Tony Huang ID: 915090688
Github: tzphuang CSC415 Operating Systems

Assignment 4 - Word Blast

Description:

Assignment 4 – Word blast was an assignment to write a C program that intakes a text file and split it into multiple equal parts. After splitting the file the program runs a specified number of threads the user wants equal to the amount of equal splits the text file was chopped up by and process each individual chunk of the text file per thread. The processing is by tokenizing the chunk of texts into greater than or equal to 6 letter words and storing them while keeping count of how much times they show up. After processing the text file we then print out the top 10 words that occur with their count within that text file.

Approach / What I Did:

What I did for this assignment was to have a dynamically expanding list of nodes to contain the words I would process as I do not know how much 6 or greater than 6 letter words there are in the text file so I used a structure called node. This node held information about a string, its current running count in the text file, and a pointer to the previous node that was created. With this structure I could have a dynamically allocated list where the size would grow equal to what I needed it to be. After the list was up and running I created functions to insert new words, increment the counter inside a node, sort the node list for printing purposes later, and printing functions that would list the top 10 words with their counts or just list all the words for testing purposes. Once all the functions relating to the node list was done I moved onto the main function that would call my node functions and here I would multithread the program. I started by intaking the argv[2] which represented the number associated with the number of threads I would need and since argv[2] was a char * I had to convert it to a integer. I then created many for loops to chunk up my read in text file, to store it, to create multiple threads, and to have multiple thread joins. After this I did clean up where I cleaned up multiple malloced memory and closed my read in text file.

Issues and Resolutions:

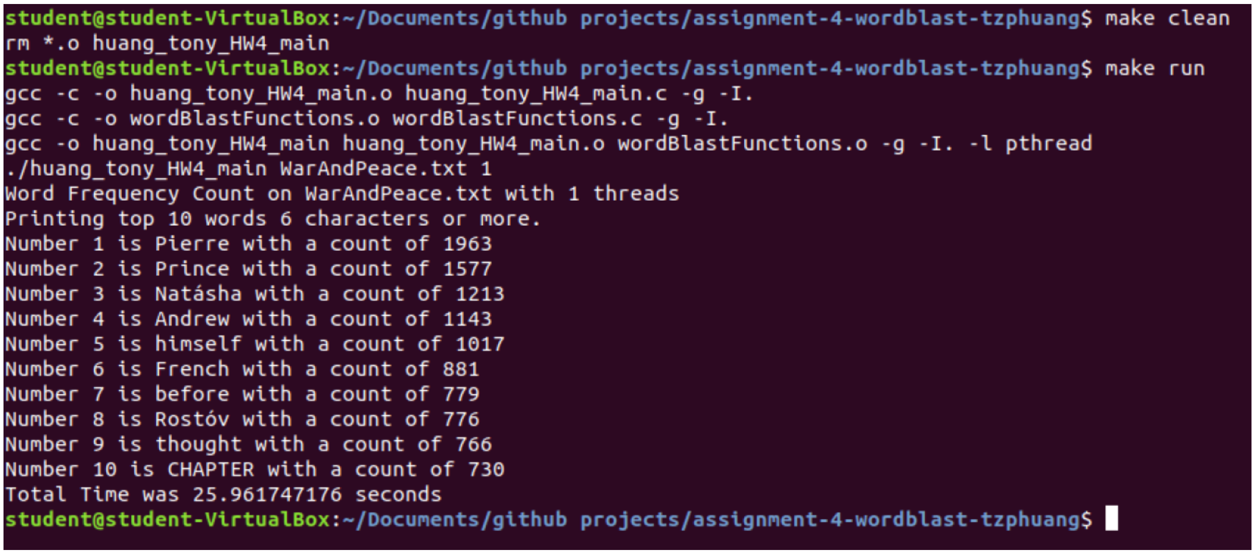
I had a very hard time understanding how to create multiple threads and loading them dynamically. This was fixed by for loops and an array of pointers to different threads. I also had a very hard time figuring out where to put my mutex locks. The way I solved this was by just placing it where I knew changes would occur ie: I serialized the portion of code relating to inserting a new node.

