Mahek Delawala SFSU ID: 922968394
Github: Mahek-Delawala CSC415 Operating Systems

Assignment 4 - Word Blast

Description:
This assignment is to write a C program that accepts two arguments, one containing the file name and other containing the Thread Count. Based on these, we need to Print the Top 10 Words with string length of six or more. Processing of the words needs to be done simultaneously in each thread based on the thread count.

Approach / What I Did:

To solve this assignment, first I carefully read the description about the assignment and tried to understand the expectations of the assignment. Then, I tried to break down the problem into smaller parts so that, it is easy to implement the small parts rather than doing the problem as a whole. Below listed is the breakdown of the problem.

1. Accept the inputs from the command line regarding the file name and thread Count.

2. First, I open the file and Calculate the File size of the input File.

3. Based on the number of threads, first I calculated the chunk size to read by each thread from the file. Chunk size was calculated based on total file size divided by the thread count. I calculated left over chunk size which will be used by the last thread in case of multithreading to process the remaining chunks of the data. I also initialize my mutex lock.

4. I finalized the data structure which I am going to use for storing the words. I am using structure which contains three member fields namely word string, frequency, and the string length. Initially, I also initialize the frequency count of my array of structure to zero.

5. Then, I start creating the threads based on the thread count and in the arguments of each thread, I pass the thread index which will be used in the thread handler function.

6. Then, I implemented the thread handler Function. Inside the thread handler function, based on the thread index, I read the data of the file into the buffer using the pread() function. I calculate the offset based on the chunk size and leftover size and pass this as an argument to the offset which reads the data of count size from the given offset. Once, get the data into the buffer, then I start the processing of the word. For this, I first split the words using strtok_r() function by passing the delimiter. After splitting the words, I store it into the array of structure. For storing the data, first I check if the word is already present in my existing array, if it is present, I simply increment the count of the word and if it is new word, I store the word in array of structure and increment the

count. The thread handler function is my critical section, so to maintain the synchronization among the threads, I lock the critical section part using mutex locks and then finally unlock once it does the processing.

7. Then, I sort my array of structure containing word details in the decreasing order based on the frequency count of each word.

8. Finally, I print down the top 10 words with string length greater than or equal to six in my main function. In my main, I wait for my threads to complete their execution by using pthread_join() function.

9. Finally, at last I clean up resources. I close the file descriptor; free the memory allocated by the buffer and destroy the mutex lock.

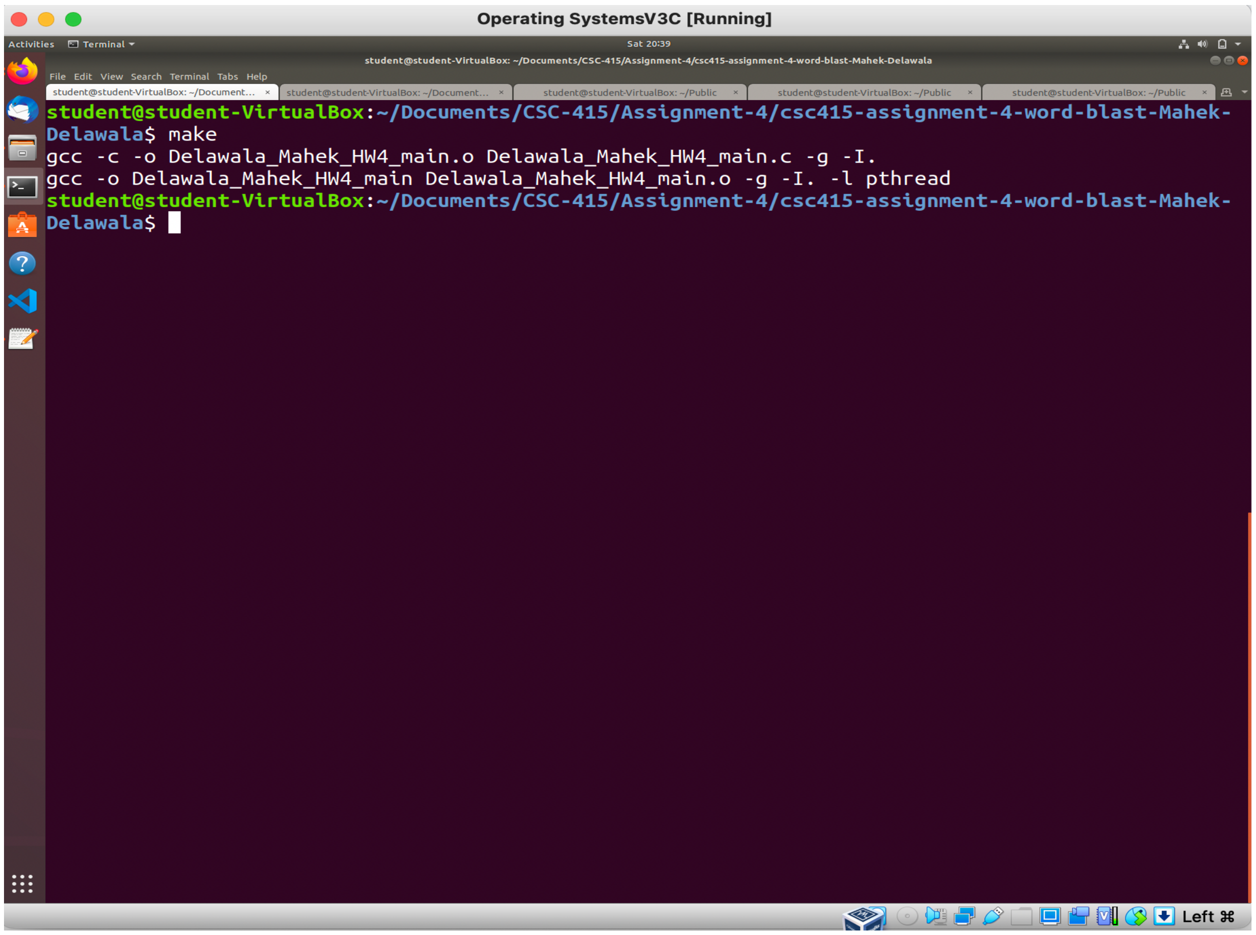
Issues and Resolutions:

