**REPORT**

The Algorithm I used is as follows:

-Dividing the file size as:

-Setting start and end points for each thread to read the file on. (So if there are 1o threads there would be 10 start and 10 end points for each thread to read the file to and from)

-These thread points of are then passed to each thread and each thread reads the same file between these boundaries.

-When reading if a thread encounters a word in that line then it stores that line number in a linked list buffer.

-after all threads have finished their execution then the stored line numbers from the linkedlist buffer are printed in an orderly fashion as the linked list have been sorted according to the thread number and the line number.

-Whereas while reading from a piped input using a cat function I’m storing chunks of data into buffer and storing them in a file and performing the same operations on the file as with pargrep.

Disadvantages:

-Since strcpy,strcat functions in the thread function were throwing segmentation faults in a Linux environment I had to copy the line numbers in the linked list buffer and then sort the linked list and then again traverse the file. Solving the issue of segmentation fault in strcat and strcpy I could have sent the full line block from each thread in a linked list or a priority queue buffer using heaps and printed the heap directly this would remove the overheads of traversing through file lines again the second time and sorting the Linked list to give a synchronized out.

**Grep Performance:**

**0.176**……… (Searching for the word ‘the’ in the text file https://www.gutenberg.org/files/1342/1342-0.txt)

**Pargrep** **performance** :

(Searching for the word ‘the’ in the text file https://www.gutenberg.org/files/1342/1342-0.txt)

(note that the time on every run this the lowest time taken after multiple computations)

|  |  |
| --- | --- |
| NO. OF THREADS | TIME TAKEN |
| 1 | 0.108 |
| 2 | 0.164 |
| 3 | 0.144 |
| 4 | 0.162 |
| 5 | 0.156 |
| 6 | 0.138 |
| 7 | 0.136 |
| 10 | 0.276 |
| 15 | 0.416 |
| 20 | 0.539 |