Zachary Blanco
CS214 - Systems Programming
Section: 04

Assignment 0: String Tokenizer/PointerSorter

Functions

The most notable functions for the implementation in this assignment are:

- char* createString
    - Given the start and end indexes in a character pointer, this will create a new character pointer for a separate substring which is then returned
- void AddToTree
    - This method will, given a string (char pointer) add a value to a Binary tree (unbalanced). There is a struct named **BNode** defined within the file to use with this function.
    - Also note this method is defined receursively.
- void traverseAndFree
    - This will traverse a binary tree and print each of the values in order successively and then free the memory allocated each node.
- int isAlpha
    - Determines if a single character is alphabetic or not.

Description

The program first starts by creating an empty binary tree and a start/end index initialized to 0. We then read in the characters one by one . When we hit a non-alphabetic character(isAlpha) the program copies the string (createString) between the two indexes into a new pointer, and inserts it into a simple Binary tree (addToTree). After which is resets the indexes to the current location and continues on to the rest of the string.

After iterating through all of the characters to the end of the string, the program finishes by first making sure the final string is copied into the binary tree which wasn't copied when the loop exits.

Then before we traverse and print everything we check to make sure the binary tree is not empty. If that check passes we then traverse the tree, printing all of the node's data and then free the corresponding memory (traverseAndFree).

Features

Code is fully documented. Functions each have descriptive comments including behavior, arguments, and return values. Worst case runtime is O(n^2) which stems from BTree insertions on an already sorted list of words. Best case is O(nlog(n)) where the insertions result in a balanced BTree.