During the implementation, I faced several issues. First problem, which I faced was how I make my threads read the data simultaneously from the file. First, I thought of implementing using the read but while going through the man page of the read function, I found better approach to read using pread() function as it is suitable for reading the data for using the multithreading.

Next during the creation of threads, I wanted to pass the thread index for each thread in the handler function. But initially, I was just passing integer argument which used to take only the last index value for all the threads as argument. After discussing with my friend, and reading the sample application code in the man page, I stored respective index in the array of pointer and passed the appropriate index of pointer. This solved my issue.

I debugged a lot in finalizing my critical section code for using the mutex locks. Based on the prints and running multiple times, I finalized my critical section and then used the locks appropriately.

Screen shot of compilation:

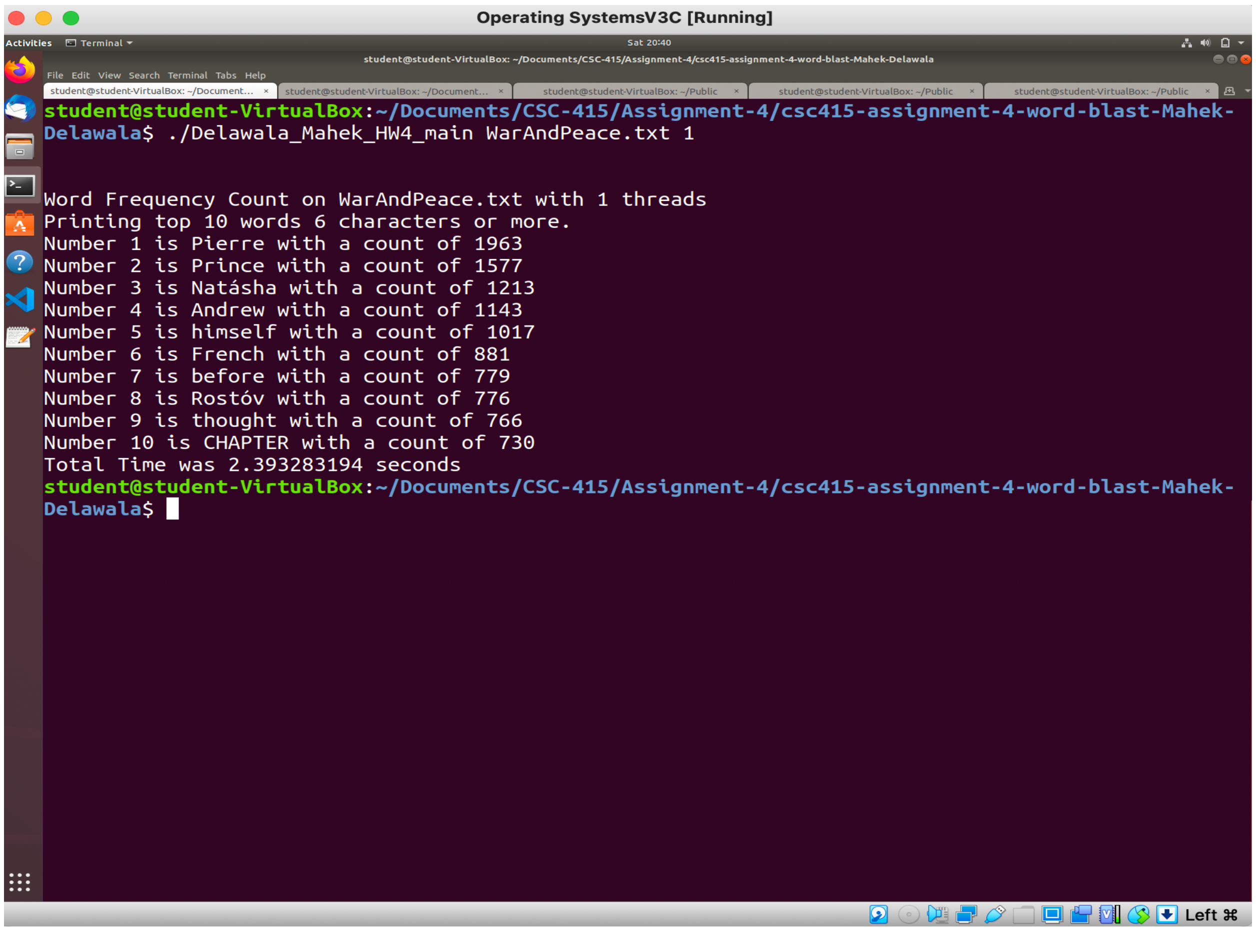


```
student@student-VirtualBox:~/Documents/CSC-415/Assignment-4/csc415-assignment-4-word-blast-Mahek-Delawala$ make
gcc -c -o Delawala_Mahek_HW4_main.o Delawala_Mahek_HW4_main.c -g -I.
gcc -o Delawala_Mahek_HW4_main Delawala_Mahek_HW4_main.o -g -I. -l pthread
student@student-VirtualBox:~/Documents/CSC-415/Assignment-4/csc415-assignment-4-word-blast-Mahek-Delawala$ 
```

Operating SystemsV3C [Running]
Activities
Terminal
Sat 20:39
File Edit View Search Terminal Tabs Help
Left ⌘

