Super Simple File Indexer

Create a multi-threaded text file indexing command line application that works as follows:

1. Accept as input a file path on the command line. Please name the executable ssfi. For example, we'd be able to run your application like this: ./ssfi /usr/share/doc

2. Have one thread that is responsible for searching the file path, including any sub-directories, for text files (ending in '.txt')

3. When a text file is found, it should be handed off to a worker thread for processing, and the search thread should continue searching.

4. There should be a fixed number (N) of worker threads (say, N=3) that handle text file processing. Adding a command line option to specify N is recommended. For example, to specify worker threads, ./ssfi -t 3 /usr/share/doc

5. When a worker thread receives a text file to process, it opens the file and reads the contents to parse the words inside. Words are delimited by any character other than **A-Z, a-z,** or **0-9**. Words should be matched case insensitive.

6. Once the file search is complete and all text files finish processing, the program prints out the top 10 words and their counts. Please output the top 10 list of words as separate lines containing the word, followed by a tab, followed by the count. For example:

   oneword 1000
   another 850
   ...

7. A Makefile should be provided to compile your program. Please, no Cmake, Scons, Ant, bjam, Shell scripts, etc. A simple Makefile is all that is needed.

8. Any libraries used like Boost, should be available to any modern Linux distribution. Fedora, Ubuntu, Gentoo, Arch, CentOS, etc.

In summary - we want to find the top 10 words across a directory tree of text files.