

**Lucas Jensen**

**jenseluc@oregonstate.edu**

**CS472 Spring 2023**

**Project 3: Mutex Stack Challenge**

## Commentary

I ran this program on my local computer, which is running Pop!\_OS 22.04 LTS x86\_64 and used g++-12 as the compiler. Based on the lecture slides which correspond to this material, implementing mutex locks (and unsetting them) was fairly straightforward and was generally done as follows:

```
void PushAll()
{
    if (USE_MUTEX)
        omp_set_lock(&Lock);

    for (int i = 0; i < NUMN; i++)
    {
        Push(i);
    }

    if (USE_MUTEX)
        omp_unset_lock(&Lock);
}
```

PopAll was handled the same way

Within the main function, and right before defining the omp parallel sections, I also called the following in order to initialize the lock:

```
#pragma omp parallel sections
{
    #pragma omp section
    {
        PushAll();
    }
}
```

The above code was not my initial (or even second) attempt at correctly implementing mutex locks. But once I had done so successfully, the results were immediately apparent. By locking the Pop or Push function (and then releasing the lock when finished), neither thread can modify a location in memory being actively used by the other thread.

The non-mutex way does work occasionally, but it is extremely rare. I had to run the suggested bash script multiple times to find one occurrence of `USE_MUTEX = false` and an error percentage of 0.

Changing NUMN does not change the failure percentage in a meaningful way, or at least it doesn't seem like it. A low value of NUMN seemed slightly more prone to spikes (or dips) in the failure percentage, but this was not a hard rule. All values of NUMN seemed to generate an average failure percentage of around 45%, and all values had outliers. Anecdotally, with only a handful of runs of the bash script, the higher the value of NUMN, the more "settled" the failure percentage was.

There is certainly a difference in execution time between mutex and non-mutex runs. Mutex runs are fairly consistent at 40-60 microseconds, whereas non-mutex runs are just all over the place. Some non-mutex runs also land at about 40-60 microseconds, but most are on the order of hundreds of microseconds. Some go well into the thousands or hundreds of thousands of microseconds. With such a huge range of data, I'm unsure of what is considered "very large" and therefore rightfully discarded due to TIMEOUT.

I'm not entirely sure why this is. A naive guess before running this program would have had me speculate that a non-mutex run would actually be faster but significantly less accurate due to each thread just blindly doing whatever it wants, whenever it wants. However, after playing around with the program, my best guess would be that chaos generally ensues when two threads access the same location in memory. Beyond the logic behind a stack breaking down and failures occurring, each thread may run into significant performance issues when trying to read the same byte of data as it's being manipulated by another thread. One would think that the non-mutex run would be faster because the threads don't have to wait for one another, but the data doesn't show this to be the case.

## Raw Data

```
NUMN = 1024 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 31.01 microseconds
NUMN = 1024 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 66.01 microseconds
NUMN = 1024 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 63.02 microseconds
NUMN = 1024 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 66.42 microseconds
NUMN = 1024 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 62.06 microseconds
NUMN = 1024 , USE_MUTEX = false , NumPopErrors = 673 = 65.72% , Elapsed time = 66665.58 microseconds
NUMN = 1024 , USE_MUTEX = false , NumPopErrors = 627 = 61.23% , Elapsed time = 60025.61 microseconds
NUMN = 1024 , USE_MUTEX = false , NumPopErrors = 332 = 32.42% , Elapsed time = 76.73 microseconds
NUMN = 1024 , USE_MUTEX = false , NumPopErrors = 361 = 35.25% , Elapsed time = 70.38 microseconds
NUMN = 1024 , USE_MUTEX = false , NumPopErrors = 433 = 42.29% , Elapsed time = 75.19 microseconds
NUMN = 1024 , USE_MUTEX = false , NumPopErrors = 545 = 53.22% , Elapsed time = 76.40 microseconds
NUMN = 1024 , USE_MUTEX = false , NumPopErrors = 312 = 30.47% , Elapsed time = 78.99 microseconds
NUMN = 1024 , USE_MUTEX = false , NumPopErrors = 326 = 31.84% , Elapsed time = 72.04 microseconds
NUMN = 1024 , USE_MUTEX = false , NumPopErrors = 432 = 42.19% , Elapsed time = 76.95 microseconds
NUMN = 1024 , USE_MUTEX = false , NumPopErrors = 304 = 29.69% , Elapsed time = 78.05 microseconds

NUMN = 2048 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 70.56 microseconds
NUMN = 2048 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 69.49 microseconds
NUMN = 2048 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 71.31 microseconds
NUMN = 2048 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 36.01 microseconds
NUMN = 2048 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 69.53 microseconds
NUMN = 2048 , USE_MUTEX = false , NumPopErrors = 635 = 31.01% , Elapsed time = 88.84 microseconds
NUMN = 2048 , USE_MUTEX = false , NumPopErrors = 976 = 47.66% , Elapsed time = 96.75 microseconds
NUMN = 2048 , USE_MUTEX = false , NumPopErrors = 1072 = 52.34% , Elapsed time = 82.04 microseconds
NUMN = 2048 , USE_MUTEX = false , NumPopErrors = 610 = 29.79% , Elapsed time = 83.29 microseconds
NUMN = 2048 , USE_MUTEX = false , NumPopErrors = 895 = 43.70% , Elapsed time = 98.52 microseconds
NUMN = 2048 , USE_MUTEX = false , NumPopErrors = 795 = 38.82% , Elapsed time = 93.73 microseconds
NUMN = 2048 , USE_MUTEX = false , NumPopErrors = 1358 = 66.31% , Elapsed time = 108201.25 microseconds
NUMN = 2048 , USE_MUTEX = false , NumPopErrors = 887 = 43.31% , Elapsed time = 97.01 microseconds
NUMN = 2048 , USE_MUTEX = false , NumPopErrors = 860 = 41.99% , Elapsed time = 97.47 microseconds
NUMN = 2048 , USE_MUTEX = false , NumPopErrors = 847 = 41.36% , Elapsed time = 92.76 microseconds

NUMN = 4096 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 76.91 microseconds
NUMN = 4096 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 77.92 microseconds
NUMN = 4096 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 80.31 microseconds
NUMN = 4096 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 77.17 microseconds
NUMN = 4096 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 79.48 microseconds
NUMN = 4096 , USE_MUTEX = false , NumPopErrors = 1750 = 42.72% , Elapsed time = 133.05 microseconds
NUMN = 4096 , USE_MUTEX = false , NumPopErrors = 1575 = 38.45% , Elapsed time = 123.85 microseconds
NUMN = 4096 , USE_MUTEX = false , NumPopErrors = 1685 = 41.14% , Elapsed time = 122.35 microseconds
NUMN = 4096 , USE_MUTEX = false , NumPopErrors = 2044 = 49.90% , Elapsed time = 136250.71 microseconds
NUMN = 4096 , USE_MUTEX = false , NumPopErrors = 1742 = 42.53% , Elapsed time = 127.36 microseconds
NUMN = 4096 , USE_MUTEX = false , NumPopErrors = 1912 = 46.68% , Elapsed time = 99.27 microseconds
NUMN = 4096 , USE_MUTEX = false , NumPopErrors = 1563 = 38.16% , Elapsed time = 128.56 microseconds
NUMN = 4096 , USE_MUTEX = false , NumPopErrors = 1982 = 48.39% , Elapsed time = 57735.02 microseconds
NUMN = 4096 , USE_MUTEX = false , NumPopErrors = 1708 = 41.70% , Elapsed time = 102.52 microseconds
NUMN = 4096 , USE_MUTEX = false , NumPopErrors = 1827 = 44.60% , Elapsed time = 1952.81 microseconds

NUMN = 8192 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 99.12 microseconds
NUMN = 8192 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 103.60 microseconds
NUMN = 8192 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 102.47 microseconds
NUMN = 8192 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 83.24 microseconds
NUMN = 8192 , USE_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 102.95 microseconds
NUMN = 8192 , USE_MUTEX = false , NumPopErrors = 3797 = 46.35% , Elapsed time = 194.70 microseconds
NUMN = 8192 , USE_MUTEX = false , NumPopErrors = 4732 = 57.76% , Elapsed time = 154.46 microseconds
NUMN = 8192 , USE_MUTEX = false , NumPopErrors = 3960 = 48.34% , Elapsed time = 288936.82 microseconds
NUMN = 8192 , USE_MUTEX = false , NumPopErrors = 2089 = 25.50% , Elapsed time = 169.55 microseconds
NUMN = 8192 , USE_MUTEX = false , NumPopErrors = 3813 = 46.55% , Elapsed time = 46473.41 microseconds
```

