**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Программирование»**

Тема: *Динамические структуры данных*

|  |  |  |
| --- | --- | --- |
| Студент гр. 6303 |  | Фокин К.С. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

**Цель работы:** требуется написать программу, моделирующую работу стека для определения валидности кода html-страницы.

**Ход работы:**

В ветке **fokin\_sem2\_lr2** репозитория **pr-2016-6303** создал папку **fokin\_sem2\_lr2** с проектом, состоящим из 1 файла ***main\_lr2.c***

Формулировка задания:

*Требуется написать программу, получающую на вход строку, (без кириллических символов и не более 500 символов) представляющую собой код "простой" html-страницы и проверяющую ее на валидность. Программа должна вывести****correct****если страница валидна или****wrong****.  
  
html-страница, состоит из тегов и их содержимого, заключенного в эти теги. Теги представляют собой некоторые ключевые слова, заданные в треугольных скобках. Например, <****tag****> (где tag - имя тега). Область действия данного тега распространяется до соответствующего закрывающего тега </****tag****> который отличается символом /. Теги могут иметь вложенный характер, но не могут пересекаться   
<tag1><tag2></tag2></tag1> - верно  
<tag1><tag2></tag1></tag2> - не верно  
  
Существуют теги, не требующие закрывающего тега.   
  
Валидной является html-страница, в коде которой всякому открывающему тегу соответствует закрывающий (за исключением тегов, которым закрывающий тег не требуется)  
  
Во входной строке могут встречаться любые парные теги, но гарантируется, что в тексте, кроме обозначения тегов, символы < и > не встречаются. аттрибутов у тегов также нет.   
Теги, которые не требуют закрывающего тега: <br>, <hr>  
  
Стек (который потребуется для алгоритма проверки парности тегов) требуется реализовать самостоятельно на базе****списка****.*

Ниже приведен код программы:

/\*

Программа main\_lr2.c (Л/р №2) определяет валидность кода html-страницы.

By foksen98 02.04.2017 (Last update 13.04.2017).

\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// максимальная длина входной строки

#define LENGTH 3000

// структура элемента списка

typedef struct stack

{

char\* tag;

struct stack\* prev;

} stack;

// функция добавляет в список структуру с именем последнего открывающего

stack\* push (char\* tag, stack\* current)

{

stack\* next = (stack\*)malloc(sizeof(stack));

next->tag = tag;

next->prev = current;

return next;

}

// функция удаляет последний элемент списка

stack\* pop (stack\* current)

{

stack\* temp = current->prev;

free(current);

return temp;

}

// функция извлекает значение последнего элемента списка

char\* top (stack\* current)

{

return current->tag;

}

// функция проверяет список на пустоту

stack\* isEmpty (stack\* current)

{

return current->prev;

}

// функция освобождает динамическую память (весь список)

void delete (stack\* current)

{

stack\* temp;

while (current->prev)

{

temp = current->prev;

free(current);

current = temp;

}

free(current);

}

int main()

{

char str[LENGTH + 2];

char \*ptrOpen, \*ptrClose, \*tag;

stack\* current = (stack\*)malloc(sizeof(stack));

current->tag = "null";

current->prev = NULL;

fgets(str, (LENGTH + 2), stdin);

ptrOpen = strstr(str, "<");

ptrClose = strstr(str, ">");

while (ptrOpen)

{

if (\*(ptrOpen + 1) == '/')

if (!strncmp((ptrOpen + 2), top(current), (ptrClose - ptrOpen - 2)))

current = pop(current);

else

{

delete(current);

printf("wrong\n");

return 0;

}

else

{

tag = strtok((ptrOpen + 1), ">");

if (strcmp(tag, "br") && strcmp(tag, "hr"))

current = push(tag, current);

}

ptrOpen = strstr((ptrClose + 1), "<");

ptrClose = strstr((ptrClose + 1), ">");

}

// если были закрыты все открывающие теги (осталась только голова списка),

// то выводим "correct"

if (!current->prev)

printf("correct\n");

// иначе "wrong"

else

printf("wrong\n");

// очищаем оставшуюся динамическую память

delete(current);

return 0;

}

1. Добавление в стек структуры с именем последнего открывающего (но требующего закрытия) тега реализовано при помощи функции **push()**.
2. Удаление последнего элемента стека выполняется функцией **pop()**.
3. За извлечение значения последнего элемента стека отвечает функция **top()**.
4. Функция i**sEmpty()** определяет, пуст ли стек.
5. Функция **delete()** отвечает за освобождение динамической памяти, выделенной под элементы списка.

Созданная папка с файлом загружена в репозиторий на **Github** с помощью следующих команд:

* git add fokin\_sem2\_lr2
* git commit –m “Fokin LR №2 done”
* git push origin fokin\_sem2\_lr2

**Вывод:** в ходе лабораторной работы изучены возможности применения динамических структур в языке Си. Полученные знания закреплены в ходе разработки программы **main\_lr2.c** для определения валидности кода html-страницы.