**Lab 4 QnA**

(Demonstrate your understanding of the threaded C++ code)

The general outline is the program generates mapper threads to concurrently read the files, searching and counting the target word. Once these threads have recorded their results, the individual data from each mapper thread is reduced (through reducers) into a single answer.

1. How many critical sections are in the Mapper threads (void map)?   
   **Answer**: Two. A critical section is a section of code that must be executed serially
   1. Identify the critical section areas of code (copy and paste below)

// Get work

    { // *Critical Sections ↓*

      unique\_lock<mutex> lck(fq->m);

      if(fq->filenames.size() > 0)

      {

        fname = fq->filenames.front();

        fq->filenames.pop\_front();

      }

      else // nothing in the work queue

      {

        return;

      }

    } // *Critical Sections ↑*

// lck destroyed here

// Store result

    {// *Critical Sections ↓*

      unique\_lock<mutex> lck(cq->m);

      cq->counts.push\_back(count);

      cq->who.push\_back(this\_thread::get\_id());

    }// *Critical Sections ↑*

Between the braces are the *critical sections.*

1. Why is there no mutex lock protecting the function to populate the fq object with file names? (code is copied here)

//setup some sample files for the threads to read...

//(to give it "enough" data, files are processed multiple times :)

for(int i = 0; i < 10; i++)

{

fq.filenames.push\_back("modest-proposal.txt");

fq.filenames.push\_back("flatland.txt");

}

**Answer:** Multi-threads are not running here. When code executes at this block, main thread are running. Which means there are no more threads and need no protection.

1. The thread join function utilizes two features you probably have not seen in C++. Identify these two items in the for loop code below.

//make sure all threads have finished work and join main before continuing

for(auto& m : mappers)

{

m.join();

}

**Answer:** 1. A thread which calls a {@code join()} of other thread, will pause until the other thread is over. In this case, main thread calls the {@code join()} of {@code mappers} thread, main thread will pause until the sub-thread, which is mapper thread, is finished. 2. {@code auto} is a new feature offered by C++ 11. {@code auto} automatic

1. Will searching for “The” and “the” return different results? If so, how could the program be changed to provide the same results? (Just explain in general terms, you don’t have to provide any code!)

**Answer:** Use {@code tolower()} to change {@code word} to lower case then compare to {@code needle} then we can get same results, which is the number of “The” and “the”.