CS2263 Assignment 4

- Rinor Komorani
- March 27th 2024
- Przemyslaw Rafal Pochec

Table of Contents

Question 1

For my **htags3** 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 **htags2**.

The big difference here is that instead of using fixed data sizes, dynamic allocation for size was used on the heap.

I allocated my memory for **2** data structures in my program. The first data structure consisted of a **1D array** that stored enough space for a singular tag *(including a null terminating character)*

I would keep track of the space allocated each time a *'<'* opening tag was met, and continuted counting the amount of space needed for 1 char until a *'>'* closing tag was met. The count was reset once a new opening *'<'* was met.

This way of counting helped me when I allocated space for my other data structure, a **2D array**. I used **realloc()** for my rows in my data structure, adding memory for **1 char pointer** if a tag was not yet seen (*using strings.h for comparasion*). I used **malloc()** for each column of that new row to store the tag. The size I used to allocate that column memory came from my *1D array* size counter.

Question 2

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

// reads and frees elements of 2D array
void readArray(char ** array, int length)
{
    printf("\n");
    printf("Unique words between HTML tags:\n");
    for (int i = 0; i < length; i++)
    {
        printf("%s\n", array[i]);
        free(array[i]);
    }
    free(array);
    printf("\n");
}

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

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

    bool inTag = false, inComment = false;
    int newAllocation = 0, c, totalAllocation = 0, totalToPrint = 0, index
= 0, opener = '<', closer = '>', unique_words_count = 0;
    char * word = NULL, ** uniqueWord = NULL;

    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
            {
                // reset new word memory allocation
                newAllocation = 0;

                newAllocation += 2;
                totalAllocation += 2;

                inTag = true;

                // allocate new memory on heap of first 2 valid read-in
chars
                // for the temp array
                word = malloc(newAllocation * sizeof(char));

                word[index++] = opener;
                word[index++] = c;
                continue;
            }
        }

        else if (c == '>')
        {
            if (inComment)
            {
                inComment = false;
                continue;
            }
            else
            {
                newAllocation += 2;
                totalAllocation += 2;

                word = realloc(word, totalAllocation);

                inTag = false;
                word[index++] = closer;
                word[index] = '\0';

                // printf("Current Allocation of tag %d bytes\n",
newAllocation);

                // check if the word is unique
                bool found = false;
                for (int i = 0; i < unique_words_count; i++)
                {
                    // found word in 2D array
                    if (strcmp(word, uniqueWord[i]) == 0)
                    {
                        found = true;
                        break;
                    }
                }
                // store unique word in array of strings (2D)
                // using malloc (size of the current tag)
                if (!found) {
                    totalToPrint += newAllocation;
                    printf("Current allocated tag memory: %d\n",
newAllocation);

                    // row implementation
                    uniqueWord = realloc(uniqueWord, (unique_words_count +
1) * sizeof(char *));

                    // col implementation
                    uniqueWord[unique_words_count] = malloc(newAllocation
* sizeof(char));

                    // adding word to 2D array
                    strcpy(uniqueWord[unique_words_count], word);
                    unique_words_count++;
                }

                // reset counters & free temp word array
                index = 0;
                free(word);
            }
        }

        // still in word -- continue storing characters
        if (inTag && !inComment)
        {
            newAllocation++;
            totalAllocation++;

            word = realloc(word, totalAllocation);

            word[index++] = c;
        }
    }

    // print unique words & free uniqueWords allocation
    readArray(uniqueWord, unique_words_count);
    fclose(file);

    printf("Total allocated memory: %d bytes\n", totalToPrint);
    printf("\n");
}

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 Assgn4 % ./htags3 test.html
Current allocated tag memory: 7
Current allocated tag memory: 4
Current allocated tag memory: 5
Current allocated tag memory: 8

Unique words between HTML tags:

<b>
</b>
</html>

Total allocated memory: 24 bytes

rinor@rinors-mac Assgn4 %
```

Valgrind output ensuring no memory leaks

```
ronald@ronald-ThinkPad-T480:~/Documents/CS2263/Assignment4$ valgrind
./htags3 test.html
==94344== Memcheck, a memory error detector
==94344== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==94344== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright
info
==94344== Command: ./htags3 test.html
==94344==
Current allocated tag memory: 7
Current allocated tag memory: 4
Current allocated tag memory: 5
Current allocated tag memory: 8

Unique words between HTML tags:
<html>
<b>
</b>
</html>

Total allocated memory: 24 bytes

==94344==
==94344== HEAP SUMMARY:
==94344==     in use at exit: 0 bytes in 0 blocks
==94344==   total heap usage: 32 allocs, 32 frees, 5,987 bytes allocated
==94344==
==94344== All heap blocks were freed -- no leaks are possible
==94344==
==94344== For lists of detected and suppressed errors, rerun with: -s
==94344== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
ronald@ronald-ThinkPad-T480:~/Documents/CS2263/Assignment4$
```

Question 4