Fig - 1

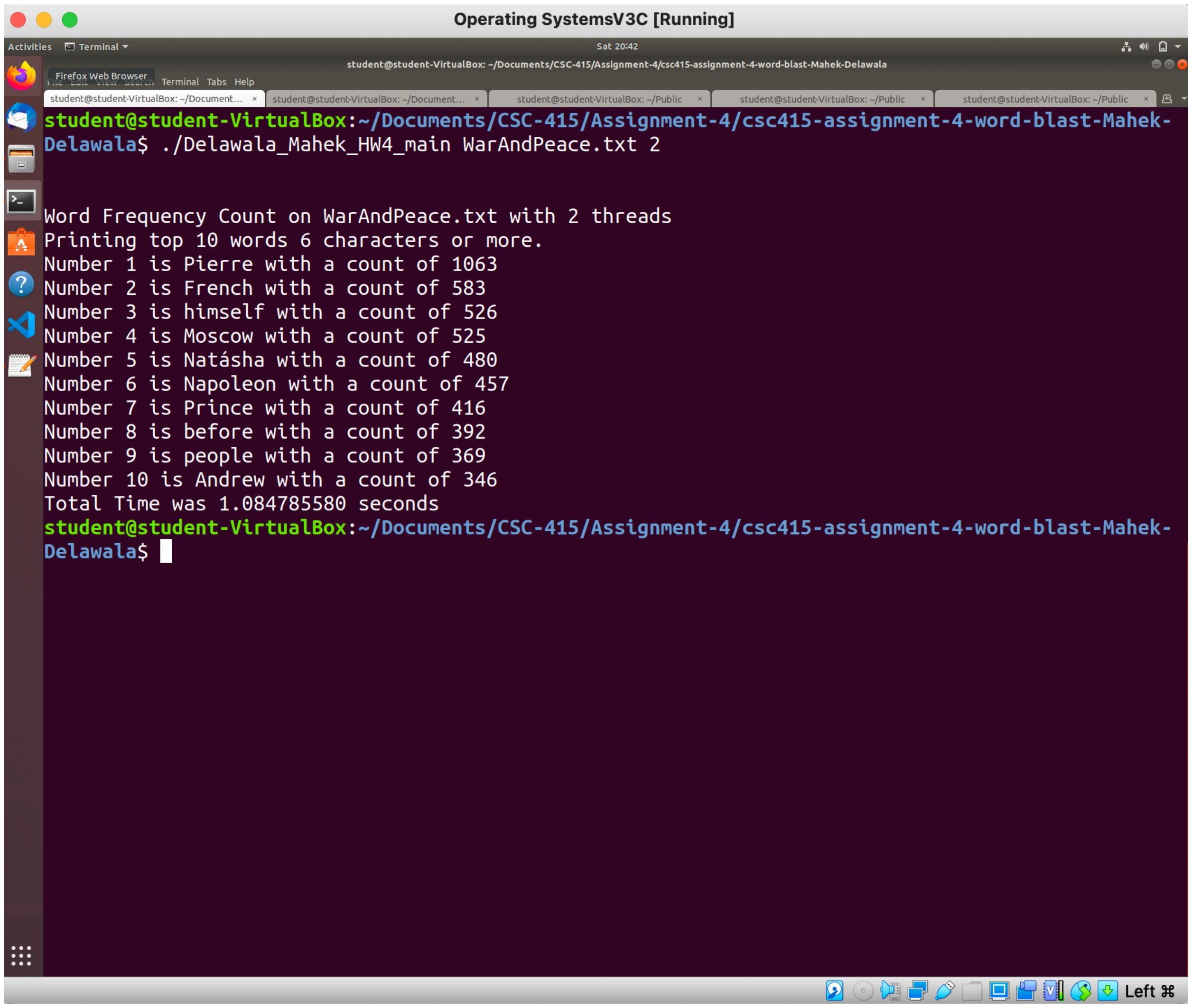
Screen shot(s) of the execution of the program:



```
student@student-VirtualBox:~/Documents/CSC-415/Assignment-4/csc415-assignment-4-word-blast-Mahek-
Delawala$ ./Delawala_Mahek_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 2.393283194 seconds
student@student-VirtualBox:~/Documents/CSC-415/Assignment-4/csc415-assignment-4-word-blast-Mahek-
Delawala$ 
```

