

readme

Jimmy Le
netid: jll1415
CS211 - SUMMER 2015

PROJECT: PARTIAL TOKENIZER

File: tokenizer.c

Description:

-Program that tokenizes strings, and targets octal, hexadecimal, decimal, and float types. The program prints out each of the individual numbers/characters as either octal, hexadecimal, decimal, and floats.

-Reads and extracts each token by reading for white spaces such as space and tab.

-Program may also read special characters such as parenthesis and brackets, however must provide error messages in hexadecimal form of the special case.

Assumptions/Concerns:

- Since all form of the char "0" can be represented as octal and decimal, I have assumed that it may be zero (a new category) by itself.

-Provided cases for just 0 such as "0" and " 0 ", which will return by default as Zero "0".

-However, any other version with just zero, will return zero.

-For the special char cases such as parenthesis (), I have provided hexadecimal forms of the special characters, which will print as an output after printing out any tokens before it.

CASE 1:

```
./tokenizer " 0x46 1234 0x() "  
OUTPUT: hexadecimal "0x46"  
        decimal "1234"  
        Error [0x29]
```

**This only works if the special char is attached to a numerical value such as floats, octals, decimals, and hexadecimal.

CASE 2:

```
./tokenizer " () 0x2b "  
OUTPUT: Error [0x28]  
        Error [0x29]  
        hexadecimal "0x2b"
```

**THIS REQUIREMENT WAS IN ASSIGNMENT INTRODUCTION.

CASE 3:

```
./tokenizer " abc () "  
OUTPUT: Error "abc"  
        Error [0x28]  
        Error [0x29]
```

Problems:

-Memory management were very problematic on how to save memory when cycling through each token.

-Problematic switch statements: too many to evaluate.

-Assumptions for error outputs with no specification of case by case value were problematic.

-The most problematic part of the project was the White Characters.

-isspace takes care of the initial '\', however trying to define \n and the others, were problematic since \ is always associated with white space, so it could not be defined properly.

-therefore you will see error messages for letters such as n,f,d, and others.

Features:

- Identifiable Error Error Messages:
 - Identifies Chars, Words, and Operators
 - Prints Hex Representation if Char or Operators

Analysis of Program:

- O NOTATION: $O(n)$