An addendum to Programming Assignment 3.

The following details are also required portions of assignment 3.

These 2 changes may sound like more work than they actually are - but if you have already completed the assignment with 1 thread per file, this addendum is quite straightforward and great coding practice.

1.) Do not use global variables (worth 5 points of your grade) - instead pass parameters you need in via a struct to the pthread_create function.

2.) There seems to be some confusion regarding proper requester thread behavior, so to clarify, we will assign this expectation:

You specify the number of requester threads your program should run with via a command line argument. Furthermore, this is important as at the end when you are finished, the python script will be used to evaluate performance with respect to how the number of both requester and resolver threads changes. In order to have multiple requester threads, assign them in a round robin fashion with the following functionality. If you have 5 threads and 5 files, each thread is to begin by servicing a single file, when it finishes its file, it will go on to the next file and aid the current requester thread there. If a thread becomes free/finishes working on its current file, it will move to the next file which was the LAST file to receive a new thread.

A few examples to help illustrate the concept:

8 threads, 5 files. Assign T1 to F1, T2 to F2, ..., T5 to F5, T6 to F1, T7 to F2, T8 to F3..... if F2 finishes (threads T2 and T7) will then be assigned to F4 and F5, you will continue passing completed threads around in a round robin fashion to incomplete processed files.

This should make for an effective use of the many threads and you should see performance scale up similarly to how performance scales up when you increase the resolver thread count (up to a certain point of course). Although you may have multiple readers to a file - you will not necessarily want this - and should handle/avoid duplicates in a proper fashion.

This will be discussed further in recitations and an outline for how you might approach this will be given.