Rushit Sanghrajka

Ethan Berroa

**CS 214: Programming Assignment 1**

**Tokenizer in C**

Theoretically, a tokenizer is a core systems tool that accepts text, or blocks of code as it’s input, and breaks the text up into meaningful elements for the computer to understand, called tokens. The tokens that are generated by the tokenizer are further used for processing and parsing by the compiler.

The tokenizer that we have implemented is a basic version of the tokenizers present in current day compilers, however, we have implemented the core features of a basic tokenizer.

Structure of our Tokenizer:

The tokenizer basically takes in the text, and creates a dynamic Tokenizer struct. This tokenizer struct stores the text in it. We then pass this Tokenizer instance to the getNextToken method. In the getNextToken, we scan through the string to recognize and separate a token from another. The getNextToken returns the token that is found. In order to find tokens that are not separated to each other by a whitespace, we make the following assumptions:

* If a string starts with special symbols but is followed by alphabets or numbers, we break the special symbols into a separate token.
* A sequence of numbers followed by alphabets/ special symbols is still considered as one token.
* Multiple special character symbols adjacent to each other but not mentioned in the C reference card are assumed as malicious. For example, “?>)” would be recognized as malicious, but “+=” would be recognized as “plus equals”.
* If a string that started with alphabets encounters a special symbol, it will consider the alphabets as a separate token. For example, in “Bread, butter and cheese” we would split the “Bread” and the comma sign.
* However, incase of alphanumeric tokens with an adjacent special symbol, it would consider everything as one token.

In order to separate these cases, we created an internal FSM structure to separate these tokens before trying to find their type.

Once the token has been returned by getNextToken, we pass that token to getTokenType.