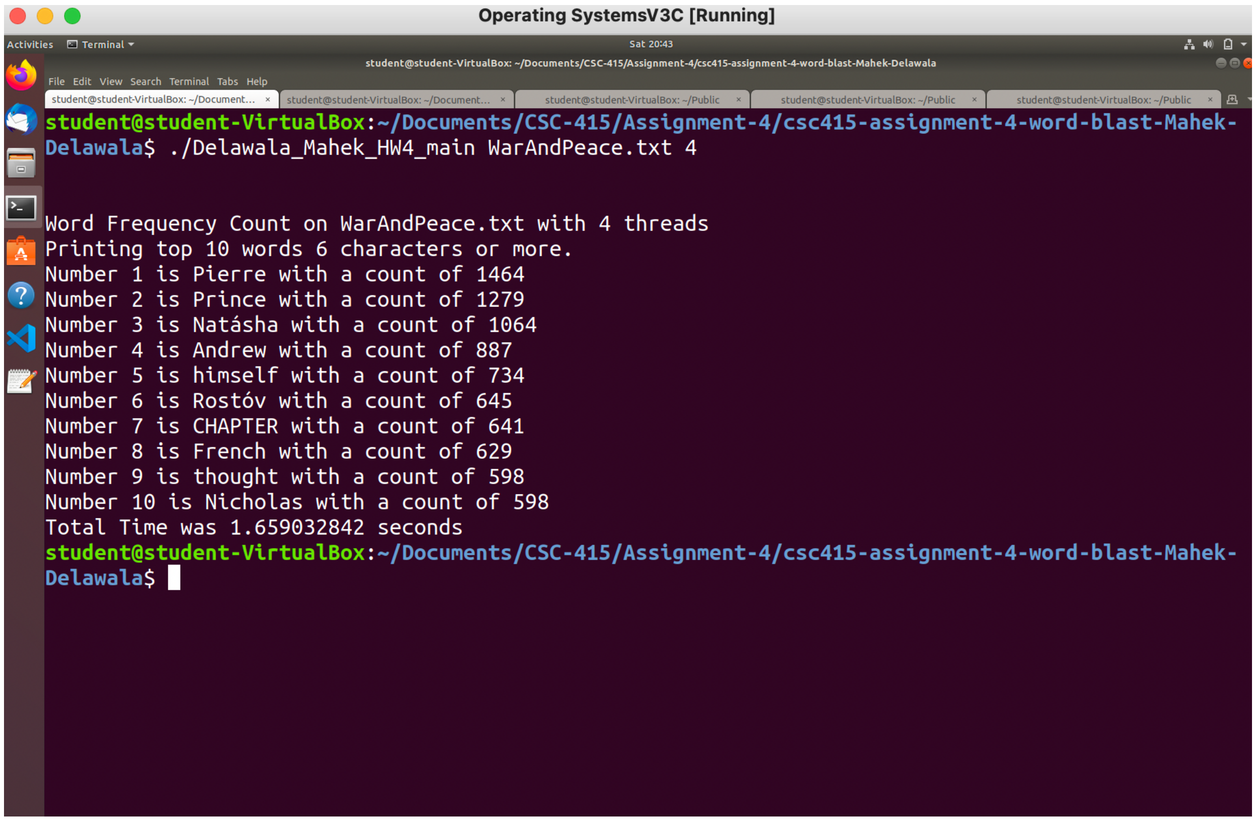
Fig - 2



```
student@student-VirtualBox:~/Documents/CSC-415/Assignment-4/csc415-assignment-4-word-blast-Mahek-Delawala$ ./Delawala_Mahek_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 1063
Number 2 is French with a count of 583
Number 3 is himself with a count of 526
Number 4 is Moscow with a count of 525
Number 5 is Natásha with a count of 480
Number 6 is Napoleon with a count of 457
Number 7 is Prince with a count of 416
Number 8 is before with a count of 392
Number 9 is people with a count of 369
Number 10 is Andrew with a count of 346
Total Time was 1.084785580 seconds
student@student-VirtualBox:~/Documents/CSC-415/Assignment-4/csc415-assignment-4-word-blast-Mahek-Delawala$
```

