

This program creates an inverted index by moving through the following steps:

1) Traverses through the directory and creates a global linked list of files. The nodes hold the filenames and a sequence number associated with the file. Different files of the same name have the same sequence number even though they have unique nodes associated with them. This is how the word counts for files of the same name are brought together when creating the inverted index.

2) A second traversal occurs where each file is turned into a long string, tokenized, and each word is added to a linked list of word\_nodes representing each word. Each word node holds an int array whose indices are associated with file sequence numbers and the count of the word in each file is held the index equal to the file's sequence number.

3) Once we have indexed and counted all tokens, we build a final output list. This output list is just the token list with an added field in each token node to point to another list that contains ONLY the files that that token occurs in along with its count in the file. This list is built in the invertedIndexer.c file and it is built in proper order with the counts being ordered from highest to lowest.

4) Finally we make our output file according to the proper format outlined in the assignment description using the output list.