Nyan Ye Lin ID: 921572181
Github: yye99 CSC415 Operating Systems

Assignment 4 - Word Blast

Description:

The purpose of this assignment to famarilize using multiple threads in paralle, mutex locks, and linux file functions such as open,close,read,lseek and pread.

Approach / What I Did:

I created linkedlist to store the words and the count of the words. I used a global linkedlist to store all the words and count.

Step 1

- Got the file name and number of threads from the command line.

Step 2

- Created findFileSize() function to find out the size of the file and lseek() is used to return the size of the file.

Step 3.

- Created readFile() function with pread() function to read the file into a buffer.

Step 4.

- Calculated each file chunk size based on the number of threads. The size of the last chuck is each chunk size + remainder (fileSize % threads).

Step 5.

- Created ThreadData Struct which has variables such as fileName, chunkSize and offset and created ThreadData array with size of threadCount and each of the element of ThreadData store the fileName, chunkSize it need to process and offset of it need start process. Each element of ThreadData array will later pass into void* function,

Step 6.

- Created void* function and the argument of the function is later deferenced and type cast into ThreadData. Inside the function, I created the buffer array with chunkSize and read the assigned chunk with readFile function. After reading the file into buffer, each word in the buffer is tokenize with strtok_r() (threadsafe). Each tokenized words which has six or more characters are stored in a new linked list with the count. This linked list is later stored back into the global linked list. This part is mutex locked and unlocked.

Step 7.

- The global linked list is sorted with insertion sort and top ten words with highest frequency are printed.

Issues and Resolutions:

The first Issue is using char pointer array to store chunks of the file to tokenize with strtok_r and strtok_r not working. I later found out that strtok_r does not work with string literals since it has to modified the string in the stack to tokenize the substrings. This issue is fixed by using char arrays.

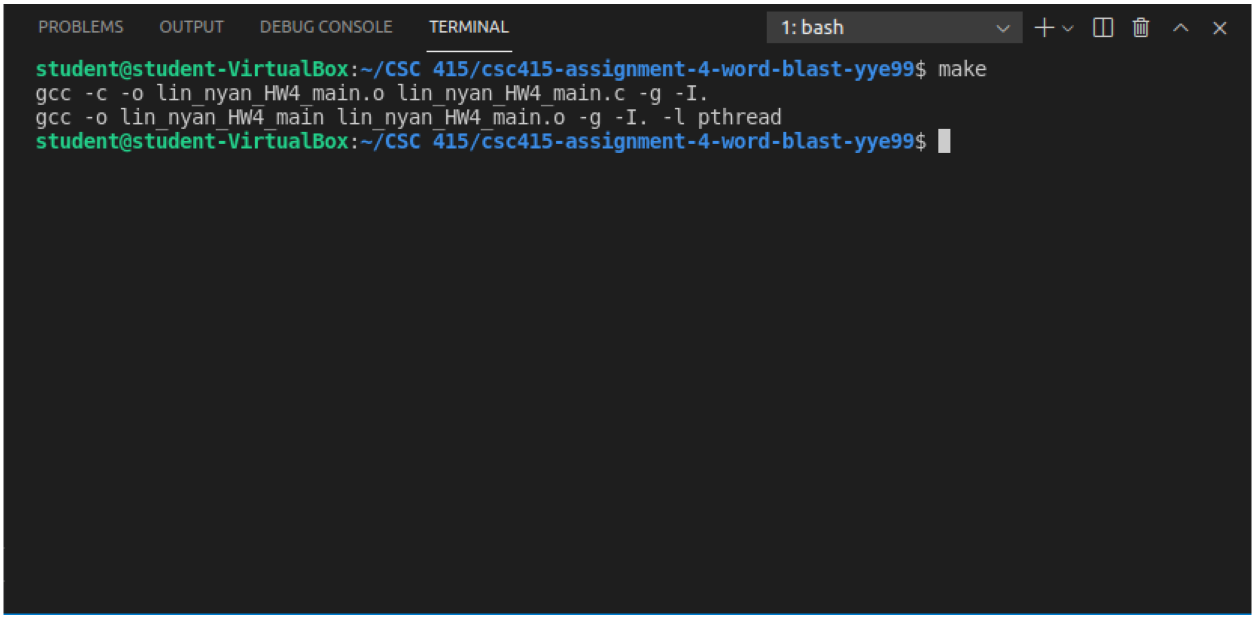
The second issues is my word counts are not the same with each number of threads. One thread give me the right number of words and frequency. The more threads I have, the less words were counted and I still can't figure out where I went wrong. However, time gets shorter with more threads.

