**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе №3**

**по дисциплине «Программирование»**

Тема: [**Строки. Рекурсия, циклы, обход дерева**](https://stepik.org/course/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5-(%D0%9B%D0%AD%D0%A2%D0%98)-1096/syllabus?module=12)

|  |  |  |
| --- | --- | --- |
| Студент гр. 6303 |  | Филиппенко Д.Р. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

**Цель работы:** научиться использовать рекурсивный обход дерева папок, найти файл, в котором содержится запись «Minotaur».

**Задание:**

**Лабиринт**

Дана некоторая корневая директория, в которой может находиться некоторое количество папок, в том числе вложенных. В этих папках хранятся некоторые текстовые файлы, имеющие имя вида *<filename>*.txt.

Требуется найти файл, который содержит строку "Minotaur" (файл-минотавр).

Файл, с которого следует начинать поиск, всегда называется file.txt (но полный путь к нему неизвестен).

Каждый текстовый файл, кроме искомого, может содержать в себе ссылку на название другого файла (эта ссылка не содержит пути к файлу). Таких ссылок может быть несколько.

**Ход работы:**

1. Подключены необходимые для работы заголовочные файлы

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <dirent.h>

#include <string.h>

Объявлена константа, размер строки

#define SIZE 20

2. Объявлена структура, содержащая строку, в которую записывается путь, и счетчик

typedef struct Path

{

    char\*\* array;

    int count;

}Path;

3. Объявлены и определены функции, необходимые для дальнейшей работы программы.

Функция findMinotaur – поиске минотавра среди файлов

int findMinotaur(const char \*, char \*, Path \*);

int findMinotaur(const char \*startdir, char \*filename, Path \*Pt) //на вход подается корневая директория, указатель на название файла, который требуется найти и указатель на структуру

{

    int log = 0; //логическая переменная

    char current\_path[1000];

    strcpy(current\_path,startdir); //копируем стартовую директорию в строку

    DIR \*dir = opendir(current\_path); //открываем директорию

    struct dirent \*de = readdir(dir); //создаем структру, считываем директорию

    if(dir && !log) //если директория не пуста

while(de)

        {

         if( de->d\_type == DT\_DIR &&

             0!=strcmp(".",de->d\_name) && 0!=strcmp("..",de->d\_name)) //определяем, является ли папкой, при этом не текущей и не вышестоящей

            {

                strcat(strcat(current\_path,"/"),de->d\_name); //записываем в конец текущего пути / и имя папки

                log += findMinotaur(current\_path, filename, Pt);

                current\_path[path\_len] = '\0';

            }

            if (de->d\_type == DT\_REG && !strcmp(de->d\_name, filename)) //проверяем, является ли файлом, и сравнием названия файлов

            {

                int path\_len = strlen(current\_path);

                strcat(strcat(current\_path,"/"),de->d\_name);//записываем в конец текущего пути / и имя файла

                FILE \*f = fopen(current\_path, "r");//открываем файл на чтение

                char \*str = (char \*)malloc(51);

                fgets(str, 50, f);

                if (strstr(str, "Minotaur")) //определяем, является ли содержимое файла минотавром

                {

                    strcpy((Pt->array)[Pt->count], current\_path); //копируем путь к нему

                    Pt->count++;

                    return 1;

                }

                else

                {

                    if (strstr(str, "Deadlock")) //если тупик

                    {

                        if (Pt->count) Pt->count--;

                        return 0;

                    }

                    else

                    {

                        while(strstr(str, "\n")>0)//пока не дойдем до пустой строки

                        {

                            strcpy((Pt->array)[Pt->count], current\_path);

                            Pt->count++;

                            char \*obj = strtok(str," \n");

                            obj = strtok(NULL, " \n"); //разбиваем содержимое строки

                            log += findMinotaur(".", obj, Pt); //ищем минотавра

                            if (!log) //если его не оказалось

                                {

                                    fgets(str, 50, f); //считываем следующую строку файла

                                }

                            else

                            return 1;

                        }

                    }

                }

            }

            de = readdir(dir);

        }

    closedir(dir);//закрываем директорию

    return log;

}

* Функция print – функция печати пути к минотавру

void print(Path );

void print(Path Pt)

{

    for(int i=0; i<Pt.count; i++)

    {

        printf("%s\n", (Pt.array)[i]); //выводим путь

    }

}

4. В main()

* Объявлена структура типа Path, выделяется память под двумерный массив указателей, заводим счетчик.

Path Pt;

Pt.array = (char \*\*)malloc(SIZE\*sizeof(char \*));

for (int i = 0; i < SIZE; ++i)

{

(Pt.array)[i] = (char\*)malloc(1000\*sizeof(char));

}

Pt.count = 0;

* Проверка на то, сколько аргументов подается в main, если 2 элемента, то функция поиска минотавра начинает стартует с указанной директории, и с заданного файла

Иначе поиск происходит в текущей папке

if(c==2)

    {

        findMinotaur(v[1], "file.txt", &Pt);

    }

else

{

findMinotaur(".","file.txt", &Pt);

}

5. Затем файл main.c с кодом написанной программы и отчёт о проведенной работе был загружен на github c помощью консоли:

* Создана новая ветка Filippenko\_sem2\_lr3

git checkout –b Filippenko\_sem2\_lr3

* Создана новая директория Filippenko\_sem2\_lr3 и файл lr3.c в ней, куда помещен код написанной программы

mkdir Filippenko\_sem2\_lr3

cd Filippenko\_sem2\_lr3

Созданные файлы добавлены для загрузки

git add Filippenko\_sem2\_lr3

* Добавлен комментарий

git commit –m “LR3 done”

* Файлы загружены на репозиторий Git

git push origin Filippenko\_sem2\_lr3

**Вывод:** научились рекурсивно обходить дерево папок, смогли создать алгоритм, позволяющий найти минотавра в файлах.