**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: Рекурсия, циклы, обход дерева

|  |  |  |
| --- | --- | --- |
| Студентка гр. 6304 |  | Прозорова А.Д. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

**Цель работы**

Написать программу, которая выводит верный путь до файла, содержащего слово «Minotaur», с помощью рекурсивного обхода дерева директорий.

**Задание**

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. Подключение библиотек для работы и файлами и директориями

#include <stdio.h>

#include <sys/types.h>

#include <dirent.h>

#include <string.h>

#include <stdlib.h>

#define MAX\_WAYS 200

#define WAY\_SIZE 1000

#define MAX\_TOKENS 200

2. Функция, считывающая информацию из файла

char \*get\_file\_info(const char \*way\_buff, int \*file\_size){

FILE \*file=fopen(way\_buff,"rt"); //открываем файл для чтения

//находим размер файла

fseek(file,0, SEEK\_END);

\*file\_size = ftell(file);

fseek(file,0, SEEK\_SET);

//выделяем память по строку

char \*str=(char\*)malloc(sizeof(char)\*(\*file\_size));

//считываем информацию в строку

fread(str, 1, \*file\_size, file);

//закрываем файл

fclose(file);

return str;

}

3. Обход директории в поиске файла, содержащего слово Minotaur, возвращает длину пути до этого файла

int search\_minotaur(char \*new\_way, char \*start\_dir, char\* file\_name, char \*\*all\_ways)

{

char way\_buff[WAY\_SIZE]; //временный буфер

int length\_way=0; //Длина пути

strcpy(way\_buff,new\_way); //Запись пути для дальнейшего использования и изменения

DIR \*dir = opendir(way\_buff); //Открываем директорию

struct dirent \*file\_info = readdir(dir); //Читаем информацию о текущем файле в директории

while(file\_info) //пока в директории не закончились файлы

{

//получаем текущий путь

int path\_len = strlen(way\_buff);

strcat(way\_buff,"/");

strcat(way\_buff,file\_info->d\_name);

//ищем нужный файл

//если нужный файл это регулярный файл и его имя совпадает с искомым файлом

if(file\_info->d\_type==DT\_REG && !strcmp(file\_info->d\_name,file\_name))

{

int number\_of\_tokens = 0;

file\_size = 0;

//считываем строку из файла

char \*str=get\_file\_info(way\_buff,&file\_size);

//разбиваем строку на лексемы

char tokens\_arr[MAX\_TOKENS][file\_size]; //массив лексем

//разбиваем строку на лексемы

char \*item=strtok(str," \n"); //указатель на текущую лексему

while(item!=NULL)

{

strcpy(tokens\_arr[number\_of\_tokens++],item);

item=strtok(NULL," \n");

}

//анализируем лексемы

int i;

for(i=0;i<number\_of\_tokens;i++)

{

//если тупик, возвращаем длину пути

if(!strcmp(tokens\_arr[i],"Deadlock")) return length\_way;

//если файл-минотавр найден, запоминаем найденный путь

else if(!strcmp(tokens\_arr[i],"Minotaur"))

{

all\_ways[length\_way]=(char\*)malloc(sizeof(char)\*1000);

strcpy(all\_ways[length\_way],way\_buff);

length\_way=1;

return length\_way;

}

//если предыдущая лексема указывает на то, что данная лексема - название следующего файла, переходим к следующему файлу

else if(!strcmp(tokens\_arr[i-1],"@include"));

{

length\_way=search\_minotaur(start\_dir,start\_dir,tokens\_arr[i],all\_ways);

if(length\_way!=0)

{

all\_ways[length\_way]=(char\*)malloc(sizeof(char)\*1000);

strcpy(all\_ways[length\_way],way\_buff);

length\_way++;

return length\_way;

}

}

}

}

//если файл это директория (не родительская и не текущая), проверяем ее

if( file\_info->d\_type == DT\_DIR && strcmp(".",file\_info->d\_name)!=0 && strcmp("..",file\_info->d\_name)!=0)

{

length\_way = search\_minotaur(way\_buff,start\_dir,file\_name,all\_ways);

}

way\_buff[path\_len] = '\0'; //если слово не найдено, переходим к следующему файлу, очищаем последний найденный путь

file\_info = readdir(dir);

}

closedir(dir);

return length\_way;

}

4. Главная функция, которая вызывает функцию для поиска путей к файлу-минотавру и выводит их

int main()

{

char \*file\_name="file.txt"; //файл, с которого начинаем поиск

char \*\*all\_ways=(char\*\*)malloc(MAX\_WAYS\*sizeof(char\*)); //Массив путей

int length\_way=0;

//ищем пути к файлу-минотавру

length\_way=search\_minotaur(".",".",file\_name,all\_ways);

//Вывод путей и очищение памяти

int i;

for(i=length\_way-1;i>=0;i--){

printf("%s\n",all\_ways[i]);

free(all\_ways[i]);}

free(all\_ways);

return 0;

}

**Вывод**

В ходе данной работы было изучено применение языка программирования C (а именно рекурсии) для работы с файловой системой.