PROJECT – 1A

Design Description

In my implementation I have used 3 IO bounded threads and 1 CPU bounded thread for finding the largest 'k' occurrences of a word.

Initially I will split the file in to three parts depending on the number of lines presented in the given input file. I used **"wc –l file_name "** Linux command to get the number of lines in the file and used split command to split the input files in to three files (**split –l number_of_lines file_name**). Split command will create three file naming "**split_fileaa**", "**split_fileab**", and "**split_fileac**" respectively. Mentioned files are memory mapped separately and they are processed concurrently with three IO threads.

In a file we may have several number of valid words depending on the input. We need a data structure to efficiently store the word and its occurrences. I have implemented **Chain-Hashing Algorithm** in C. It has 26 buckets. Each bucket corresponds to individual alphabet like "Buck 1 - 'A' ". Each bucket will maintain a linked list of words for a given starting alphabet. For example bucket 1 will have a reference to a linked list whose words are starting with 'A' and its corresponding occurrences.

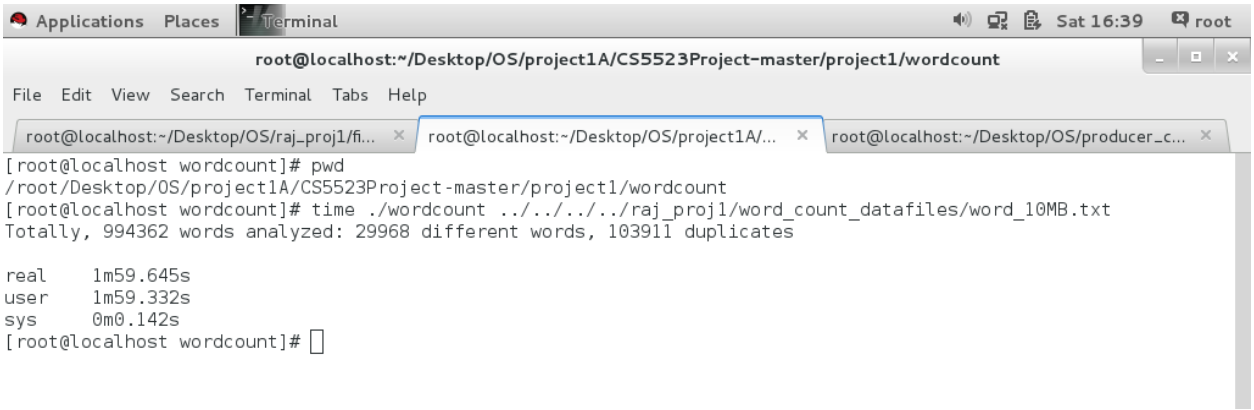
After bookkeeping the list of words and occurrences using hashing, we need an efficient sorting algorithm to display the largest 'k' occurrences in the list. I have implemented **Min-Heap Algorithm** in c to find the largest 'k' occurrences. One CPU thread will perform sorting in the hash structure and finally stores it in the Min-Heap.

The 3 CPU bound thread processes the file concurrently, in the meantime 1 CPU bound thread will wait for the processing for files to complete. After that, it will start running to provide the top most occurrences.

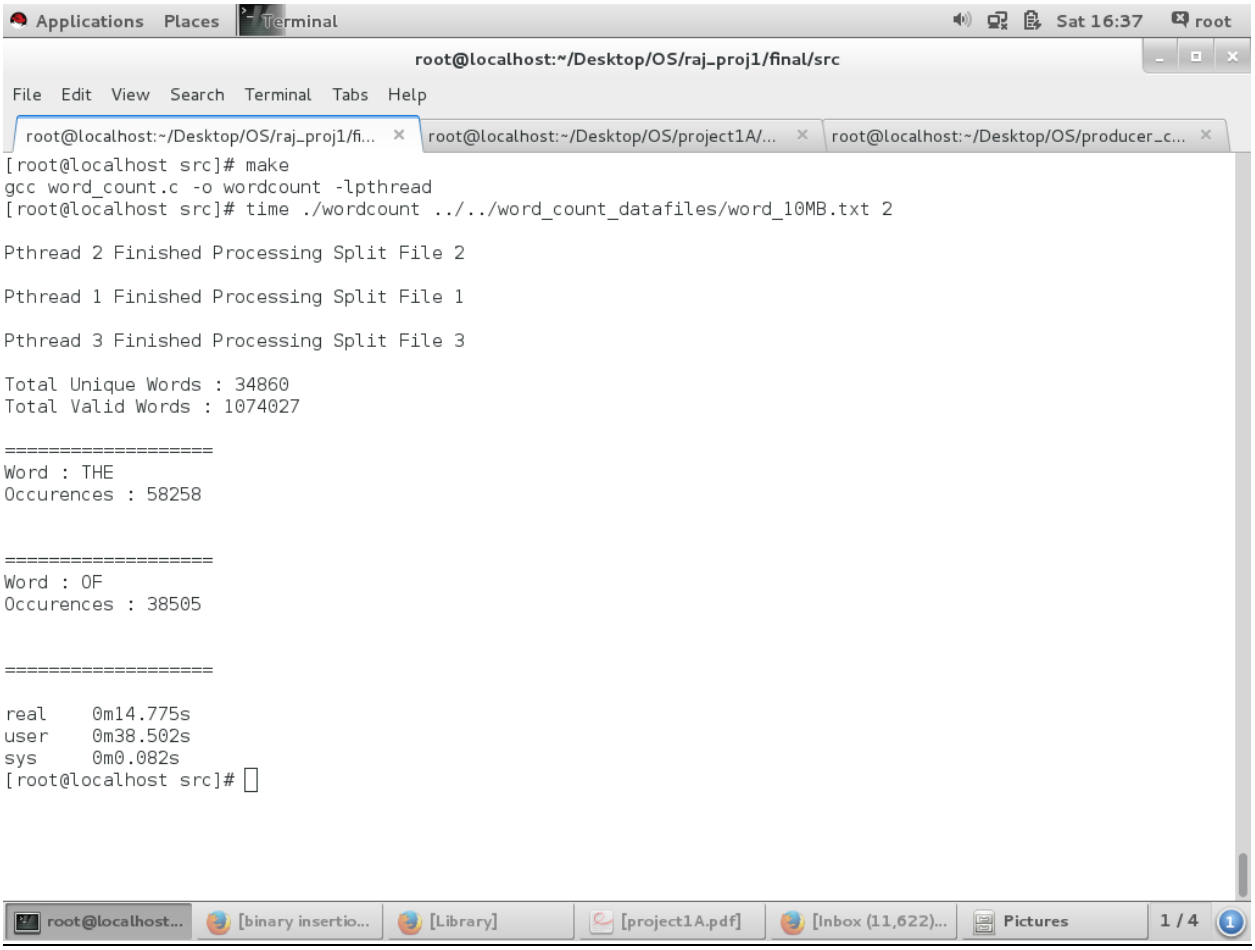
Performance Comparison

Performance is considerably improved with multithreading compared single threading. Below are the screen shorts of same program with both single and multithreading.

Single Threaded Program



Multithreaded Program



The above outputs may vary depending on the specifications of the computer. But multithreaded program will always have minimum execution time when compared to single threaded program.

The above input in observed in the system with following specifications: