Jeff DiPaola
CS 214
9/24/14

Assignment 1 – Readme.pdf

The tokenizer.c file takes two string arguments, one a string to be traversed and the other a collection of one character sized symbols. The program should then read through the input string looking for the symbols and removing them. Each time a separator symbol is found everything before that symbol should be printed on a single line as a "token". The printed part of the input string should no longer be referenced and the tokenizer should continue traversing and repeating the process until each part of the input string has been output on its own line. Additionally, each token should be checked for escape characters *that were not separators* before being printed and separators should be replaced with their appropriate hex value between square brackets.

In order to do all of this the tokenizer.c file contains several methods, each which I've briefly described below:

TKCreate creates an object consisting of both the input string and the possible separator characters. This allows us to use the new object in other places to avoid editing the input data. It also contains a second pointer that references our location in the input string for traversal purposes.

TKDestroy removes the tokenizer object in its entirety from memory to avoid memory leakage. This is done by first removing the three sub elements in the tokenizer and then finally removing the tokenizer itself.

FindTheSymbol is a method I implemented that is used in the "TKGetNextToken" method to make it a bit cleaner. FindTheSymbol searches for the first/next instance of any of the symbols originally given to us in string that is passed to it. It does this via utilizing nested loops to traverse through the string and compare each character to all the possible symbols.

EscChk is called in the code from main after TKGetNextToken returns it's token. This is the point where we check the token we are about to output for escape characters. If it contains any we replace that escape character with its proper hex replacement in between square brackets. This method is pretty much just a giant nested if statement, however, it checks for escape characters of both types. I wrote this method before the announcement was made about escape characters it checks for both escape characters of the type '\n' and of the type '\','n'.

TKGetNextToken does exactly what it says, gets the next token. Basically, where "x" is a separator and "xtokenxtoken2xtoken3" is your input string, TKGetNExtToken would return "token" on its first call, "token2" on its second, etc.

Main is the first method that runs and is responsible for first checking that the number of arguments passed to the function is correct. It then converts the input into something that the program can read/use and calls the appropriate functions. Finally, before outputting the result it calls EscChk to be sure there are no escape characters in the token.