CS2263 Assignment 3

Student:
Name: Will Ross
Number: #3734692
Email: will.ross@unb.ca

Date: March 2nd, 2024
Due Date: March 13th, 2024

Contents

Question 1

In a few sentences describe the design of the htags2 program. Focus on the description of the algorithm used and of any data structures needed to complete the task.

Answer Q1

For my htag2 program, we will iterate through the input given by a file. It will then check if the input is a valid file. If the file is valid, it enters a while loop until we reach the end of the file `EOF`. Then checks if the `currChar` is equal to `<`. If true, we then get the next char in the file and see if it equals `!`, if true that means we are inside an HTML comment tag so we set the variable `inComment` to true. If not, that means we are in a tag and set `inTag` to true. Then we store the `currChar` into the `words` array for later. If the `currChar` didn't equal `<` then we check if `currChar` equals `>` meaning the end of a tag. If true, we check if we are in a comment if so set `inComment` to false. If we were not in a comment, we set `inTag` to false and set the last char on the `word` array to `\0` to symbolise the end of chars in the array. We then set `found` equal to zero. `found` represents if we find the same tag in the array. We then loop through all the characters stored in our array checking if `word` and `unique[i]` are the same. If they are the same we then set `found` to true and break from the for loop. One broke we check if `found` does not equal true. If found is false we string copy the contents of `word` to `unique[uniqueCount]`, then reset the `index` to zero. Then we check if the following are true `inTag && !inComment && isalpha(currChar)`, if so we then set `currChar` equal to `word[index]`. Once the while loop hits `EOF` we then pass the unique array of char and the count into a printArr function.

Question 2

Show the complete source code.

Source code Htag2

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

//MAX Equals 150 for whole D2L site
#define MAX 100


void printArr(char arr[][MAX], int size){

    printf("Unique words between HTML tags:\n");
    for (int i = 0; i < size; i++)
    {
        printf("<%s>\n", arr[i]);
    }
}

void readFile(char *filename)
{
    //If we are in a tag
    bool inTag = false;

    //If we are in a comment
    bool inComment = false;

    //The index for storing tags
    int index = 0;

    //The count of unique chars
    int uniqueCount = 0;

    //Array for all words stored
    char word[MAX] = "";

    //Array for unique tags
    char unique[100][MAX] = {""};

    //The current value read in
    int currChar;

    //If we found duplicate
    bool found = 0;
```

```
    FILE *file = fopen(filename, "r");

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

    while ((currChar = fgetc(file)) != EOF)
    {

        //check if we are at a tag start
        if(currChar == '<'){

            //get next char
            currChar = fgetc(file);

            //check if we are at a comment tag
            if(currChar == '!')
            {
                //in a comment
                inComment = true;
                continue;
            }
            else
            {
                //in a tag and not a comment
                inTag = true;

                //add the current word to be returned
                word[index++] = currChar;
                continue;
            }
        }

        //if the current is at the end of a tag
        else if (currChar == '>')
        {

            //if was in a comment change to false
            if(inComment)
            {
                inComment = false;
                continue;
            }

            //if not in comment
            else{

                //set not in tag
                inTag = false;

                //set the end of the word array
                word[index] = '\0';
```

```
                found = 0;

                //loop through the word array to make sure all unique
                for (int i = 0; i < uniqueCount; i++)
                {
                    //check if they are the same
                    if (strcmp(word, unique[i]) == 0)
                    {
                        found = true;
                        break;
                    }
                }

                // If the word is unique add it to unique
                if (!found)
                {

                    strcpy(unique[uniqueCount++], word);
                }

                //reset index
                index = 0;
            }
        }
        //stores the strings of tags
        if (inTag && !inComment && isalpha(currChar))
        {
            word[index++] = currChar;
        }
    }
    printArr(unique, uniqueCount);

}

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

Show the output from running htags2 program on the following input HTML code (store it in a file):

```
<html> <b> TEST1 </b> <b> TEST2 </b> </html>
```

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

Question 4

Show the output from running htags2 program on this HTML file, i.e. the very file describing this assignment: you need to download this file separately from D2L to your computer (do not download the entire D2L web page!). The file name is A3W2024.html.

