CS 475/575 -- Spring Quarter 2023

Project #3 - The Mutex Stack Challenge

Submitted by

Rahul Kumar Nalubandhu

Onid: nalubanr

Email : nalubanr@oregonstate.edu

1. Tell what machine you ran this on.

   I ran this on Rabbit server.

2. Tell what operating system you were using.

   I was using Linux OS on Rabbit server.

3. Tell what compiler you used.

   I used GNU Compiler Collection (GCC)

   g++ mutex03.cpp -o mutex03 -lm -fopenmp

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

```
int Stack[NUMN];
volatile int StackPtr = -1;
// ok to set the StackPtr to an illegal array index since we will never actually
//  because we will increment it before Pushing with it
// to push: increment stack pointer, then place number there
// to pop:  take number from there, then decrement stack pointer

bool WasPopped[NUMN];
int NumPopErrors;

omp_lock_t Lock;

void Push(int n)
{
    if (USE_MUTEX)
    {
        omp_set_lock(&Lock); //
    }
    StackPtr++;
    Stack[StackPtr] = n;
    if (USE_MUTEX)
    {
        omp_unset_lock(&Lock);//
    }
}
```

```
int Pop()
{
    // if the stack is empty, give the Push( ) function a chance to
    int t = 0;
    while (StackPtr < 0 && t < TIMEOUT)
        t++;

    // if there is nothing to pop, return;
    if (StackPtr < 0)
        return FAILED;

    if (USE_MUTEX)
    {
        omp_set_lock(&Lock); //
    }

    int n = Stack[StackPtr];
    StackPtr--;

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

```
int main(int argc, char *argv[])
{
#ifndef _OPENMP
    fprintf(stderr, "OpenMP is not supported here.\n");
    return 1;
#endif
    omp_init_lock(&Lock); //
    // this array is here to be sure all the pops actually happen
    for (int i = 0; i < NUMN; i++)
    {
        WasPopped[i] = false;
    }
```

5. Tell us what you discovered by doing this:

    1. Does the non-mutex way of doing this *ever* work? If so, how often?

        The non-mutex way of handling this operation can work, but it is unpredictable and prone to errors due to given conditions. In this data, we see that the non-mutex approach results in varying amounts of NumPopErrors, ranging from 11.38% to as high as 98.56%. These errors occur when multiple threads access shared data simultaneously without proper synchronization, leading to unexpected results.

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

        Yes, changing NUMN can make a difference in the NumPopErrors percentage. Suppose If the mutex is false then, increasing NUMN can lead, to in an increase in the NumPopErrors percentage. Conversely, reducing NUMN decreases the NumPopErrors percentage.

        In the case for mutex = true, the NumPopErrors percentage remains at 0% regardless of the value of NUMN.

    3. 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.)

        Yes, there is a time difference between using a mutex = true and not mutex = false.
        When you use a mutex, the program runs slower because the threads need to wait for their turn to access shared data. This prevents errors from happening.

        When you don't use a mutex, the program runs faster because there is no waiting. However, this can cause errors since multiple threads might change the shared data at the same time.

        So, using a mutex makes the program slower but safer, while not using a mutex makes it faster but riskier.

Below is Variety of NUMN values with mutexes in use and mutexes not in use. I just consider few NUMN values to display here.

