System Programming Homework 1 Report

First, I checked whether the number of arguments is valid or not. If it is valid, I splitted the command line argument with '/' separator. Then I checked the correction of each splitted element. If it is valid, program loop starts.

For each step, a specific number of characters is read from the input file. For this string of characters replace operation(s) occur(s).

How program loop works:

Each replace operation's replaced string, inserted string and case sensitivity is saved to array. When this arguments are ready, the string which is read from file is started to be searched for given sub-string. For each string read from file, this operation repeats.

How replace operation is handled:

For each replace operation, the string which is about to be replaced is searched in string read from file. StartandEndIndexes function returns the start and end indexes of found sub-strings. Then strReplace function replaces the sub-string, which is between the given start and end indexes, with wanted string.

How StartandEndIndexes function finds start and end indexes of given sub-string:

For case a) '/str1/str2/'

Brute force sub-string search algorithm is used.

For case b) '/str1/str2/i'

If the read characters are not the same, it is checked if the case is different but the letter is the same. If it is, then function continues from next elements.

For case c) '/str1/str2/i;/str3/str4/'

Program loop calls the same function for each replace operation again.

For case d) '/[zs]tr1/str2/'

When the [operator is encountered, each letter which is between square brackets is compared to the current element in the string which is read from the file. To be able to remember which character is expected next if we encounter with the * character next, found character is saved.

For case e) .'/^str1/str2/'

When the ^ operator is encountered, we checked if the character in the string is the first character of the file or, current element is \n. If it is the case, then function continues from the next element.

For case f) ‘/str1$/str2/‘

When the $ operator is encountered, we checked if the character in the string is \n. If it is the case, then function continues from the next element.

For case g) ‘/st*r1/str2/‘

When the * operator is encountered, there is 3 possibilities.

1) Last element is not in the string read from file.

When this is the case, last character is not matched with the character in the string. We checked the next element is * operator or not. If it is, then then function continues from the next element.

2) Last element is not followed any character which is same with it.

Program continues as if * operator does not exist.

3) Last element repeats

Index which is in the replaced string is incremented until the last same element.

I used a temporary file to save refined strings so that program does not have to hold all file in the main memory. When all file is processed. I copied the content of the temporary file to the given file.

Requirements:

Case 1)

Program does not work happens when the buffer cuts the searched sub-string from half. When the program detects that situation may happen, it prints this message to stderr:

Buffer may be cut the searched word from half. Unfortunately, this feature is not implemented as mentioned in the report.

Case 2)

To make $ operator work for the last line, additional new line is needed at the end of the file. I could not read EOF character from the file, that’s why $ operator works only for \n character.