Rizwan Chaudry

Read me

My indexer program works by taking in a file, tokenizing the file, then takes those tokens and stores them in a file record struct.

This file record struct contains the file name, the token, and the frequency of that token. This file record is then stored in a node and put into an array. My array which I call hashmap is an array of linked list pointers.

I decided to use this implementation rather than a hash table because the previous assignments we did coincided extremely well with this current assignment. This implementation also makes it quite easy to output the information in alphabetical order. I believe to be more efficient next time I should use a hash table.

My program is not as efficient as I would like since I am looping through the entire array in order to then search each linked list and then output that information to file. Also not every slot in the array may contain a linked list which causes there to be unused space. In the future it might be better to malloc space based on the number of lists and the realloc to accommodate more lists being made.

Random errors I encounter during my testing phase where when I tried to check if the file the user entered already existed. When I prompt the user to choose whether to rewrite the file or not if the user chose yes my program would segfault. I'm not sure exactly why this happened. Also when outputting the information, previous runs of the programs output would get appended to the new output file. I'm not sure if this is a bug with my program or just a random glitch.