

```

/*****
* Project: Assignment 4 – Word Blast
* File: WriteUp4.pdf
*
* Description: This is the write up file for the program and it includes a
* description of the project, what I did, issues I had, resolutions for the
* issues, reason for different times on different thread runs, compilation
* and execution output from the original program.
*****/

```

Description: In this program we read a .txt file (WarandPeace.txt, in this case) and count and tally each of the words that are 6 or more characters long. Also made sure that only Linux file functions are used such as open, close, read, lseek and not library functions. We do this process by using threads and each thread takes a chunk of the file and processes it. It then returns the results to the main and main tallies and prints the top 10, 6 or more-character words with the highest frequencies and their tallies. In the main, I used lseek to get the file size and then divided the file size into equal chunks. I create threads and pass each of the chunks into their respective blocks and then call the pthread_create. The adding of the words is done inside a mutex lock. I made a data structure of arrays and named it Pair. This is implemented into a function addWord. This function checks if a match is found and increments the counter by 1 and if its not found then it adds it to the words array. After it is done, the top 10 words are printed each with their respective frequencies in descending order. The total time taken to perform the threading is also displayed at the end. After the process completes, cleanup is also performed to free all the memory for further usage. There were also a number of issues I had during this assignment. The most difficult was to create a data structure and assign all the words in the array and also to count each of their frequency. The latter was easy as we discussed in the class such as creating threads, waiting for them, destroying them and using mutex locks.

The reason for different times on different thread runs is that as the number of threads increases, the time decreases because of parallelism. The program divides the file into number of threads and performs the operations in parallel. Also, if there are more cores on the operating system, then these threads also use that thus decreasing the time taken for the program to execute.

Compilation and Execution using 1, 2, 4 and 8 threads respectively:

1 Thread :

```
student@student-VirtualBox: ~/Desktop/csc415-assignment-4-word-blast-developer-soni
File Edit View Search Terminal Help
student@student-VirtualBox:~/Desktop/csc415-assignment-4-word-blast-developer-soni$ make
gcc -c -o soni_dev_HW4_main.o soni_dev_HW4_main.c -g -I.
gcc -o soni_dev_HW4_main soni_dev_HW4_main.o -g -I. -l pthread
student@student-VirtualBox:~/Desktop/csc415-assignment-4-word-blast-developer-soni$ make run
./soni_dev_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 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 916
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 1.250578602 seconds
```

2 Threads :

```
student@student-VirtualBox: ~/Desktop/csc415-assignment-4-word-blast-developer-soni
File Edit View Search Terminal Help
student@student-VirtualBox:~/Desktop/csc415-assignment-4-word-blast-developer-soni$ make run
./soni_dev_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 1964
Number 2 is Prince with a count of 1928
Number 3 is Andrew with a count of 1143
Number 4 is himself with a count of 1020
Number 5 is Princess with a count of 916
Number 6 is French with a count of 881
Number 7 is before with a count of 833
Number 8 is thought with a count of 767
Number 9 is CHAPTER with a count of 732
Number 10 is Moscow with a count of 722
Total Time was 0.737575666 seconds
student@student-VirtualBox:~/Desktop/csc415-assignment-4-word-blast-developer-soni$
```

4 Threads :

```
student@student-VirtualBox: ~/Desktop/csc415-assignment-4-word-blast-developer-soni
File Edit View Search Terminal Help
student@student-VirtualBox:~/Desktop/csc415-assignment-4-word-blast-developer-soni$ make run
./soni_dev_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 1928
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 1020
Number 6 is princess with a count of 916
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 0.709057106 seconds
student@student-VirtualBox:~/Desktop/csc415-assignment-4-word-blast-developer-soni$
```

8 Threads :

```
student@student-VirtualBox: ~/Desktop/csc415-assignment-4-word-blast-developer-soni
File Edit View Search Terminal Help
student@student-VirtualBox:~/Desktop/csc415-assignment-4-word-blast-developer-soni$ make run
./soni_dev_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 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 916
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 0.747526981 seconds
student@student-VirtualBox:~/Desktop/csc415-assignment-4-word-blast-developer-soni$
```