Lab 8: File Handling

CS2110 Programming Lab

October 12, 2014

Introduction

Have a look at *file_ handling.c* . Go through the code before attempting the questions. Also I think the assignment is pretty clear.Do point out if you feel the test case output files are wrong but refrain from asking irrelevant doubts.

1 Find and Replace [50 marks]

It's literally a find and replace problem. So you need to process the input file and write to the output file. When you see a word which needs to be replaced, you need to write the new word to the output file instead of the old one.

1. Take command line arguements which should be as follows.

$$./main1 \ < flag > \ < input\_file > \ < old\_word > \ < new\_word > \ < output\_file >$$

2. If the flag is set to 1 it means that the operation has to be case sensitive, 0 denotes case insensitive.
3. Strtok can be useful while parsing input file.
4. Check sample files in 'replace' folder for the required output.
5. Check the README file in the 'replace' folder and undergo the given operations on the input files to get output file.
6. Use 'diff' to check your output with the given output files. Use the man page for it and look into the flags (i,b,B,w). Remember that diff is a line-oriented program and a newline character always ends a line.

2 Indent [50 marks]

Indent the given C code. To make it simpler assume that indentation should be done only after a curly bracket opening.

1. Take command line arguements which should be as follows.

$$./main2 \ < input\_file > \ \ < output\_file >$$

2. Assume 4 space (or tab space) indentation (not 8).

3. The challenging part of this question is how you will take care of semi colons. Assume that semi colons either come at end of a code line or inside a 'for' loop.

   Hint: When you see the word 'for', set a flag, such that for the two semi colons after that ,you will consider them as part of the 'for' loop. Else if the flag is not set , it means that the semi colon is denoting end of a code line.

   This part is for *10 marks*. So if the first two test cases pass you still get marks. Only the third input file has 'for' loops.

4. Assume that the header is already indented.

5. The opening curly brackets can be printed at the end of earlier line or in a new line in the output. The sample test files have printed them on a separate line.

6. You don't need to take care of any extra cases other than the ones in the test files. If the test files run, it is good enough. Don't post doubts on the forum other than the ones about the test files.

7. Check sample files in the 'indent' for the required output.

8. Use 'diff' to check your output (Need not use diff, just view your output file and see if it looks indented!).