CSE344 – SYSTEM PROGRAMMING

HW1 REPORT

Introduction
This is a simple text editor to replace all occurrences of a string with a desired string in a text file via using system calls.

Usage
The compilation can be done with “make” command.

User of the program must run the executable with 2 more argument. Otherwise program will not be executed. First argument is a string replacement command and the other is the input file path which can be relative path or absolute path.

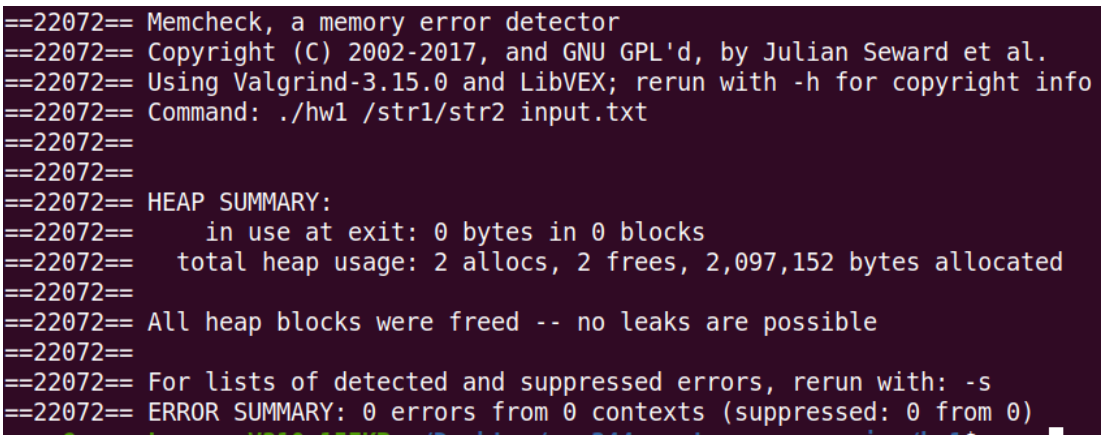
The input file will also be used as output file.

Program Design
The program takes given arguments and uses them to decide how to manipulate the given input text file. Details of the valid commands are given in the homework document by the professor of the course.

I always checked the system calls if they are executed correctly. If they are not, the program gives error relatively and terminates itself with a return value of 1.

First of all I read the input file and copied its content to a buffer string “buf”. Then I created a new buffer string called “new_buf” for the manipulated text. I copied the correctly manipulated string to this new buffer string and write back it to the input file.

After all the necessary operations are done, I set free all the used heap memory and closed the open file descriptor. There is no memory leak in our program and it is checked using Valgrind.



String Replacement
The p pointer points to the fist occurrence location of the str1 in the given input string. Cursor follows the input string as manipulated and helps to keep track of where we left at manipulation of the string. Manipulation continues from the cursor to the end of the file.

From cursor to the occurrence of the str1 copied to the new buffer. After that replaced string str2 is put in the place of str1. In front of the newly replaced string manipulation goes on with the same logic.

For case insensitive replacement, I used strcasestr instead of strstr function from the string.h library.

To be able to take multiple commands at the same line I used strtok_r with ";" delimiter. Why I used strtok_r instead of strtok? Because I also wanted to simultaneously tokenize command string with "/" delimiter to take str1 and str2 parameters. To continue this process for the other commands, I used a do-while loop.

Bugs & Results
In my consideration and tests, there are no bug and program runs normally. Of course there might some problems that I missed.

I completed the a, b, and c parts of the homework. Rest is not implemented because of my poor time management.

Further, the rest of the homework should be implemented and the multi-process lock handling should be added.