htag2.c output

NOTE: I had to change the Max size of the array to 150 so we could do the whole file

```
$ ./htags2 A3W2024.html
Unique words between HTML tags:
<html>
<head>
<metahttpequivContentTypecontenttexthtmlcharsetwindows>
<metanameGeneratorcontentMicrosoftWordfiltered>
<style>
</style>
</head>
<bodylangENCAlinkbluevlinkFstylewordwrapbreakword>
<divclassWordSection>
<pclassMsoNormal>
<b>
<spanlangENUSstylefontsizeptfontfamilyTimesNewRomanserif>
</span>
</b>
</p>
<pclassMsoNormalstyletextalignjustify>
<spanlangENUSstylefontfamilyTimesNewRomanserif>
<u>
</u>
<ahrefhttpswwweducbacomtypesoftagsinhtml>
<spanstyletextdecorationnone>
</a>
<spanstylefontfamilyTimesNewRomanserif>
<pclassMsoBodyTextIndentstylemargintopptmarginrightcmmarginbottomcmmarginleftcmmar
ginbottomptttextindentcmlineheightpt>
<i>
</i>
<pclassMsoBodyTextIndentstylemargintopptmarginrightcmmarginbottomcmmarginleftptmar
ginbottomptttextindentcmlineheightpt>
<spanlangENUSstylefontfamilyCourierNew>
<pclassMsoBodyTextIndentstylemargintopptmarginrightcmmarginbottomcmmarginleftptmar
ginbottomptttextindentptlineheightpt>
<pclassMsoBodyTextIndentstylemargintopptmarginrightcmmarginbottomptmarginleftpttex
tindentptlineheightpt>
<br>
<pclassMsoBodyTextIndentstylemargintoppttextindentcmlineheightpt>
</div>
</body>
</html>
```

Question 5

Show the output from running htags2 program on another HTML file, of your choice.

For the last test I used this websites source code Assignment 1 Report

htag2.c D2L Output

NOTE: I had to change the Max size of the array to 150 so we could do the whole file

```
willr@wills-bookpro MINGW64 ~/Documents/GitHub/Cs2263/Assignments/Assignment3
(main)
$ ./htags2 index2.html
Unique words between HTML tags:
<html>
<head>
<metahttpequivContentTypecontenttexthtmlcharsetunicode>
<metanameGeneratorcontentMicrosoftWordfiltered>
<style>
</style>
</head>
<bodylangENCAstylewordwrapbreakword>
<divclassWordSection>
<pclassMsoNormal>
<b>
<spanlangENUSstylefontsizeptfontfamilyTimesNewRomanserif>
</span>
</b>
</p>
<pclassMsoNormalstyletextalignjustify>
<spanlangENUSstylefontfamilyTimesNewRomanserif>
<pclassMsoBodyTextIndentstylemargintopptmarginrightcmmarginbottomcmmarginleftcmmar
ginbottomptttextindentcmlineheightpt>
<u>
</u>
<pclassMsoBodyTextIndentstylemargintopptmarginrightcmmarginbottomcmmarginleftptmar
ginbottomptttextindentcmlineheightpt>
<br>
<i>
<spanstylefontfamilyTimesNewRomanserif>
</i>
<pclassMsoBodyTextIndentstylemargintopptmarginrightcmmarginbottomptmarginleftpttex
tindentptlineheightpt>
<spanstylefontsizeptfontfamilyTimesNewRomanserif>
<spanstylefontfamilyCourierNew>
<pclassMsoBodyTextIndentstylemargintopptmarginrightcmmarginbottomcmmarginleftptmar
ginbottomptttextindentptlineheightpt>
<strong>
<spanstylefontfamilyTimes>
</strong>
<pclassMsoBodyTextIndentstylemargintoppttextindentcmlineheightpt>
<spanstylecolorred>
```

```
</div>
</body>
</html>

willr@wills-bookpro MINGW64 ~/Documents/GitHub/Cs2263/Assignments/Assignment3
(main)
$
```