Nikhil Mashettiwar
Alekhya Dulur
Systems Programming
Professor Russell

Search Assignment

Our search program reads a text file and tokenizes the file in order to get the file names. A linked list is used to store each word and its respective file names. The tokens "<list>" and "</list>" are used to distinguish new words and create a new node. Each node in the linked list stores the word as well as a pointer to the head of the linked list of files and a pointer to the next node in the linked list.

Whenever "so" is prompted, the program calls the "searchor" function. This function uses a linked list to store the searched filenames to be outputted. The input words are stored in a linked list, so each one of these words is found in the global linked list of all words found in the indexed file. Once found, all the files of this word are stored in the linked list of searched words. Then, the next input word is found in the global linked list and it's filenames are retrieved. If a particular filename of this word already exists in the linked list of output words then it is not added; if a filename does not exist, it is added to the end of the linked list. This search of words and filenames is repeated for all the input words. Once completed, the filenames are output on to the console.

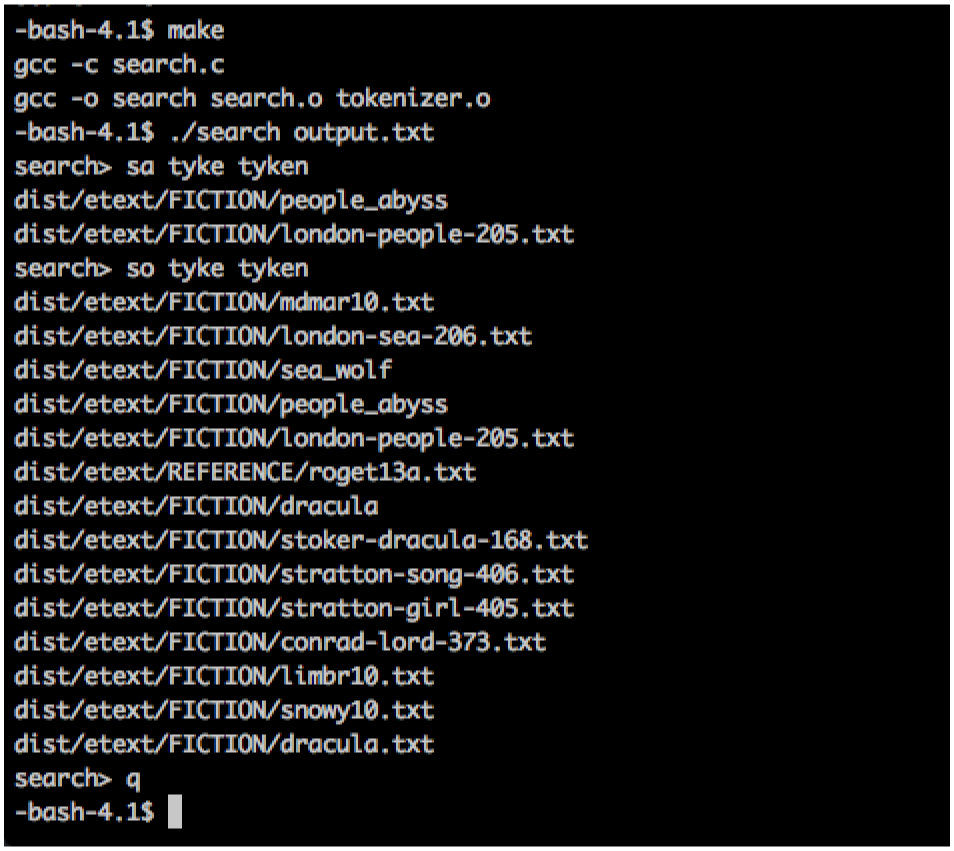
If "sa" is prompted, the "searchand" function is called. This function takes in the first input word and puts all its filenames into the linked list of output words. Then, all succeeding input words are one by one searched for in the global linked list of all words from the indexed file. Each file in the linked list of output words is compared to every file of a specific word. If a match is found, then the file is kept in the linked list of output words. If a match is not found, then that file is not common to both words and hence has to be deleted from the linked list of output words (When doing an AND search, a filename has to be common to all the input words, so when a filename is not matched it is deleted from the linked list of output words). Because this function needs to check each input word with each file of every word in the indexed file, it has an efficiency of O(n^4). This does not seem very efficient for our program but this was the only way we could compare each one of the filenames.

Our program prompts the user to enter either “sa” or “so” along with words to be searched. What our program does is tokenize the string that is input and retrieve the first word; depending on that a specific function will be called. If “q” is entered, the search engine is stopped and the program will terminate. The input string is stored in a linked list where each node stores the word to be searched.

Every node that is created and every linked list that is created is destroyed and freed at the end of the program, so there shouldn’t be any memory leaks.

We have attached a file, output.txt, which we tested our program with. The “searchor” and “searchand” functions only print out to the console when filenames are found. If the program prompts “search>” again, the user should know that no filenames in common have been found (only applies to the “and” search because there will always be files for the “or” search). If the input words do not exist in the linked list of words in the indexed file, a statement will be printed prompting to only enter words that exist in the indexed file.

Example of running our test file:



```
-bash-4.1$ make
gcc -c search.c
gcc -o search search.o tokenizer.o
-bash-4.1$ ./search output.txt
search> sa tyke tyken
dist/etext/FICTION/people_abyss
dist/etext/FICTION/london-people-205.txt
search> so tyke tyken
dist/etext/FICTION/mdmar10.txt
dist/etext/FICTION/london-sea-206.txt
dist/etext/FICTION/sea_wolf
dist/etext/FICTION/people_abyss
dist/etext/FICTION/london-people-205.txt
dist/etext/REFERENCE/roget13a.txt
dist/etext/FICTION/dracula
dist/etext/FICTION/stoker-dracula-168.txt
dist/etext/FICTION/stratton-song-406.txt
dist/etext/FICTION/stratton-girl-405.txt
dist/etext/FICTION/conrad-lord-373.txt
dist/etext/FICTION/limbr10.txt
dist/etext/FICTION/snowy10.txt
dist/etext/FICTION/dracula.txt
search> q
-bash-4.1$ 
```