Analysis:

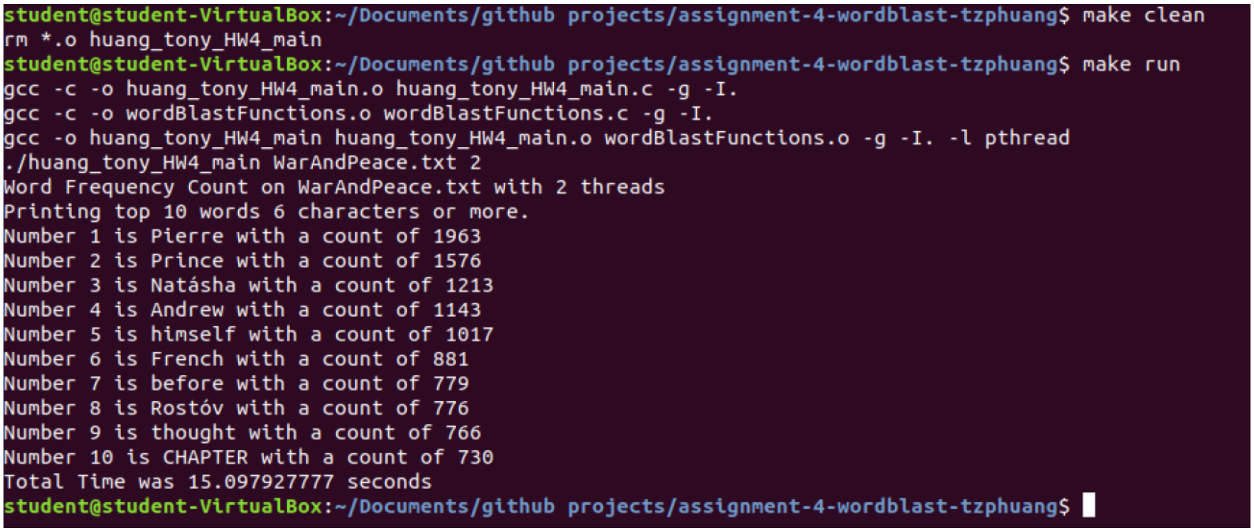
The reason why the times are what they are is simple. Thread 1 should be the longest time as everything is serialized and since it is all serialized and nothing is multi-threaded it would take the longest. The threaded 2/4/8 run programs are around the same time and I believe the reason for this is because as Professor Bierman said “its like a funnel” the parts of code that mutex lock was serializing are the things stalling the program and we see the biggest change come from 1 to 2 threads as in the double threaded program where the only time things get serialized is when the mutex lock is activated during inserting a new node this means that the cpu can still work on other things from the other thread while that inserting is happening. Thus making more efficient time for the program. But this means that since the serializing of the code is with the mutex happens, then for the multi threaded program with 4/8 threads it means that it would be essentially like the 2 threaded program. It would mean that when a thread is waiting for its turn to insert a new node it will doing nothing which means added time, and more threads means more likelihood of that happening over and over as each individual thread can interfere with each other while in the 2 thread program they are really only worried about each other finishing.