The third issue is words and the word count getting overwritten with each threads due to wrong placement of mutex locks and not putting back the words stored in a linkedlist created to the global linkedlist. This issue is solved by putting the locks before adding to the list and after adding to the list.

Nyan Ye Lin
Github: yye99

ID: 921572181
CSC415 Operating Systems

Screenshot of compilation:

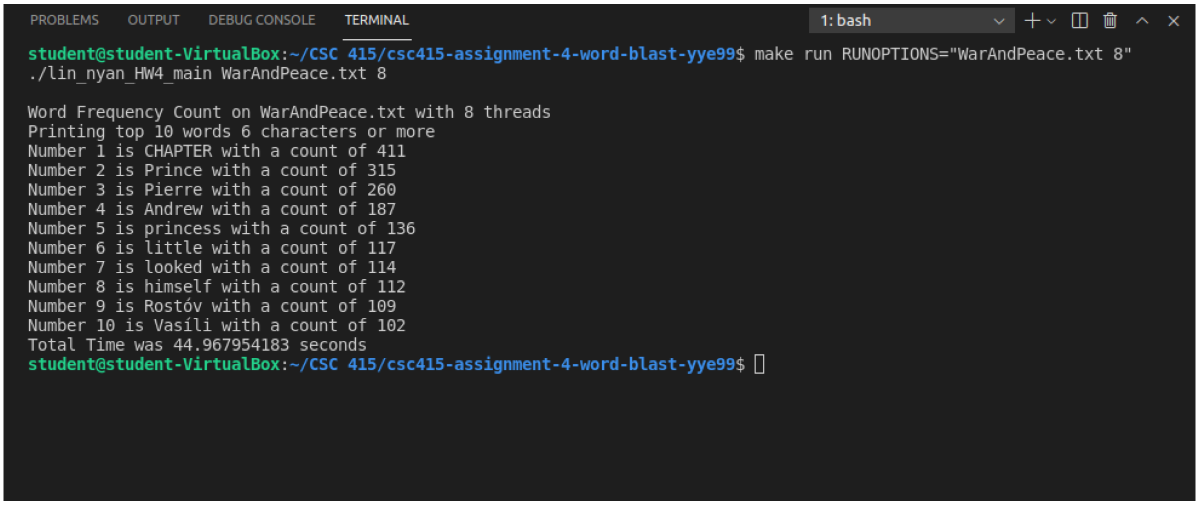


```
student@student-VirtualBox:~/CSC 415/csc415-assignment-4-word-blast-yye99$ make
gcc -c -o lin_nyan_HW4_main.o lin_nyan_HW4_main.c -g -I.
gcc -o lin_nyan_HW4_main lin_nyan_HW4_main.o -g -I. -l pthread
student@student-VirtualBox:~/CSC 415/csc415-assignment-4-word-blast-yye99$
```

Screenshot of the execution of the program:

Reading with 8 threads:

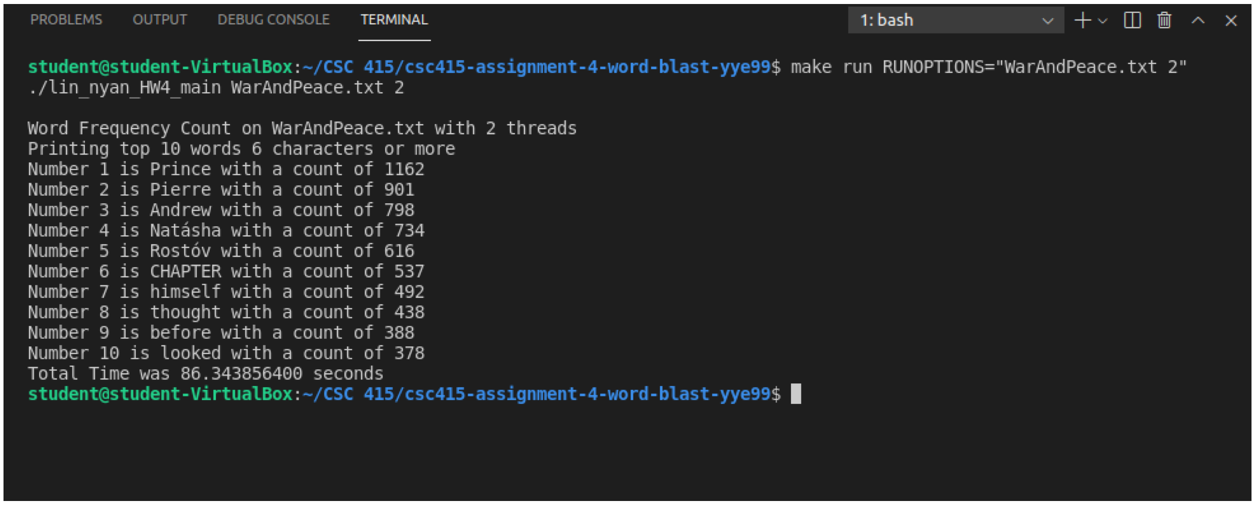
Time Taken : 44.9 seconds



```
student@student-VirtualBox:~/CSC 415/csc415-assignment-4-word-blast-yye99$ make run RUNOPTIONS="WarAndPeace.txt 8"
./lin_nyan_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 CHAPTER with a count of 411
Number 2 is Prince with a count of 315
Number 3 is Pierre with a count of 260
Number 4 is Andrew with a count of 187
Number 5 is princess with a count of 136
Number 6 is little with a count of 117
Number 7 is looked with a count of 114
Number 8 is himself with a count of 112
Number 9 is Rostóv with a count of 109
Number 10 is Vasíli with a count of 102
Total Time was 44.967954183 seconds
student@student-VirtualBox:~/CSC 415/csc415-assignment-4-word-blast-yye99$
```

Reading with 2 threads:

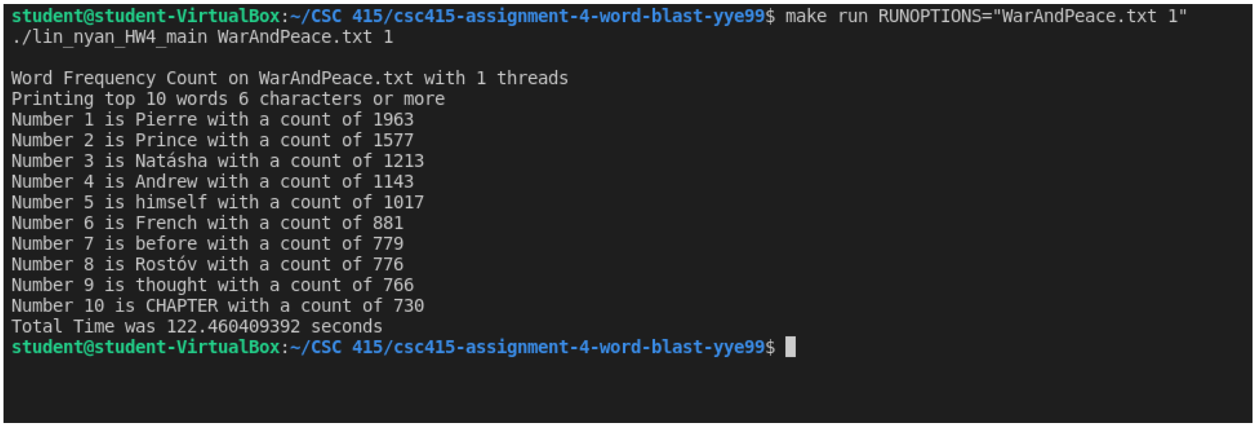
Time Taken: 86.34 seconds



```
student@student-VirtualBox:~/CSC 415/csc415-assignment-4-word-blast-yye99$ make run RUNOPTIONS="WarAndPeace.txt 2"
./lin_nyan_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 Prince with a count of 1162
Number 2 is Pierre with a count of 901
Number 3 is Andrew with a count of 798
Number 4 is Natásha with a count of 734
Number 5 is Rostóv with a count of 616
Number 6 is CHAPTER with a count of 537
Number 7 is himself with a count of 492
Number 8 is thought with a count of 438
Number 9 is before with a count of 388
Number 10 is looked with a count of 378
Total Time was 86.343856400 seconds
student@student-VirtualBox:~/CSC 415/csc415-assignment-4-word-blast-yye99$
```

