Bilal Quadri, Yvgeniy Demo, Aiser Sheikh

Our indexer is designed with one sorted linked list and one unsorted linked list. One list being the words in sorted order and each word contains a list of the files it appears in. We call the getDelims function to get every character that’s not alphanumeric. We used ftw function to recursively iterate through the folders and executes our call function which adds the nodes to our lists. fileCallback opens the file allocates memory for the line and word, while there is a line loop through all the words spaced out by our delims and once it finds a word it calls the addCount function. addCount which makes sure that the wordlist exists, if not it allocates and creats the list. Then it checks if it needs to create a new Head for the list. Otherwise it will loop through the word list and check to see if the word already exists and finds the proper filename and increases the count. If the word doesn’t exists it adds it to the wordlist and creates and new filelist node for it. Afterwards we print out all the data in the proper format and free all of our memory. The code will run in K operations to run through each word W (total number of words). The the wordlist needs to be sorted with the number of unique words which will be U. The list will insert in sorted order so it will be linear time through U. The total runtime is O(KWU).