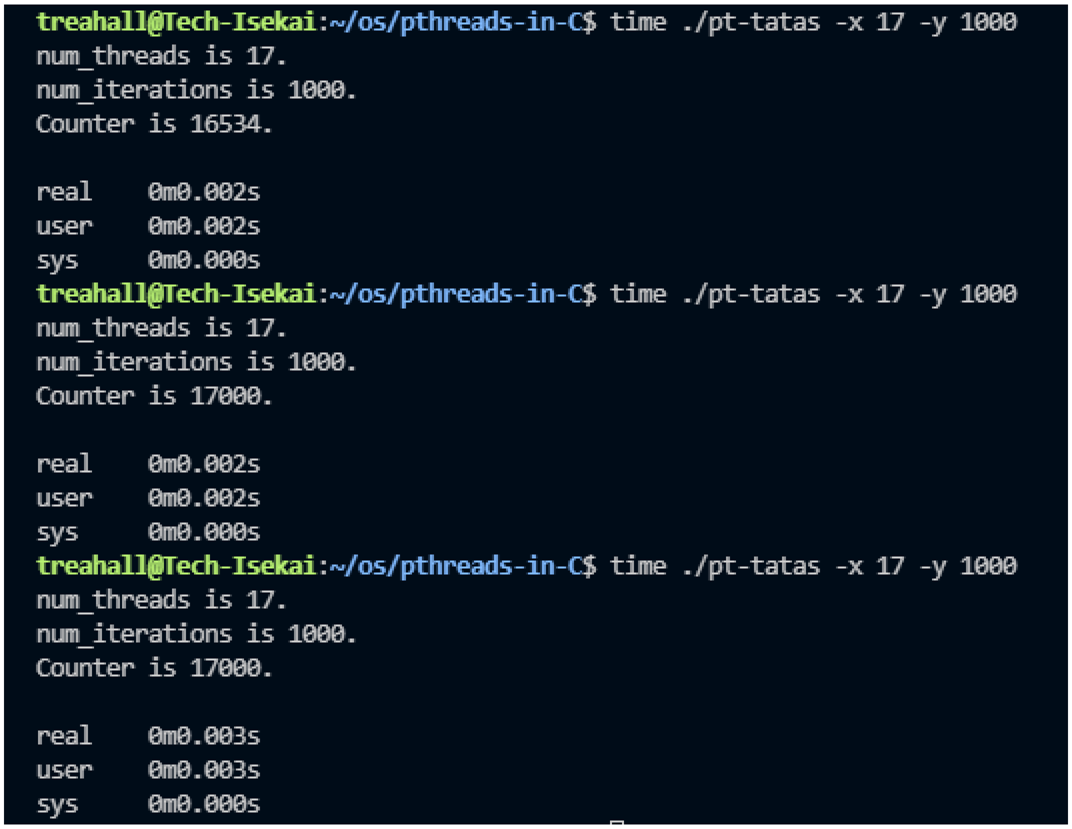


```
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-tatas -y 100000
num_threads is 4.
num_iterations is 100000.
^C

real    0m1.880s
user    0m7.506s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-tatas -y 100000
num_threads is 4.
num_iterations is 100000.
^C

real    0m1.013s
user    0m4.038s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-tatas -y 100000
num_threads is 4.
num_iterations is 100000.
^C

real    0m4.272s
user    0m17.055s
sys     0m0.000s
```

More Threads Than CPUs

Lscpu reported my computer has 16 CPUs, so I ran this test using 17 threads. Could not use the default interactions due to blocking issues, so I lowered the iterations to 1000. You can see that it is improved over TAS, because TATAS never blocked at this level.

- Execution Time – between .002 and .003 seconds in real time
- Final Counter Values – usually the expected value with few exceptions.



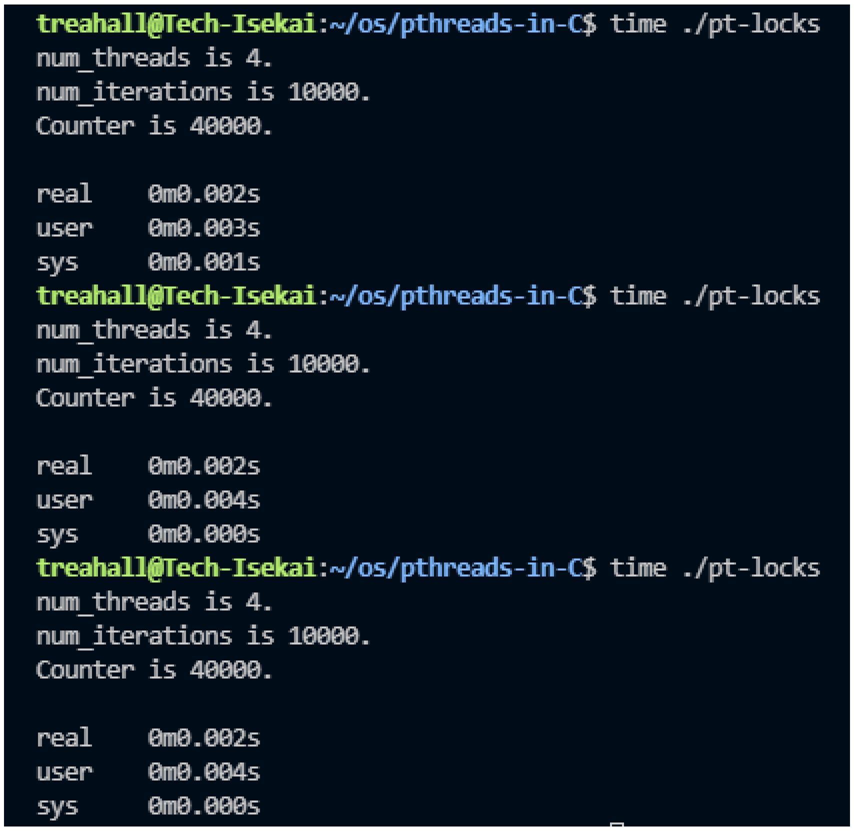
```
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-tatas -x 17 -y 1000
num_threads is 17.
num_iterations is 1000.
Counter is 16534.

real    0m0.002s
user    0m0.002s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-tatas -x 17 -y 1000
num_threads is 17.
num_iterations is 1000.
Counter is 17000.

real    0m0.002s
user    0m0.002s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-tatas -x 17 -y 1000
num_threads is 17.
num_iterations is 1000.
Counter is 17000.

real    0m0.003s
user    0m0.003s
sys     0m0.000s
```

Pthread Locks

Testing Default Values

The default settings are 4 threads with 10,000 each. It always returned and gave the expected value of counter.

- Execution Time – between .002 and .003 seconds
- Final Counter Values – always the expected value of 40000

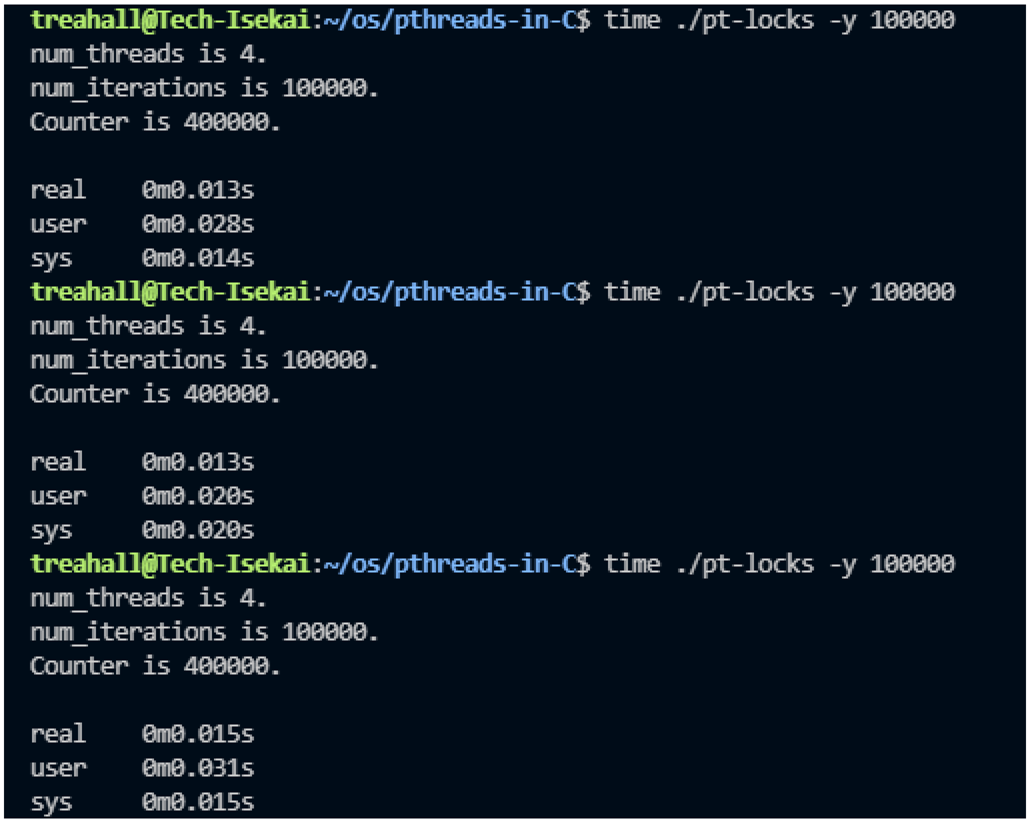


```
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-locks
num_threads is 4.
num_iterations is 10000.
Counter is 40000.

real    0m0.002s
user    0m0.003s
sys     0m0.001s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-locks
num_threads is 4.
num_iterations is 10000.
Counter is 40000.

real    0m0.002s
user    0m0.004s
sys     0m0.000s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-locks
num_threads is 4.
num_iterations is 10000.
Counter is 40000.

real    0m0.002s
user    0m0.004s
sys     0m0.000s
```

