Word Indexer

High Level Design

1. Introduction

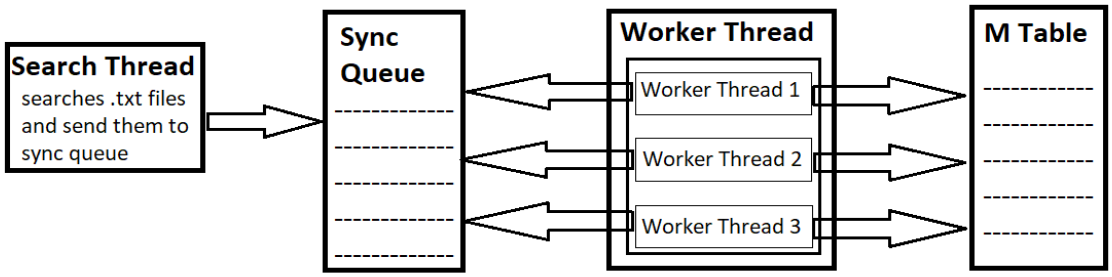
1.1 Problem

Create a multi-threaded word indexing command line application in c++ that works as follows:

1. Accepts as input a directory or path to directory containing text files and a attribute 'h' to display top 10 words or 't' to display bottom 10 words or leave it blank to display all words on the command line
2. Have one thread which searches for any text files present in the directory or subdirectories
3. Whenever a text file is found, it should be handed over to the worker thread for further processing and the search thread should continue its working simultaneously
4. Threre should be a fixed number of worker threads that handle file processing
5. Whenever the worker thread receives any file, it reads the contents and any word other than alphanumeric characters delimits the words
6. A master table in memory, shared between all threads keeps track of all unique words Number of times it occured in the files, if it is first occurence, it is added to the table else, the count of occurence is increased. Words should be matched case insensitive and without punctuation.
7. Once the file search is complete all text files have been processed, the program prints the words and their counts according to the attribute given.

2. Architecture

2.1 Architectural Diagram



2.2 Modules

There are three modules, SearchThread, SyncQueue and WorkerThread

2.2.1 SearchThread

This module search for .txt files in the path specified as command line arguement. Whenever it encounters any .txt file, it sends the file to the SyncQueue module to get processed by the worker thread. Once the search is over, the search thread stops working.

2.2.2 SyncQueue

This module sends the incoming files in a syncronized queue. This module provides files to the WorkerThread module for processing. It provides access to its queue to only 1 worker thread at a time.