```
rabbit ~/cs575/Project_3 1001$ g++ mutex03.cpp -o mutex03 -lm -fopenmp
rabbit ~/cs575/Project_3 1002$ ./mutex03
NUMN =  32768 , USE_MUTEX =  true , NumPopErrors =     0 =   0.00% , Elapsed time =   8464.74 microseconds
rabbit ~/cs575/Project_3 1003$ g++ mutex03.cpp -o mutex03 -lm -fopenmp
rabbit ~/cs575/Project_3 1004$ ./mutex03
NUMN =  32768 , USE_MUTEX = false , NumPopErrors = 11016 =  33.62% , Elapsed time =   1449.91 microseconds
rabbit ~/cs575/Project_3 1005$
```

NUMN = 1024	USE_MUTEX = true	NumPopErrors = 0 = 0.00%	Elapsed time = 547.44 microseconds
NUMN = 1024	USE_MUTEX = true	NumPopErrors = 0 = 0.00%	Elapsed time = 360.92 microseconds
NUMN = 1024	USE_MUTEX = true	NumPopErrors = 0 = 0.00%	Elapsed time = 253.85 microseconds
NUMN = 1024	USE_MUTEX = true	NumPopErrors = 0 = 0.00%	Elapsed time = 337.35 microseconds
NUMN = 1024	USE_MUTEX = true	NumPopErrors = 0 = 0.00%	Elapsed time = 384.09 microseconds
NUMN = 1024	USE_MUTEX = false	NumPopErrors = 298 = 29.10%	Elapsed time = 183.56 microseconds
NUMN = 1024	USE_MUTEX = false	NumPopErrors = 276 = 26.95%	Elapsed time = 160.35 microseconds
NUMN = 1024	USE_MUTEX = false	NumPopErrors = 357 = 34.86%	Elapsed time = 128.08 microseconds
NUMN = 1024	USE_MUTEX = false	NumPopErrors = 342 = 33.40%	Elapsed time = 183.48 microseconds
NUMN = 1024	USE_MUTEX = false	NumPopErrors = 247 = 24.12%	Elapsed time = 139.15 microseconds
NUMN = 1024	USE_MUTEX = false	NumPopErrors = 273 = 26.66%	Elapsed time = 165.50 microseconds
NUMN = 1024	USE_MUTEX = false	NumPopErrors = 328 = 32.03%	Elapsed time = 98.89 microseconds
NUMN = 1024	USE_MUTEX = false	NumPopErrors = 287 = 28.03%	Elapsed time = 192.82 microseconds
NUMN = 1024	USE_MUTEX = false	NumPopErrors = 542 = 52.93%	Elapsed time = 131.10 microseconds
NUMN = 1024	USE_MUTEX = false	NumPopErrors = 271 = 26.46%	Elapsed time = 119.32 microseconds
NUMN = 2048	USE_MUTEX = true	NumPopErrors = 0 = 0.00%	Elapsed time = 831.41 microseconds

NUMN = 2048	USE_MUTEX = true	NumPopErrors = 0 = 0.00%	Elapsed time = 600.46 microseconds
NUMN = 2048	USE_MUTEX = true	NumPopErrors = 0 = 0.00%	Elapsed time = 585.31 microseconds
NUMN = 2048	USE_MUTEX = true	NumPopErrors = 0 = 0.00%	Elapsed time = 663.54 microseconds
NUMN = 2048	USE_MUTEX = true	NumPopErrors = 0 = 0.00%	Elapsed time = 1088.00 microseconds
NUMN = 2048	USE_MUTEX = false	NumPopErrors = 1047 = 51.12%	Elapsed time = 248.99 microseconds
NUMN = 2048	USE_MUTEX = false	NumPopErrors = 697 = 34.03%	Elapsed time = 168.28 microseconds
NUMN = 2048	USE_MUTEX = false	NumPopErrors = 519 = 25.34%	Elapsed time = 203.34 microseconds
NUMN = 2048	USE_MUTEX = false	NumPopErrors = 1068 = 52.15%	Elapsed time = 265.39 microseconds
NUMN = 2048	USE_MUTEX = false	NumPopErrors = 617 = 30.13%	Elapsed time = 124.17 microseconds
NUMN = 2048	USE_MUTEX = false	NumPopErrors = 721 = 35.21%	Elapsed time = 171.81 microseconds
NUMN = 2048	USE_MUTEX = false	NumPopErrors = 233 = 11.38%	Elapsed time = 156.91 microseconds
NUMN = 2048	USE_MUTEX = false	NumPopErrors = 728 = 35.55%	Elapsed time = 148.30 microseconds
NUMN = 2048	USE_MUTEX = false	NumPopErrors = 1034 = 50.49%	Elapsed time = 280.81 microseconds
NUMN = 2048	USE_MUTEX = false	NumPopErrors = 463 = 22.61%	Elapsed time = 215.33 microseconds
NUMN = 4096	USE_MUTEX = true	NumPopErrors = 0 = 0.00%	Elapsed time = 1297.98 microseconds
NUMN = 4096	USE_MUTEX = true	NumPopErrors = 0 = 0.00%	Elapsed time = 1142.01 microseconds
NUMN = 4096	USE_MUTEX = true	NumPopErrors = 0 = 0.00%	Elapsed time = 1180.16 microseconds
NUMN = 4096	USE_MUTEX = true	NumPopErrors = 0 = 0.00%	Elapsed time = 1181.17 microseconds

NUMN = 4096	USE_MUTEX = true	NumPopErrors = 0 = 0.00%	Elapsed time = 1125.73 microseconds
NUMN = 4096	USE_MUTEX = false	NumPopErrors = 892 = 21.78%	Elapsed time = 351.99 microseconds
NUMN = 4096	USE_MUTEX = false	NumPopErrors = 1437 = 35.08%	Elapsed time = 524.92 microseconds
NUMN = 4096	USE_MUTEX = false	NumPopErrors = 1259 = 30.74%	Elapsed time = 411.14 microseconds
NUMN = 4096	USE_MUTEX = false	NumPopErrors = 1235 = 30.15%	Elapsed time = 373.85 microseconds
NUMN = 4096	USE_MUTEX = false	NumPopErrors = 1315 = 32.10%	Elapsed time = 259.72 microseconds