Kartik Srivastava

Student ID – 3668516

Assignment 4

1. My program has 5 functions including the main(). display() just prints out the tags and how many times they were found. tagCounter() counts the tags and updates everytime the same tag is encountered. fileReader() reads the HTML file and checks for the tags and calls the tagCounter() to update the number of times the tag is found. memoryFreeing() frees the heap memory.

**htag.c**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define MAX\_SIZE 100000

#define TOTAL\_TAGS 100

#define MAX\_TAG\_SIZE 30

void display(char \*\*tagArr, int \*tagCounter)

{

    printf("Tag\tCount");

    printf("\n");

    for(int i = 0;i < TOTAL\_TAGS; i++)

    {

        if(tagCounter[i]!=-1)

        {

            printf("%s \t %d\n",tagArr[i], tagCounter[i]);

        }

    }

}

void tagCounter(char \*charac, char \*\*tagArr, int \*counter)

{

    int i = 0;

    int tag = 0;

    for(i = 0; i < TOTAL\_TAGS; i++)

    {

        if(tagArr[i] == NULL)

        {

            break;

        }

        else if(strncmp(charac, tagArr[i], strlen(charac)) == 0)

        {

            tag = 1;

            break;

        }

    }

    if (tag)

    {

        counter[i] = counter[i]+1;

    }

    else

    {

        tagArr[i] = charac;

        counter[i] = 1;

    }

}

void memoryFreeing(char \*ch, char \*\*tagsArr, int \*counting)

{

    free(ch);

    for(int i = 0;i < TOTAL\_TAGS;i++)

    {

        free(tagsArr[i]); //freeing each element of tagsArr array

    }

    free(tagsArr); //freeing tagsArr array

    free(counting); //freeing counting Array

}

int fileReader(char \*charac1, char \*\*tag, int \*counter)

{

    int len = 0;

    int num = 0;

    char \*tagPtr = NULL;

    int indexPos = 0;

   if ((len = read(0, charac1, MAX\_SIZE)))

   {

       for(int i = 0; i < len; i++)

       {

            if (num == 0)

            {

                //checking for HTML tags

                if(charac1[i] == '<')

                {

                    tagPtr = (char \*)malloc(MAX\_TAG\_SIZE);

                    memset(tagPtr, '\0', MAX\_TAG\_SIZE);

                    num = 1;

                }

            }

            else

            {

                if(charac1[i] == '>' || charac1[i] == '/'|| charac1[i] == ' ')

                {

                    num = 0;

                    indexPos = 0;

                    if(charac1[i]!='!')

                    {

                        tagCounter(tagPtr, tag, counter);

                    }

                    else

                    {

                        free(tagPtr);

                    }

                }

                else

                {

                    tagPtr[indexPos++] = charac1[i];

                }

            }

       }

   }

}

int main()

{

    char \*input=NULL;

    char\*\* tagPtr=NULL;

    int \*tagsCounter=NULL;

    input = (char \*)malloc(sizeof(char) \* MAX\_SIZE+1);

    tagPtr = (char \*\*) malloc(sizeof(char \*) \* TOTAL\_TAGS);

    tagsCounter = (int \*) malloc(sizeof(int) \* TOTAL\_TAGS);

    memset(input,'\0', MAX\_SIZE+1);

    for(int i = 0; i < TOTAL\_TAGS; i++)

    {

        tagPtr[i]=NULL;

        tagsCounter[i]=-1;

    }

    fileReader(input, tagPtr, tagsCounter); //reading the file

    display(tagPtr, tagsCounter); //printing the output

    memoryFreeing(input, tagPtr, tagsCounter); //freeing all the heap memory

    return EXIT\_SUCCESS;

}

**Outputs:**

A close up of a logo

Description automatically generatedA close up of a logo

Description automatically generated

**Testing a function**

htag2.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "htag3.h"

#define TOTALTAGS 100

int main(int argc, char\*\* argv)

{

    printf("Count: \n");

}

htag3.h

#define TOTAL\_TAGS 100

void tagCounter(char \*charac, char \*\*tagArr, int \*counter)

{

    int i = 0;

    int tag = 0;

    for(i = 0; i < TOTAL\_TAGS; i++)

    {

        if(tagArr[i] == NULL)

        {

            break;

        }

        else if(strncmp(charac, tagArr[i], strlen(charac)) == 0)

        {

            tag = 1;

            break;

        }

    }

    if (tag)

    {

        counter[i] = counter[i]+1;

    }

    else

    {

        tagArr[i] = charac;

        counter[i] = 1;

    }

}