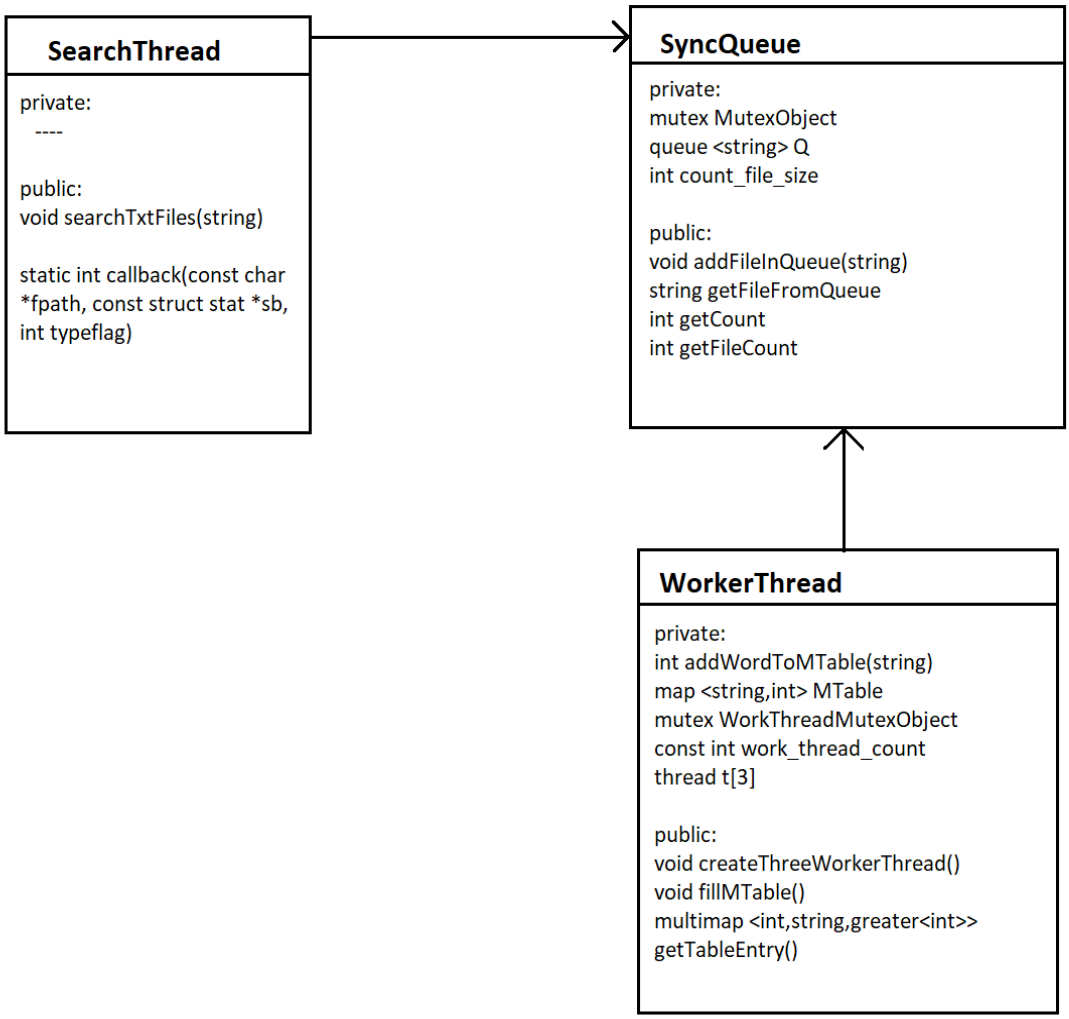
2.2.3 WorkerThread

This module has 3 working threads and each thread gets a file from the SyncQueue for processing. After fetching a file, each worker thread reads the contents of the file and extract words from it to save in a data structure called Mtable.
Mtable contains unique words with their frequencies ordered by their decreasing occurences.

Class Diagram

The program has been divided into 3 classes:

1. SearchThread
2. SyncQueue
3. WorkerThread



Development

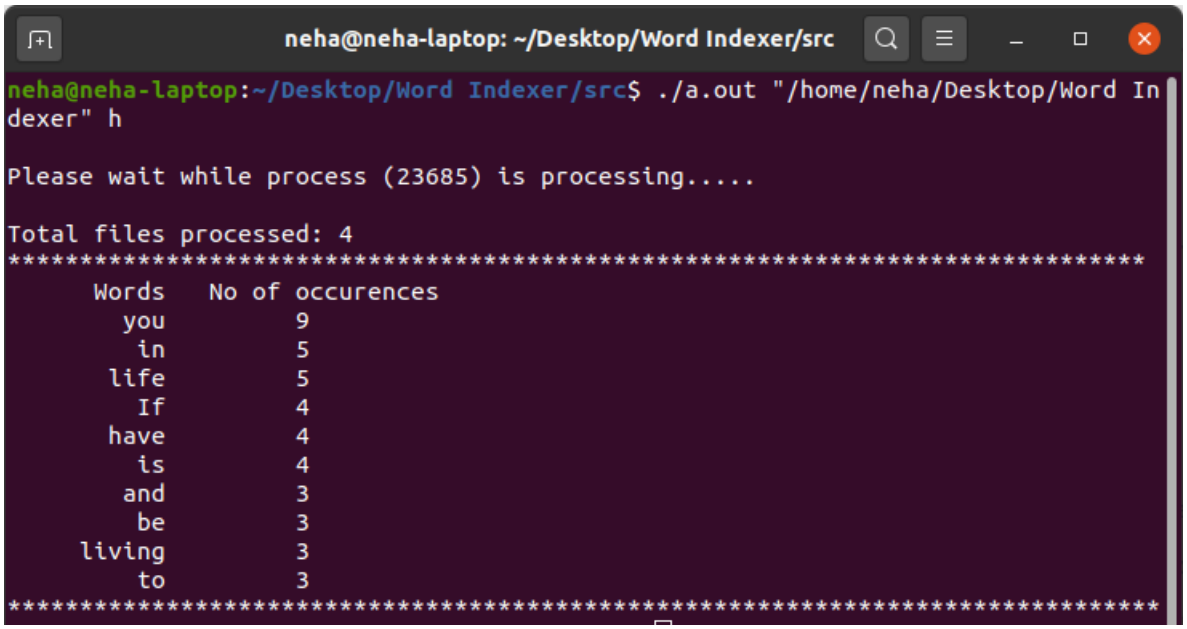
Development is done on Ubuntu 20.04 LTS using C++ 14 language

