PA3 Submission Instructions

When submitting PA3 files please do so in the following manner:

Make a folder named **firstname_lastname_PA3.**

example: brennan_mcconnell_PA3

Inside of this folder, place all .c and .h files needed to run your program. Additionally, include the Makefile and the input directory which should contain the 5 input files. (The input directory should be inside of your firstname_lastname_PA3 folder).

Name your main .c file that you created ‘multi-lookup.c’ (if you have a .h file, name this multi-lookup.h), and via the makefile this should generate an executable named ‘multi-lookup’.

NOTE* You will only include files that your multi-lookup.c needs to compile. For example, you will include util.c and util.h because those are needed by multi-lookup.c to compile. You will not include files such as lookup.c or queueTest.c, etc, because these files are not needed to compile your multi-lookup. You should make modifications to the Makefile prior to submitting that remove any of these files you will not submit. For example, don't try to compile lookup.o inside the Makefile when you will not be submitting lookup.c.

Do not submit the executable. Do not submit an output file with your results (we will generate this when we test). Do not submit .o files etc. (you should run make clean to clean up any object files and executables prior to submitting).

After you have this folder with its necessary contents, zip this folder and submit it. (You are not zipping all of these files together, your are putting all of the files in a folder and then zipping said folder).

When grading, we should be able to unzip your folder, cd into the folder, run ‘make’ and run ‘./multi-lookup input/* output.txt’ to see your results.

Notes for the assignment:

You must comment your code, be detailed yet concise! You will lose points if you do not use comments within your code. Explain your code with these comments so that when we are grading, we know that you understand your own solution. Your comments should explain your design decisions, how you approached providing synchronization via different mutexes, etc. Use your best judgment. Code cleanliness is very important for this PA.

Lastly, you should have some type of output similar to the following when running your executable…

```
MacBook-Pro:OS_PA3 Brennan$ ./multi-lookup input/* output.txt
Requester thread added 19 hostnames to queue.
Requester thread added 20 hostnames to queue.
Requester thread added 20 hostnames to queue.
Requester thread added 20 hostnames to queue.
Requester thread added 23 hostnames to queue.
Error looking up Address: nodename nor servname provided, or not known
dnslookup error: sdjjdsaf.com
Resolver thread processed 18 hostnames from queue.
Resolver thread processed 3 hostnames from queue.
Resolver thread processed 16 hostnames from queue.
Resolver thread processed 1 hostnames from queue.
Resolver thread processed 8 hostnames from queue.
Resolver thread processed 32 hostnames from queue.
Resolver thread processed 14 hostnames from queue.
Resolver thread processed 10 hostnames from queue.
```

Where we have information regarding how many ‘payloads/hostnames’ were processed by each thread. Your requester threads will simply be processing the amount of hostnames that are in the corresponding input file. Your resolver threads will be processing a different amount of hostnames each time it is ran.

This is very easy to accomplish, simply add a counter to each thread that increments when it adds/pops to the queue. And at the end of the threads life, print the count.