Reading with 1 threads:

Time Taken: 122.46 seconds



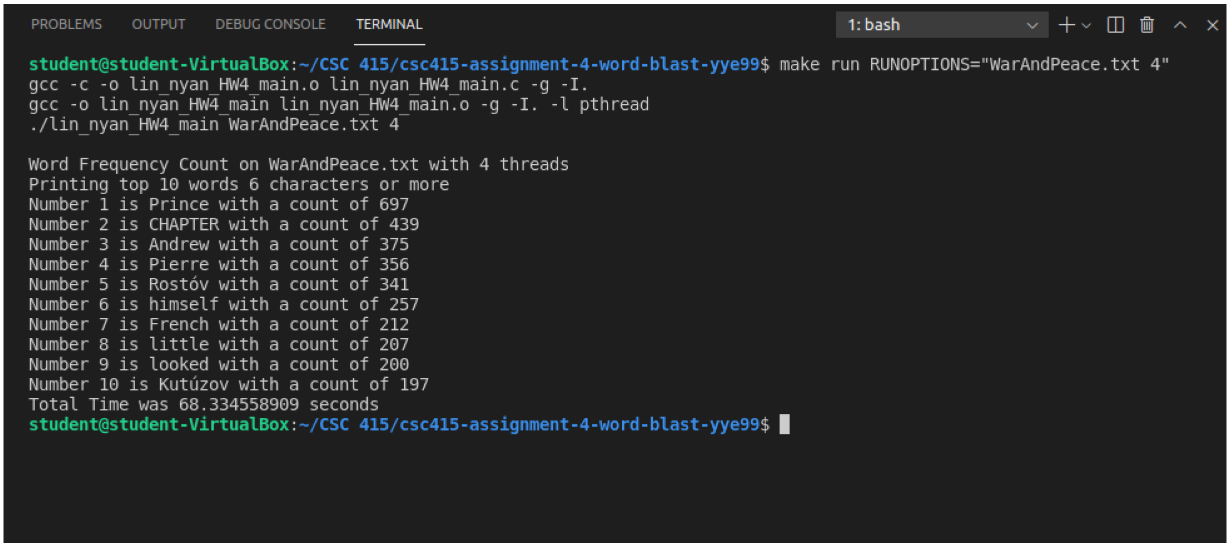
```
student@student-VirtualBox:~/CSC 415/csc415-assignment-4-word-blast-yye99$ make run RUNOPTIONS="WarAndPeace.txt 1"
./lin_nyan_HW4_main WarAndPeace.txt 1

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 122.460409392 seconds
student@student-VirtualBox:~/CSC 415/csc415-assignment-4-word-blast-yye99$
```

Nyan Ye Lin
Github: yye99

ID: 921572181
CSC415 Operating Systems

Reading with 4 threads:

Time taken with 4 threads: 68.33 seconds.



```
student@student-VirtualBox:~/CSC 415/csc415-assignment-4-word-blast-yye99$ make run RUNOPTIONS="WarAndPeace.txt 4"
gcc -c -o lin_nyan_HW4_main.o lin_nyan_HW4_main.c -g -I.
gcc -o lin_nyan_HW4_main lin_nyan_HW4_main.o -g -I. -l pthread
./lin_nyan_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 Prince with a count of 697
Number 2 is CHAPTER with a count of 439
Number 3 is Andrew with a count of 375
Number 4 is Pierre with a count of 356
Number 5 is Rostóv with a count of 341
Number 6 is himself with a count of 257
Number 7 is French with a count of 212
Number 8 is little with a count of 207
Number 9 is looked with a count of 200
Number 10 is Kutúzov with a count of 197
Total Time was 68.334558909 seconds
student@student-VirtualBox:~/CSC 415/csc415-assignment-4-word-blast-yye99$ 
```

Analysis:

In my program, one thread take the longest time with over 120 seconds since it has to process the whole file with one thread, two threads take the second longest with 86.34 seconds, four threads result in 68.33 seconds and the 8 threads takes the fastest with 44.9 seconds. The more threads take less time and it shows that threads are working as expected. They are dividing the file into chucks and each chunk is being processed with each thread in parallel.