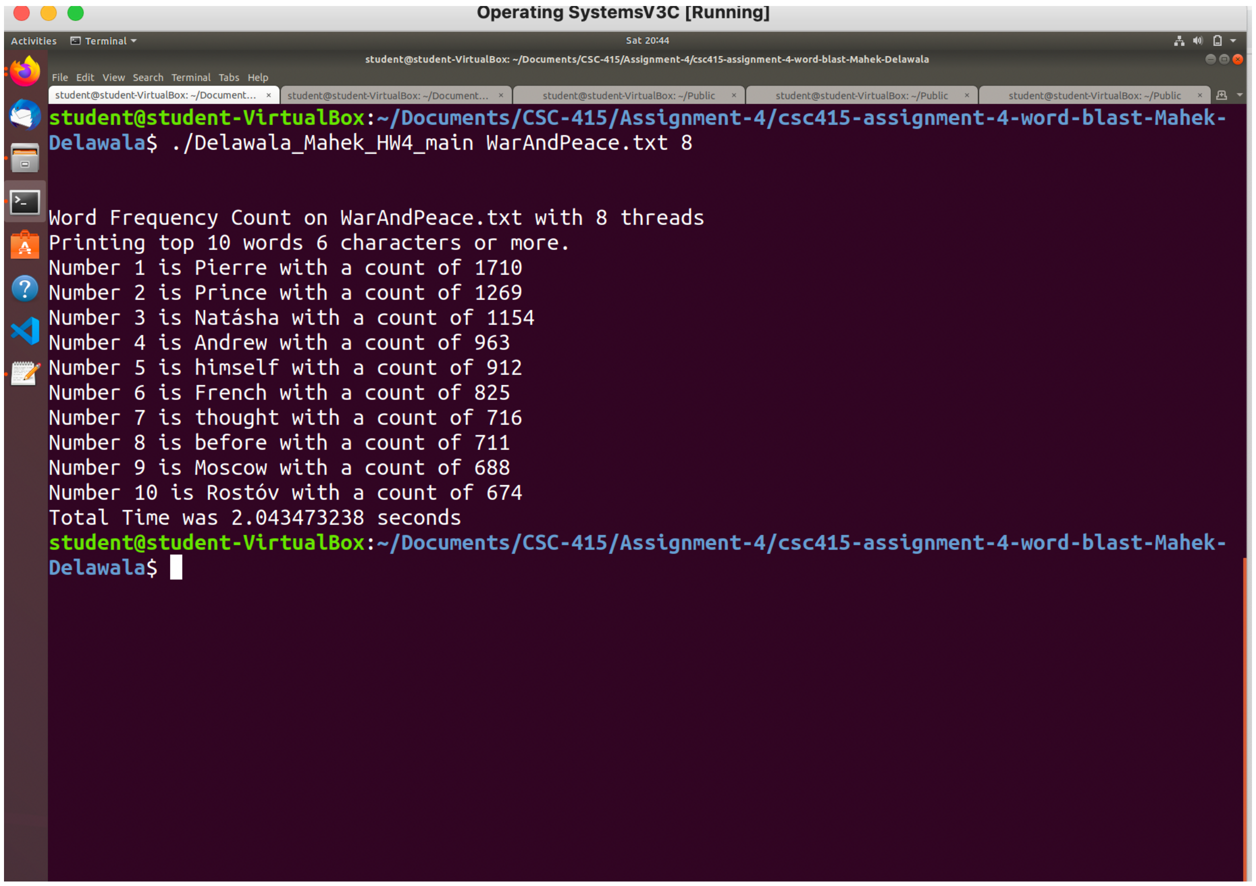
Fig – 3


Operating SystemsV3C [Running]
Activities
Terminal
Sat 20:43
student@student-VirtualBox: ~/Documents/CSC-415/Assignment-4/csc415-assignment-4-word-blast-Mahek-Delawala
File Edit View Search Terminal Tabs Help

```
student@student-VirtualBox:~/Documents/CSC-415/Assignment-4/csc415-assignment-4-word-blast-Mahek-
Delawala$ ./Delawala_Mahek_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 1464
Number 2 is Prince with a count of 1279
Number 3 is Natásha with a count of 1064
Number 4 is Andrew with a count of 887
Number 5 is himself with a count of 734
Number 6 is Rostóv with a count of 645
Number 7 is CHAPTER with a count of 641
Number 8 is French with a count of 629
Number 9 is thought with a count of 598
Number 10 is Nicholas with a count of 598
Total Time was 1.659032842 seconds
student@student-VirtualBox:~/Documents/CSC-415/Assignment-4/csc415-assignment-4-word-blast-Mahek-
Delawala$ 
```

Fig - 4


Operating SystemsV3C [Running]
Activities
Terminal
Sat 20:44
student@student-VirtualBox: ~/Documents/CSC-415/Assignment-4/csc415-assignment-4-word-blast-Mahek-Delawala
File Edit View Search Terminal Tabs Help

```
student@student-VirtualBox:~/Documents/CSC-415/Assignment-4/csc415-assignment-4-word-blast-Mahek-
Delawala$ ./Delawala_Mahek_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 1710
Number 2 is Prince with a count of 1269
Number 3 is Natásha with a count of 1154
Number 4 is Andrew with a count of 963
Number 5 is himself with a count of 912
Number 6 is French with a count of 825
Number 7 is thought with a count of 716
Number 8 is before with a count of 711
Number 9 is Moscow with a count of 688
Number 10 is Rostóv with a count of 674
Total Time was 2.043473238 seconds
student@student-VirtualBox:~/Documents/CSC-415/Assignment-4/csc415-assignment-4-word-blast-Mahek-
Delawala$
```

Fig - 5

By running the output, based on the number of threads, I observed that processing time of threads was reduced to lesser than or equal to half time when we are doing processing of the words with 2 threads instead of one thread. As shown in Fig - 2, when running application with single thread execution time was around 2.39 seconds but when we run the application with two threads as shown in Fig - 3, the execution time is 1.08 second which is less the half of the original time.

Ideally, going further by running application with 4 and 8 threads, the execution time should be further reduced to the half of the time and 1/8 time respectively. However, as shown in Fig - 4, and Fig - 5, we cannot see the reduction in the execution time because our virtual machine is running with two cores. So, increasing the number of threads will have no effect on the execution as it might increase the thread overheads.