4.1 Directory Structure

```
Word Indexer -->|
                |--- src
                        |-- SearchThread.cpp
                        |-- SearchThread.h
                        |-- WorkerThread.cpp
                        |-- WorkerThread.h
                        |-- SyncQueue.cpp
                        |-- SyncQueue.h
                |--- Input Files
                        |-- file1.txt
                        |-- file2.txt
                        |-- file3.txt
                        |-- file4.txt
                |--- word_indexer.out
                |--- MakeFile
```

4.2 Output of the Program

* 'h' or head attribute output



*** 't' or tail attribute output**

```
neha@neha-laptop: ~/Desktop/Word Indexer/src
neha@neha-laptop:~/Desktop/Word Indexer/src$ ./a.out "/home/neha/Desktop/Word In
dexer" t

Please wait while process (23723) is processing.....

Total files processed: 4
*******************************************************************************
      Words    No of occurences
        -          1
       your          1
      would          1
    without          1
       with          1
       will          1
      which          1
       when          1
       were          1
         we          1
*******************************************************************************
```

*** output when no attribute is given**

```
neha@neha-laptop: ~/Desktop/Word Indexer/src
neha@neha-laptop:~/Desktop/Word Indexer/src$ ./a.out "/home/neha/Desktop/Word Indexer"

Please wait while process (23752) is processing.....

Total files processed: 4
*******************************************************************************
      Words    No of occurences
        you          9
         in          5
       life          5
         If          4
       have          4
         is          4
        and          3
         be          3
     living          3
         to          3
       what          3
        The          2
         at          2
       else          2
         it          2
         ll          2
       look          2
      never          2
      other          2
          s          2
          t          2
       time          2
        Don          1
       Life          1
       Your          1
          a          1
      above          1
     always          1
      begin          1
       busy          1
        but          1
         by          1
```

4.3 Glossary

- Mtable is a data structure which contains words with their frequencies.
- WorkerThread1, WorkerThread2 and WorkerThread3 are 3 worker threads which are part of the WorkerThread module and are responsible for filling up the Mtable.
- Queue is synchronized queue which contains files

Sample Input

file1.txt

The greatest glory in living lies not in never falling, but in rising every time we fall
The way to get started is to quit talking and begin doing.

file2.txt

Your time is limited, so dont waste it living someone else life. Don't be trapped by dogma – which is living with the results of other people's thinking.

file3.txt

If life were predictable it would cease to be life, and be without flavour
If you look at what you have in life, you'll always have more. If you look at what you don't have in life, you'll never have enough.

file4.txt

If you set your goals ridiculously high and its a failure you will fail above everyone else's success
Life is what happens when you're busy making other plans.