NUMN = 8192 , USE\_MUTEX = false , NumPopErrors = 3962 = 48.36% , Elapsed time = 112952.39 microseconds  
NUMN = 8192 , USE\_MUTEX = false , NumPopErrors = 3566 = 43.53% , Elapsed time = 155.24 microseconds  
NUMN = 8192 , USE\_MUTEX = false , NumPopErrors = 3717 = 45.37% , Elapsed time = 68899.21 microseconds  
NUMN = 8192 , USE\_MUTEX = false , NumPopErrors = 3938 = 48.07% , Elapsed time = 206.86 microseconds  
NUMN = 8192 , USE\_MUTEX = false , NumPopErrors = 3137 = 38.29% , Elapsed time = 226.37 microseconds

NUMN = 16384 , USE\_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 145.12 microseconds  
NUMN = 16384 , USE\_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 144.88 microseconds  
NUMN = 16384 , USE\_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 143.10 microseconds  
NUMN = 16384 , USE\_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 143.67 microseconds  
NUMN = 16384 , USE\_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 141.10 microseconds  
NUMN = 16384 , USE\_MUTEX = false , NumPopErrors = 10685 = 65.22% , Elapsed time = 1072182.50 microseconds  
NUMN = 16384 , USE\_MUTEX = false , NumPopErrors = 8604 = 52.51% , Elapsed time = 730137.48 microseconds  
NUMN = 16384 , USE\_MUTEX = false , NumPopErrors = 5963 = 36.40% , Elapsed time = 342.38 microseconds  
NUMN = 16384 , USE\_MUTEX = false , NumPopErrors = 7333 = 44.76% , Elapsed time = 365.97 microseconds  
NUMN = 16384 , USE\_MUTEX = false , NumPopErrors = 8170 = 49.87% , Elapsed time = 183436.35 microseconds  
NUMN = 16384 , USE\_MUTEX = false , NumPopErrors = 7700 = 47.00% , Elapsed time = 326.52 microseconds  
NUMN = 16384 , USE\_MUTEX = false , NumPopErrors = 6760 = 41.26% , Elapsed time = 306.58 microseconds  
NUMN = 16384 , USE\_MUTEX = false , NumPopErrors = 7375 = 45.01% , Elapsed time = 334.16 microseconds  
NUMN = 16384 , USE\_MUTEX = false , NumPopErrors = 6160 = 37.60% , Elapsed time = 339.78 microseconds  
NUMN = 16384 , USE\_MUTEX = false , NumPopErrors = 10740 = 65.55% , Elapsed time = 214.09 microseconds

NUMN = 32768 , USE\_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 219.59 microseconds  
NUMN = 32768 , USE\_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 227.35 microseconds  
NUMN = 32768 , USE\_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 230.54 microseconds  
NUMN = 32768 , USE\_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 225.22 microseconds  
NUMN = 32768 , USE\_MUTEX = true , NumPopErrors = 0 = 0.00% , Elapsed time = 222.81 microseconds  
NUMN = 32768 , USE\_MUTEX = false , NumPopErrors = 16920 = 51.64% , Elapsed time = 601008.50 microseconds  
NUMN = 32768 , USE\_MUTEX = false , NumPopErrors = 14882 = 45.42% , Elapsed time = 93312.63 microseconds  
NUMN = 32768 , USE\_MUTEX = false , NumPopErrors = 16545 = 50.49% , Elapsed time = 546618.72 microseconds  
NUMN = 32768 , USE\_MUTEX = false , NumPopErrors = 14482 = 44.20% , Elapsed time = 737.63 microseconds  
NUMN = 32768 , USE\_MUTEX = false , NumPopErrors = 16125 = 49.21% , Elapsed time = 334503.91 microseconds  
NUMN = 32768 , USE\_MUTEX = false , NumPopErrors = 16503 = 50.36% , Elapsed time = 359646.22 microseconds  
NUMN = 32768 , USE\_MUTEX = false , NumPopErrors = 14901 = 45.47% , Elapsed time = 592.18 microseconds  
NUMN = 32768 , USE\_MUTEX = false , NumPopErrors = 17085 = 52.14% , Elapsed time = 510.05 microseconds  
NUMN = 32768 , USE\_MUTEX = false , NumPopErrors = 18395 = 56.14% , Elapsed time = 578.77 microseconds  
NUMN = 32768 , USE\_MUTEX = false , NumPopErrors = 15332 = 46.79% , Elapsed time = 191517.71 microseconds