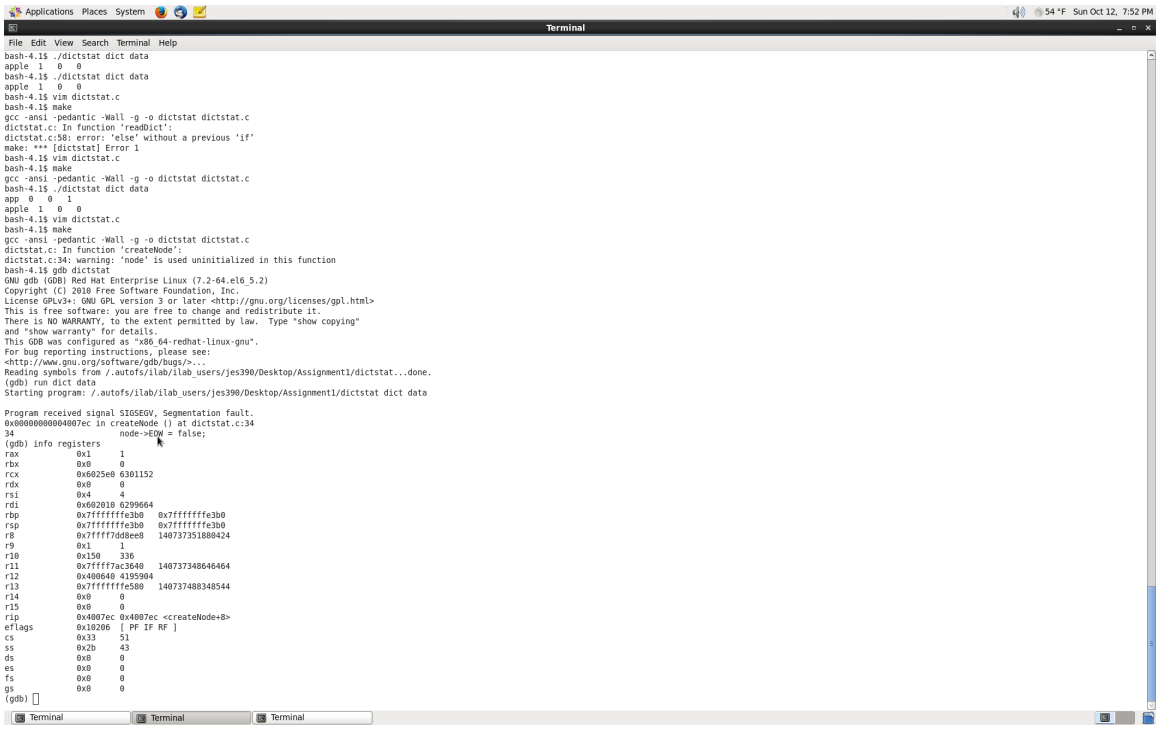

BIG O Notation

The struct I used is a trie structure. Variables are defined as 'n' words and 'm' characters. It takes O(n*m) time referencing to the amount of characters that are being used in this structure.

We search the words, which takes 'm' times. For every 'm' words we do 'n' counts for each character making it O(n*m)

To add words to the structure we go character by character starting from the root node and traversing down for every unique character the structure sees. If we come across a character that already exists as a child node we set our pointers equal to that node rather than creating a new one. Once the structure is built we look through our data file and find the words and compare them to our dictionary file by looking at the trie structure and DFSing it. I update the occurrence prefix and superWord by updating as I DFS. Memory is at most O(n*m) because of the malloc size of the nodes which holds the words and characters.

GDB

For GDB I had a segmentation failure. A seg fault is an error with compiling because my program was trying to access a piece of memory that didnt exist. To fix

this I used GDB on the dictstat file and I run the program and used break points to find out what my data looks like at certain points of my code. I used info registers command to see where in my register the data was. I didn't see it there so I knew I needed to malloc some piece of memory and I found my error

Challenges

The biggest challenge I faced during this project was learning the syntax for c90. Its very specific so when I got warnings I was confused on how to actually fix them. Using online resources taught me a lot about C programming