CS2263 Assignment 3

- Rinor Komorani
- March 1st 2024
- Przemyslaw Rafal Pochec

Table of Contents

Question 1

For my **htags2** program, the logic of finding an opening tag, and continuing to do a certain operation until a closing tag was met; was the same as **htags1**.

The big difference here is that -- instead of printing characters one-by-one until a closing '>' tag was met, I instead stored each character into a **char array**.

This char array had an index to keep track of how many characters were being stored. It started counting once an opening tag (*and the character after that is not '!'*) was met, and finished counting once a closing tag was met. I then incremented the index once more to store the null-terminating '\0' character. The index would then be reset to 0 to continue reading new words with each iteration of a tag.

The char array to store words would continiously be compared with an array of strings (*a 2D array*) to see if the word currently stored from the char array already exists in this 2D array. If not, it would add this word accordingly, with an index of it's own. If it already existed -- it simply would not add the word.

Question 2

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>
#include <ctype.h>
#define MAX 100

void readFile(char *filename)
{
    FILE *file = fopen(filename, "r");

    if (file == NULL)
    {
        printf("ERROR: Could not open file\n");
        return;
    }

    // *** attempting to read inbetween htags ***
    bool inTag = false, inComment = false;
    int index = 0, unique_words_count = 0, opener = '<', closer = '>', c;
    char word[MAX] = "", unique_words[MAX][MAX] = {" "};

    while ((c = fgetc(file)) != EOF)
    {
        // inside opening tag
        if (c == '<')
        {
            c = fgetc(file);

            // check if next character is in comment or not
            if (c == '!')
            {
                inComment = true;
                continue;
            }
            else
            {
                inTag = true;
                word[index++] = opener;
                word[index++] = c;
                continue;
            }
        }

        else if (c == '>')
        {
            if (inComment)
            {
                inComment = false;
                continue;
            }
            else
            {
                inTag = false;
                word[index++] = closer;
                word[index] = '\0';

                // check if the word is unique
                bool found = false;
                for (int i = 0; i < unique_words_count; i++)
                {
                    if (strcmp(word, unique_words[i]) == 0)
                    {
                        found = true;
                        break;
                    }
                }
                // store unqiue word in array of strings (2D)
                if (!found)
                {
                    strcpy(unique_words[unique_words_count++], word);
                }

                // reset word index to store next word inbetween tags
                index = 0;
            }
        }

        // still in word -- continue storing characters
        if (inTag && !inComment)
        {
            word[index++] = c;
        }
    }

    // print unique words
    printf("Unique words between HTML tags:\n");
    for (int i = 0; i < unique_words_count; i++)
    {
        printf("%s\n", unique_words[i]);
    }
}

int main(int argc, char **argv)
{
    if (argc != 2)
    {
        printf("ERROR: Provide a file to read.\n");
        return EXIT_FAILURE;
    }

    readFile(argv[1]);
    return EXIT_SUCCESS;
}
```

Question 3

```
rinor@rinors-mac Assgn3 % ./readInTest test.html
Unique words between HTML tags:
<html>
<b>
</b>
</html>
rinor@rinors-mac Assgn3 %
```

Question 4

```
rinor@rinors-mac Assgn3 % ./readInTest A3W2024.html
Unique words between HTML tags:
<html>
<head>
<meta http-equiv=Content-Type content="text/html; charset=windows-1252">
<meta name=Generator content="Microsoft Word 15 (filtered)">
<style>
</style>
</head>
<body lang=EN-CA link=blue vlink="#954F72" style='word-wrap:break-word'>
<div class=WordSection1>
<p class=MsoNormal>
<b>
<span lang=EN-US style='font-size:14.0pt;font-family:
"Times New Roman",serif'>
</span>
</b>
</p>
<p class=MsoNormal style='text-align:justify'>
<span lang=EN-US
style='font-family:"Times New Roman",serif'>
<u>
<span lang=EN-US style='font-family:"Times New Roman",serif'>
</u>
<a href="https://www.educba.com/types-of-tags-in-html/">
<span style='text-decoration:
none'>
</a>
<span style='font-family:"Times New Roman",serif'>
<p class=MsoBodyTextIndent style='margin-top:6.0pt;margin-
right:0cm;margin-bottom:
0cm;margin-left:0cm;margin-bottom:.0001pt;text-indent:0cm;line-
height:12.0pt'>
<i>
```

```
<span
style='font-family:"Times New Roman",serif'>
</i>
<p class=MsoBodyTextIndent style='margin-top:6.0pt;margin-
right:0cm;margin-bottom:
0cm;margin-left:7.1pt;margin-bottom:.0001pt;text-indent:0cm;line-
height:12.0pt'>
<span
lang=EN-US style='font-family:"Times New Roman",serif'>
<span lang=EN-US style='font-family:"Courier New"'>
<span lang=EN-US
style='font-family:"Courier New"'>
<span
lang=EN-US style='font-family:"Courier New"'>
<span
lang=EN-US style='font-size:14.0pt;font-family:"Times New Roman",serif'>
<span lang=EN-US style='font-size:14.0pt;
font-family:"Times New Roman",serif'>
<p class=MsoBodyTextIndent style='margin-top:6.0pt;margin-
right:0cm;margin-bottom:
0cm;margin-left:43.1pt;margin-bottom:.0001pt;text-indent:-18.0pt;line-
height:
12.0pt'>
<span
lang=EN-US style='font-size:7.0pt;font-family:"Times New Roman",serif'>
<span lang=EN-US style='font-family:
"Courier New"'>
<p class=MsoBodyTextIndent style='margin-top:6.0pt;margin-
right:0cm;margin-bottom:
12.0pt;margin-left:43.1pt;text-indent:-18.0pt;line-height:12.0pt'>
<br>
<p class=MsoBodyTextIndent style='margin-top:6.0pt;text-indent:0cm;line-
height:
12.0pt'>
</div>
</body>
</html>
rinor@rinors-mac Assgn3 %
```