

## Assignment 4 – Word Blast

### Description:

Word blast is supposed to show us how to multithread to divide and conquer a task without overwriting data of critical sections. We are supposed to demonstrate this by decreasing time spent condign a specific condition (in this case if a word is 6 or more characters.). Then tally it in ascending order by displaying the top 10.

### Approach:

In the example of the read.me, we see the task being evenly split of reading the book in 5 sections. The case was that if you specified 5 threads like you said in the prompt, they will each have a fraction of the work (I.e. 1/5 here). Because our desired result tests 1, 2, 4 and then 8 threads, I am assuming that space and time complexity is exponential based on how we loop our arrays. As for the mutex lock, I will use the lock / unlock to ensure that only one thread can modify a counter at one time. I also think for my control flow I can use an if statement is based on the parameters passed. That way we can make the threads on execution of the program during run time. After thinking about it for some time, I came to the conclusion I would use arrays to store and sort through. I think struct might work but have more familiarity with rules of an array. Indexing also seems to be a factor which we covered in data structures.

### Issues and Resolutions:

The biggest wrongdoing I did to myself in this assignment was using strtol instead of the atoi function. Since I knew the data type that was passed through as. Command argument, I would not have had to specify the specific parameters I did using strtol. After I asked my classmates what they were doing, I figured that maybe I should have researched more my needs for the passing.

**Analysis:** I think the big take was from this assignment is time complexity. When we learned about what data is looping, you want to minimize it so the program has more resources. In my program, I have the critical section of data between the mutex locks. I tried not to hav anything else run with an iteration other than that. Since that was how we sorted a big data, I figured that this is why in class you were asking us about how many goole searches happen a data. When working with data that needs to be analyzed in subsets, it is wise for us to not step on the shoes of the other threads. This way our results were accurate and we reap the benefits of multitasking without any risk. I was running out of time, so I had to upload what I could in order to start HW 5 and finish it on a reasonable time. I would say this assignment was the one I learned most about my decision making in this class.

Screen shot of compilation:

```
student@student:~/dev/csc-415-assignment-4-word-blast-austinlikesfood$ make clean
rm *.o Kuykendall_Austin_HW4_main
student@student:~/dev/csc-415-assignment-4-word-blast-austinlikesfood$ make run
gcc -c -o Kuykendall_Austin_HW4_main.o Kuykendall_Austin_HW4_main.c -g -I.
Kuykendall_Austin_HW4_main.c:38:5: warning: built-in function 'remainder' declared as non-function [-Wbuiltin-declaration-mismatch]
   38 | int remainder = 0;
      |           ^~~~~~
```

Screen shot(s) of the execution of the program:

```
gcc -o Kuykendall_Austin_HW4_main Kuykendall_Austin_HW4_main.o -g -I. -l pthread
./Kuykendall_Austin_HW4_main WarAndPeace.txt 2
Word Frequency Count on WarAndPeace.txt with 2 threads
Printing top 10 words 6 character or more.
Number 1 is Pierre with a count of 1963
Number 2 is Prince with a count of 1928
Number 3 is Natásha with a count of 1213
Number 4 is Andrew with a count of 1143
Number 5 is himself with a count of 1020
Number 6 is princess with a count of 915
Number 7 is French with a count of 881
Number 8 is before with a count of 833
Number 9 is Rostóv with a count of 776
Number 10 is thought with a count of 767
Total Time was 1.050269484 seconds
```

