

Project 03

1. Tell what machine you ran this on, what operating system you were using and what compiler you used.

I ran this program on the flip server, through the terminal.

2. Include, in your writeup, the pieces of code where you implemented the mutexes

```
int
main( int argc, char *argv[ ] )
{
#ifdef _OPENMP
    fprintf( stderr, "OpenMP is not supported here.\n" );
    return 1;
#endif
    //inilizing Lock here
    omp_init_lock(&Lock);

    if (USE_MUTEX) {
        omp_set_lock(&Lock);
    }
    for( int i = 0; i < NUMN; i++ )
    {
        Push( i );
    }
    if (USE_MUTEX) {
        omp_unset_lock(&Lock);
    }
}
```

I initialized `omp_init_lock()` in the main function. Then I used mutex in all pop and all push like seen above.

3. Tell us what you discovered by doing this:
 - a. Does the non-mutex way of doing this *ever* work? If so, how often?

The non-mutex never works and has a high number of pop errors. Even after running the code multiple times with different values, I never found a case that it worked.

- b. Does changing NUMN make any difference in the failure percentage?

The number of NUMN has a slight difference in percentage failure but it is nothing very extreme. The more NUMU the higher the failure rate is when there is no mutex. However with mutex there was no difference in percent failure no matter how many NUMN there were(it was all consistently zero).

- c. Is there a difference in elapsed execution time between mutex and non-mutex? Why do you suppose this is? (Ignore the very large elapsed times -- these are a result of the TIMEOUT being used up.)

With less NUMU it is hard to tell the difference in elapsed time but once you get to 16,384 it is very easy to tell the difference. For mutex the time is around 253-304 but for non mutex it is about 702 - 6959 micro seconds. The high end of the mutex range(304) does not come close to the low end of the non mutex(702). This pattern continues when looking at the 32,768 NUMU data.