Testing With more Iterations

For this test I ran the code with 4 threads and 100,000 iterations.

- Execution Time – between .013 and .015 seconds
- Final Counter Values – always the expected value of 400,000

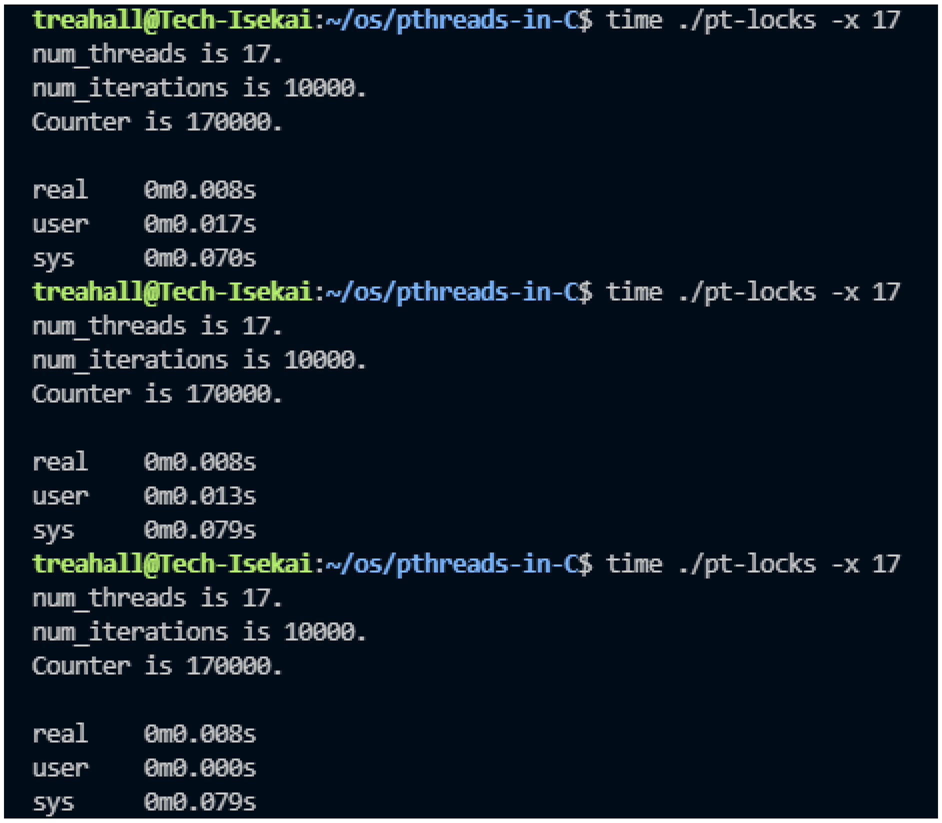


```
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-locks -y 100000
num_threads is 4.
num_iterations is 100000.
Counter is 400000.

real    0m0.013s
user    0m0.028s
sys     0m0.014s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-locks -y 100000
num_threads is 4.
num_iterations is 100000.
Counter is 400000.

real    0m0.013s
user    0m0.020s
sys     0m0.020s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-locks -y 100000
num_threads is 4.
num_iterations is 100000.
Counter is 400000.

real    0m0.015s
user    0m0.031s
sys     0m0.015s
```

More Threads Than CPUs

Lscpu reported my computer has 16 CPUs, so I ran this test using 17 threads at default iterations.

- Execution Time – it consistently took .008 seconds
- Final Counter Values – always the expected value of 170,000


```
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-locks -x 17
num_threads is 17.
num_iterations is 10000.
Counter is 170000.

real    0m0.008s
user    0m0.017s
sys     0m0.070s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-locks -x 17
num_threads is 17.
num_iterations is 10000.
Counter is 170000.

real    0m0.008s
user    0m0.013s
sys     0m0.079s
treahall@Tech-Isekai:~/os/pthreads-in-C$ time ./pt-locks -x 17
num_threads is 17.
num_iterations is 10000.
Counter is 170000.

real    0m0.008s
user    0m0.000s
sys     0m0.079s
```