Screen shot of compilation:
Compilation down below

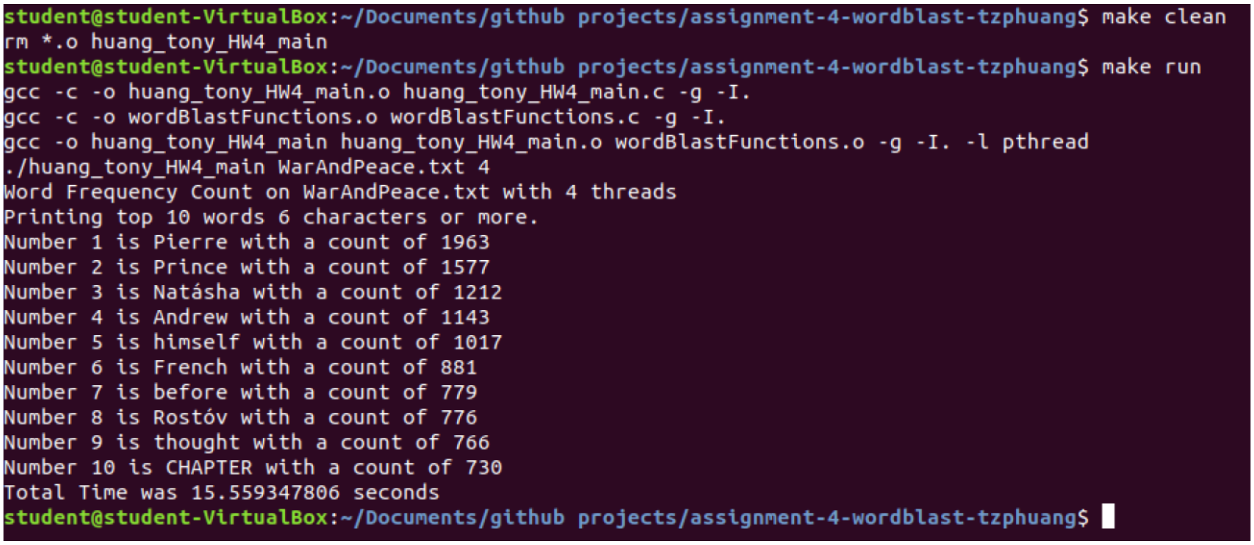
Screen shot(s) of the execution of the program:



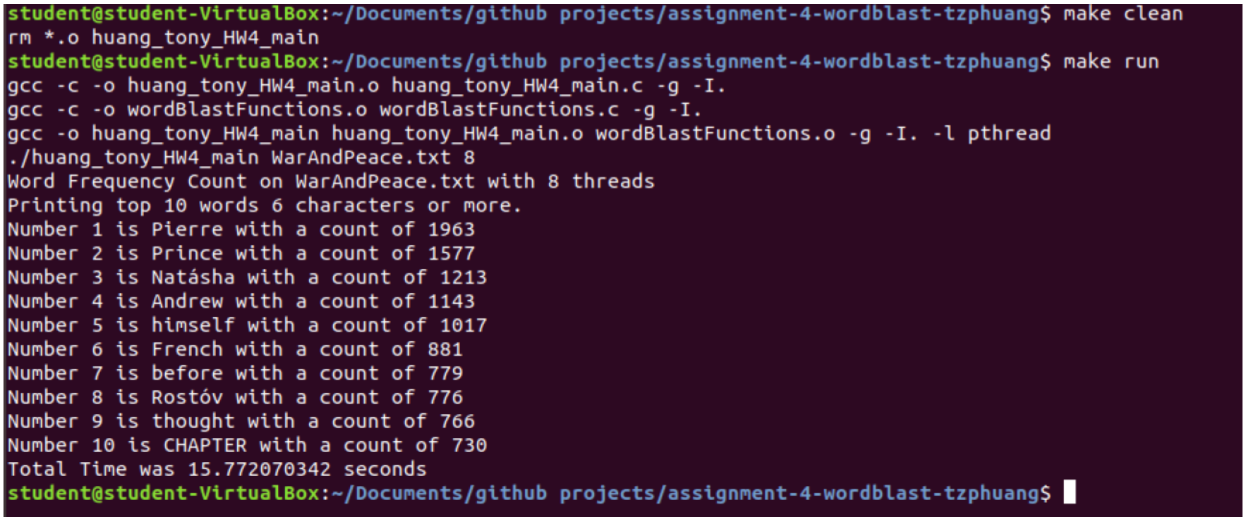
```
student@student-VirtualBox:~/Documents/github projects/assignment-4-wordblast-tzphuang$ make clean
rm *.o huang_tony_HW4_main
student@student-VirtualBox:~/Documents/github projects/assignment-4-wordblast-tzphuang$ make run
gcc -c -o huang_tony_HW4_main.o huang_tony_HW4_main.c -g -I.
gcc -c -o wordBlastFunctions.o wordBlastFunctions.c -g -I.
gcc -o huang_tony_HW4_main huang_tony_HW4_main.o wordBlastFunctions.o -g -I. -l pthread
./huang_tony_HW4_main WarAndPeace.txt 1
Word Frequency Count on WarAndPeace.txt with 1 threads
Printing top 10 words 6 characters or more.
Number 1 is Pierre with a count of 1963
Number 2 is Prince with a count of 1577
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 1017
Number 6 is French with a count of 881
Number 7 is before with a count of 779
Number 8 is Rostóv with a count of 776
Number 9 is thought with a count of 766
Number 10 is CHAPTER with a count of 730
Total Time was 25.961747176 seconds
student@student-VirtualBox:~/Documents/github projects/assignment-4-wordblast-tzphuang$ 
```



```
student@student-VirtualBox:~/Documents/github projects/assignment-4-wordblast-tzphuang$ make clean
rm *.o huang_tony_HW4_main
student@student-VirtualBox:~/Documents/github projects/assignment-4-wordblast-tzphuang$ make run
gcc -c -o huang_tony_HW4_main.o huang_tony_HW4_main.c -g -I.
gcc -c -o wordBlastFunctions.o wordBlastFunctions.c -g -I.
gcc -o huang_tony_HW4_main huang_tony_HW4_main.o wordBlastFunctions.o -g -I. -l pthread
./huang_tony_HW4_main WarAndPeace.txt 2
Word Frequency Count on WarAndPeace.txt with 2 threads
Printing top 10 words 6 characters or more.
Number 1 is Pierre with a count of 1963
Number 2 is Prince with a count of 1576
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 1017
Number 6 is French with a count of 881
Number 7 is before with a count of 779
Number 8 is Rostóv with a count of 776
Number 9 is thought with a count of 766
Number 10 is CHAPTER with a count of 730
Total Time was 15.097927777 seconds
student@student-VirtualBox:~/Documents/github projects/assignment-4-wordblast-tzphuang$ 
```



```
student@student-VirtualBox:~/Documents/github projects/assignment-4-wordblast-tzphuang$ make clean
rm *.o huang_tony_HW4_main
student@student-VirtualBox:~/Documents/github projects/assignment-4-wordblast-tzphuang$ make run
gcc -c -o huang_tony_HW4_main.o huang_tony_HW4_main.c -g -I.
gcc -c -o wordBlastFunctions.o wordBlastFunctions.c -g -I.
gcc -o huang_tony_HW4_main huang_tony_HW4_main.o wordBlastFunctions.o -g -I. -l pthread
./huang_tony_HW4_main WarAndPeace.txt 4
Word Frequency Count on WarAndPeace.txt with 4 threads
Printing top 10 words 6 characters or more.
Number 1 is Pierre with a count of 1963
Number 2 is Prince with a count of 1577
Number 3 is Natásha with a count of 1212
Number 4 is Andrew with a count of 1143
Number 5 is himself with a count of 1017
Number 6 is French with a count of 881
Number 7 is before with a count of 779
Number 8 is Rostóv with a count of 776
Number 9 is thought with a count of 766
Number 10 is CHAPTER with a count of 730
Total Time was 15.559347806 seconds
student@student-VirtualBox:~/Documents/github projects/assignment-4-wordblast-tzphuang$
```



```
student@student-VirtualBox:~/Documents/github projects/assignment-4-wordblast-tzphuang$ make clean
rm *.o huang_tony_HW4_main
student@student-VirtualBox:~/Documents/github projects/assignment-4-wordblast-tzphuang$ make run
gcc -c -o huang_tony_HW4_main.o huang_tony_HW4_main.c -g -I.
gcc -c -o wordBlastFunctions.o wordBlastFunctions.c -g -I.
gcc -o huang_tony_HW4_main huang_tony_HW4_main.o wordBlastFunctions.o -g -I. -l pthread
./huang_tony_HW4_main WarAndPeace.txt 8
Word Frequency Count on WarAndPeace.txt with 8 threads
Printing top 10 words 6 characters or more.
Number 1 is Pierre with a count of 1963
Number 2 is Prince with a count of 1577
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 1017
Number 6 is French with a count of 881
Number 7 is before with a count of 779
Number 8 is Rostóv with a count of 776
Number 9 is thought with a count of 766
Number 10 is CHAPTER with a count of 730
Total Time was 15.772070342 seconds
student@student-VirtualBox:~/Documents/github projects/assignment-4-wordblast-tzphuang$
```