```
rinor@rinors-mac Assgn4 % ./htags3 A4W2024.html
Current allocated tag memory: 7
Current allocated tag memory: 7
Current allocated tag memory: 73
Current allocated tag memory: 61
Current allocated tag memory: 8
Current allocated tag memory: 9
Current allocated tag memory: 8
Current allocated tag memory: 73
Current allocated tag memory: 25
Current allocated tag memory: 20
Current allocated tag memory: 4
Current allocated tag memory: 81
Current allocated tag memory: 8
Current allocated tag memory: 5
Current allocated tag memory: 5
Current allocated tag memory: 47
Current allocated tag memory: 63
Current allocated tag memory: 4
Current allocated tag memory: 62
Current allocated tag memory: 5
Current allocated tag memory: 57
Current allocated tag memory: 38
Current allocated tag memory: 5
Current allocated tag memory: 51
Current allocated tag memory: 163
Current allocated tag memory: 4
Current allocated tag memory: 52
Current allocated tag memory: 5
Current allocated tag memory: 165
Current allocated tag memory: 63
Current allocated tag memory: 52
Current allocated tag memory: 53
Current allocated tag memory: 53
Current allocated tag memory: 72
Current allocated tag memory: 172
Current allocated tag memory: 79
Current allocated tag memory: 54
Current allocated tag memory: 151
Current allocated tag memory: 5
Current allocated tag memory: 90
Current allocated tag memory: 7
Current allocated tag memory: 8
Current allocated tag memory: 8

Unique words between HTML tags:

<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"'>
<p class=MsoBodyTextIndent style='margin-top:6.0pt;line-height:12.0pt'>
<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>

Total allocated memory: 1982 bytes

rinor@rinors-mac Assgn4 %
```

Question 5

This is the output of the **htags2** program from the downloaded Assignment 2 page:

htags2 Output

```
rinor@rinors-mac Assgn4 % ./htags3 test3.html
Current allocated tag memory: 7
Current allocated tag memory: 7
Current allocated tag memory: 68
Current allocated tag memory: 61
Current allocated tag memory: 8
Current allocated tag memory: 9
Current allocated tag memory: 8
Current allocated tag memory: 70
Current allocated tag memory: 25
Current allocated tag memory: 20
Current allocated tag memory: 4
Current allocated tag memory: 80
Current allocated tag memory: 8
Current allocated tag memory: 5
Current allocated tag memory: 5
Current allocated tag memory: 47
Current allocated tag memory: 62
Current allocated tag memory: 4
Current allocated tag memory: 62
Current allocated tag memory: 5
Current allocated tag memory: 57
Current allocated tag memory: 36
Current allocated tag memory: 5
Current allocated tag memory: 51
Current allocated tag memory: 162
Current allocated tag memory: 4
Current allocated tag memory: 51
Current allocated tag memory: 5
Current allocated tag memory: 164
Current allocated tag memory: 62
Current allocated tag memory: 52
Current allocated tag memory: 52
Current allocated tag memory: 52
Current allocated tag memory: 168
Current allocated tag memory: 169
Current allocated tag memory: 5
Current allocated tag memory: 52
Current allocated tag memory: 170
Current allocated tag memory: 78
Current allocated tag memory: 150
Current allocated tag memory: 89
Current allocated tag memory: 7
Current allocated tag memory: 8
Current allocated tag memory: 8
```

```
Unique words between HTML tags:
=Content-Type content="text/html
ntent="text/html
<meta http-equive"
<meta name=Generator content="Microsoft Word 15 (filtered)">
<style>
</style>
</head>
<body lang=EN-CA link=blue vlink=purple 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"'>
<a
href="https://linuxcommand.org/lc3_lts0070.php">
<p class=MsoBodyTextIndent style='margin-top:6.0pt;margin-
right:0cm;margin-bottom:
0cm;margin-left:7.1pt;margin-bottom:.0001pt;text-indent:18.0pt;line-
height:
12.0pt'>
<p class=MsoBodyTextIndent style='margin-top:6.0pt;margin-
```

```
right:0cm;margin-bottom:
0cm;margin-left:25.1pt;margin-bottom:.0001pt;text-indent:18.0pt;line-
height:
12.0pt'>
<br>
<span lang=EN-US
style='font-family:"Courier New"'>
<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'>
<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'>
<p class=MsoBodyTextIndent style='margin-top:6.0pt;text-indent:0cm;line-
height:
12.0pt'>
</div>
</body>
</html>

Total allocated memory: 2222 bytes

rinor@rinors-mac Assgn4 %
```