МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №3
по дисциплине «Программирование»
Тема: «Обход файловой системы»

Студентка гр. 2384 _____ Валеева А. А.

Преподаватель _____ Гаврилов А. В.

Санкт-Петербург
2023

Цель работы

Получить практические навыки и изучить принцип работы обхода файловой системы с помощью рекурсии.

Задание

Вариант 1

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.

Пример:

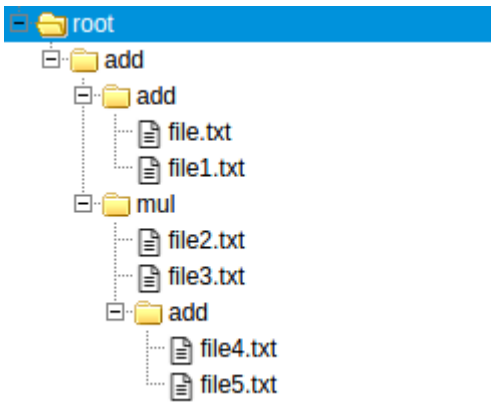
Содержимое файла a1.txt

```
@include a2.txt
@include b5.txt
@include a7.txt
```

А также файл может содержать тупик:

Содержимое файла a2.txt

```
Deadlock
```



Программа должна вывести правильную цепочку файлов (с путями), которая привела к поимке файла-минотавра.

Пример

file.txt:

```
@include file1.txt
@include file4.txt
@include file5.txt
```

file1.txt:

```
Deadlock
```

file2.txt:

```
@include file3.txt
```

file3.txt:

```
Minotaur
```

file4.txt:

```
@include file2.txt
@include file1.txt
```

file5.txt:

```
Deadlock
```

Правильный ответ:

```
./root/add/add/file.txt
./root/add/mul/add/file4.txt
./root/add/mul/file2.txt
./root/add/mul/file3.txt
```

Цепочка, приводящая к файлу-минотавру может быть только одна.

Общее количество файлов в каталоге не может быть больше 3000.

Циклических зависимостей быть не может.

Файлы не могут иметь одинаковые имена.

Ваше решение должно находиться в директории **/home/box**, файл с решением должен называться **solution.c**. Результат работы программы должен быть записан в файл **result.txt**. Ваша программа должна обрабатывать директорию, которая называется **labyrinth**.

Выполнение работы.

Для обработки директорий и файлов нужно подключить соответствующие библиотеки, такие как *dirent.h* и *string.h*. Будем считать, что максимальный по длине путь к файлу будет занимать меньше 5000 символов. Для хранения и обработки файлов реализована структура *file,* хранящая в себе название файла и его путь. Отдельно реализована функция *read_file* для чтения файла и записи результирующего ответа в файл *result.txt*. Программа сначала находит всевозможные файлы и папки начиная с предложенной директории *labyrinth*, используя рекурсию. Так создаётся массив структур, которые хранят в себе имя файла и путь к нему. Далее, начиная с *file.txt*, программа начинает читать содержимое текстовых файлов и зависимости от содержимого выполнять либо переход на другой файл, либо шаг назад в рекурсии, либо нахождение нужного файла. Все ходы записываются по порядку в массив указателей на структуры *file*.

Разработанный программный код см. в приложении А.

Результаты тестирования см. в приложении Б.

Выводы

В ходе лабораторной работы была освоена рекурсивная работа с директориями и фацлами, а также реализована программа по их обработке.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Valeyeva_Alina_lb3.c

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <dirent.h>

#define MXSZ 5000

typedef struct file{
    char name[50];
    char way[MXSZ];
} file;

void read_file(int size, file **arr){
    FILE *res = fopen("./result.txt", "w");
    for(int i = 0; i < size; i++){
        fprintf(res, "%s\n", arr[i]->way);
    }
    fclose(res);
}

void directory(char* path, file **clt, int *n){
    DIR *dir = opendir(path);
    char temp[MXSZ];
    strcpy(temp, path);
    strcat(temp, "/");
    if(!dir)
        return;
    struct dirent *elem;
    while((elem = readdir(dir))){
        if(!strcmp(elem->d_name, "..") || !strcmp(elem->d_name, "."))
            continue;
        int len = strlen(temp);
        strcat(temp, elem->d_name);
        if(elem->d_type == DT_REG){
            file *nwelem = (file *) malloc(sizeof(file));
            strcpy(nwelem->name, elem->d_name);
            strcpy(nwelem->way, temp);
            temp[len] = '\0';
            clt[(*n)++] = nwelem;

        }
        else if(elem->d_type == DT_DIR){
            directory(temp, clt, n);
            temp[len] = '\0';
        }
    }
}

    void mntr(file *elem, file **res, int idx, file **arr, int size, int
*flag){
```

```
        res[idx] = elem;
        FILE *broker = fopen(elem->way, "r");
        char txt[MXSZ];
        fgets(txt, MXSZ, broker);
        char *t;
        if(!strcmp(txt, "Minotaur")){
            *flag = 1;
            read_file(idx + 1, res);
        }
        while(!feof(broker)){
            t = strchr(txt, '\n');
            if(t){
                *(t) = '\0';
            }
            char *tok = strtok(txt, " ");
            tok = strtok(NULL, " ");
            for (int i = 0; i < size; i++) {
                if (!strcmp(arr[i]->name, tok)) {
                    mntr(arr[i], res, idx + 1, arr, size, flag);
                    break;
                }
            }
            if(*flag)
                break;
            fgets(txt, MXSZ, broker);
        }

        fclose(broker);
    }

    int main(){
        int n = 0, i, flag = 0;
        file **clt = (file **)malloc(sizeof(file *) * MXSZ), **res =
(file **)malloc(sizeof(file *) * MXSZ);
        directory("./labyrinth", clt, &n);
        for(i = 0; i < n; i++){
            if(!strcmp("file.txt", clt[i]->name)){
                mntr(clt[i], res, 0, clt, n, &flag);
            }
        }
        for(i = 0; i < n; i++){
            free(clt[i]);
        }
        free(clt);
    }
```

ПРИЛОЖЕНИЕ Б.
ТЕСТИРОВАНИЕ

Таблица Б.1 – результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	file.txt: @include file1.txt @include file4.txt @include file5.txt file1.txt: Deadlock file2.txt: @include file3.txt file3.txt: Minotaur file4.txt: @include file2.txt @include file1.txt file5.txt: Deadlock	./root/add/add/file.txt ./root/add/mul/add/file4.txt ./root/add/mul/file2.txt./root/add/mulfile3.txt	Верный ответ