**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: Рекурсия, циклы, обход дерева

|  |  |  |
| --- | --- | --- |
| Студент гр. 6303 |  | Иванов Д.В. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

**Цель работы:**

**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 <sys/types.h>

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#include <dirent.h>

1. Для удобства работы использованы константы, отвечающие за длину строки перед именем подключаемого файла и максимальную длину строки.

#define INCLUDE\_STR\_LEN 8

#define MAX\_STR\_LEN 50

1. Объявлена структура для хранения необходимой информации о подключенном файле.

typedef struct Files

{

char name[255];

char \*way;

int sizeWay;

}Files;

1. Создана функция findFiles для рекурсивного обхода директорий, записи и поиска файлов.

int findFiles(Files \*FoundFiles, char \*way, int \*sizeOfWay, int \*countFiles, int \*sizeFilesArr)

{

int wayLen = strlen(way);

DIR \*dir;

if (wayLen > ((\*sizeOfWay) - 10))

{

(\*sizeOfWay) += 100; way = (char\*)realloc(way, sizeof(char)\*(\*sizeOfWay));

}

struct dirent \*entry;

dir = opendir(way); int i;

while (entry = readdir(dir)) {

for (i = 0; i<\*countFiles + 1; i++)

{

if (!strcmp(entry->d\_name, FoundFiles[i].name))

{

strcat(way, entry->d\_name);

FILE \* file = fopen(way, "r");

char bufStr[MAX\_STR\_LEN];

int sumLen = strlen(way) + strlen(FoundFiles[i].way);

if (sumLen > FoundFiles[i].sizeWay)

{

FoundFiles[i].sizeWay = sumLen + 100;

FoundFiles[i].way = (char \*)realloc(FoundFiles[i].way, sizeof(char)\*FoundFiles[i].sizeWay);

}

strcat(FoundFiles[i].way, way); strcat(FoundFiles[i].way, "\n");

while (fgets(bufStr, 101, file))

{

if (strstr(bufStr, "@include "))

{

(\*countFiles)++;

if (\*countFiles > \*sizeFilesArr - 2)

{

sizeFilesArr += 10; FoundFiles = (Files \*)realloc(FoundFiles, sizeof(Files)\*(\*sizeFilesArr));

}

if (strchr(bufStr, '\n'))

bufStr[strlen(bufStr) - 2] = '\0';

strcpy(FoundFiles[\*countFiles].name, &bufStr[INCLUDE\_STR\_LEN + 1]);

FoundFiles[\*countFiles].sizeWay = MAX\_STR\_LEN;

if (FoundFiles[i].sizeWay > FoundFiles[\*countFiles].sizeWay)

{

FoundFiles[\*countFiles].sizeWay = FoundFiles[i].sizeWay + 100;

FoundFiles[\*countFiles].way = (char \*)realloc(FoundFiles[\*countFiles].way, sizeof(char)\*FoundFiles[\*countFiles].sizeWay);

}

FoundFiles[\*countFiles].way = (char \*)calloc(6 \* FoundFiles[\*countFiles].sizeWay, sizeof(char));

strcat(FoundFiles[\*countFiles].way, FoundFiles[i].way); }

if (!strcmp(bufStr, "Minotaur"))

{

closedir(dir);

fclose(file);

return 1;

}

}

way[wayLen] = '\0';

fclose(file);

free(FoundFiles[i].way);

FoundFiles[i] = FoundFiles[\*countFiles];

(\*countFiles)--;

}

}

if (!strchr(entry->d\_name, '.'))

{

strcat(way, entry->d\_name);

strcat(way, "/");

if (findFiles(FoundFiles, way, sizeOfWay, countFiles, sizeFilesArr))

{

closedir(dir);

return 1;

}

way[wayLen] = '\0';

}

}

closedir(dir);

return 0;

}

1. В функции main() производится инициализация необходимых переменных, а также выделение памяти. Затем вызывается функция findFiles() для поиска начального файла “file.txt”. После окончания работы функции, программа выводит путь в файлу-Минотавру. Перед закрытием освобождается выделенная память.

int main()

{

int sizeOfWay = 50;

int countFiles = 0;

int sizeFilesArr = 10;

Files \*FoundFiles = (Files \*)malloc(sizeof(Files)\*sizeFilesArr);

strcpy(FoundFiles[countFiles].name, "file.txt");

FoundFiles[countFiles].sizeWay = MAX\_STR\_LEN;

FoundFiles[countFiles].way = (char \*)calloc(MAX\_STR\_LEN, sizeof(char));

char \*way = (char \*)malloc(MAX\_STR\_LEN);

strcat(way, "./");

DIR \*dir;

dir = opendir(way);

if (!dir)

{

printf("Dir on this way %s doesn't root\n", way);

return 0;

}

struct dirent \*entry;

do {

if (findFiles(FoundFiles, way, &sizeOfWay, &countFiles, &sizeFilesArr))

{

printf("Way to minotaur: \n%s", FoundFiles[countFiles - 1].way);

break;

}

} while (countFiles>0);

closedir(dir);

int i;

for (i = 0; i < countFiles; i++)

free(FoundFiles[i].way);

free(FoundFiles);

free(way);

return 0;

}

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

* Создана новая ветка Ivanov\_lab3\_sem2

git checkout –b Ivanov\_lab3\_sem2

* Создана новая директория Ivanov\_lab3\_sem2 и файл main.c в ней, куда помещен код написанной программы

mkdir Ivanov\_lab3\_sem2

cd Ivanov\_lab3\_sem2

nano main.c

* Созданные файлы добавлены для загрузки

git add Ivanov\_lab3\_sem2

* Добавлен комментарий

git commit –m “2 semestr. Lab 3”

* Файлы загружены на репозиторий Git

git push origin Ivanov\_lab3\_sem2

**Вывод:** в ходе работы создана функция для рекурсивного обхода директорий и поиска файлов, в работе использованы циклы, рекурсия, а также динамическая память.