**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №3**

**по дисциплине «Программирование»**

Тема: **Динамические структуры данных.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Некрасов Н.А. |
| Преподаватель |  | Кринкин К.В. |

Санкт-Петербург

2017

Цель:

Написать программу, проверяющую валидность html-странички.

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

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

*! Общее количество файлов в каталоге не может быть больше 200.*

*! Циклических зависимостей быть не может.*

*! Файлы не могут иметь одинаковые имена.*

Ход работы:

* Предопределение максимальной длины пути и количества включаемых файлов

#**define** FNUM 200

#**define** PLEN 10000

* Подключение библиотек для работы с директориями, строками, вводом и выводом, а также основными функциями.

#**include** <stdio.h>

#**include** <stdlib.h>

#**include** <dirent.h>

#**include** <string.h>

* Описание прототипов функций для считывания, нахождения и вывода пути к файлу

void fileFinder(char [][PLEN], char \*, char\*);

int read(char\*, char\*, char [][PLEN]);

void filePath(char \*, char [][PLEN], int\*);

* Основная функция, в которой происходит вызов функций по поиску и выводу файлов.

int main(void){

int fnum=0;

char files[200][PLEN];

filePath(".", files, &files\_num);

fileFinder(files, "file**.txt**", "");

return 0;

}

* Описание функции для сопоставления имени файла с путём к нему и выводу его на экран.

void fileFinder(char files[][PLEN], char \*currFile, char\* empty){

char buffer[PLEN];

char \*ptext;

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

{

if(**strstr**(files[i], currFile)!=NULL)

{

strcpy(buffer, empty);

strcat(buffer, files[i]);

strcat(buffer, "\n");

if(**read**(files[i], buffer, files))

{

ptext=**strtok**(buffer, "\n");

while(ptext!=NULL)

{

printf("%s\n",(ptext+2));

ptext=**strtok**(NULL, "\n");

}

}

}

}

}

* Функция обходит все вложенные каталоги, находит пути к файлам и записывает их в массив. Приии нннахождении вложенного каталога функция удлиняет путь, при нахождении файла функция записывает текущий путь в массив.

void filePath(char \*startdir, char files[][PLEN], int\* files\_num){

char ActivePath[PLEN]; //Массив под путь

strcpy(ActivePath, startdir);

DIR \*dir;

struct dirent \*cat = readdir(dir);

if(dir = opendir(ActivePath));

while(cat = readdir(dir))//Пока есть что читать

{

if(cat->d\_type==DT\_REG)

{

int path\_len = strlen(ActivePath);

strcat(ActivePath,"/");

strcat(ActivePath,cat->d\_name);

strcpy(files[(\*files\_num)++], ActivePath);

ActivePath[path\_len] = '\0'; //Добавляем в конец \0, чтобы strcat работал

}

if(cat->d\_type == DT\_DIR && strcmp(".",cat->d\_name) && strcmp("..", cat->d\_name))

{

int path\_len = strlen(ActivePath); //Изменяем длину пути на текущий

strcat(ActivePath,"/");

strcat(ActivePath,cat->d\_name);

filePath(ActivePath, files, files\_num);

ActivePath[path\_len] = '\0';

}

}

closedir(dir);

}

* Функция для считывания содержимого файла. При нахождениие включения нового файла она записывает его имя в массив, при нахождении «Минотавра» возвращает 1 и инициирует вывод цепочки подключений

int read(char\* filename, char\* empty, char files[][PLEN]){

FILE \*file;

if ((file = fopen (filename, "r")) != NULL)//Открываем файл на чтение и проверяем на правильность

{

char str [4000]; //Буфер для содержимого

while (**fgets**(str, sizeof(str), file)!= NULL) //Считываем файл

{

if (!**strcmp**(str, "Minotaur"))

return 1;

char \*ptext;//Объвляем массив для внутренних инклюдов

ptext=**strtok**(str, " \n\r");

while(ptext!=NULL)

{

if(**strstr**(ptext, "**.txt**"))

fileFinder(files, ptext, empty); //Начинаем поиск файлов

ptext=**strtok**(NULL, " \n\r");

}

}

fclose(file);

}

else

{

perror(filename);

}

return 0;

}

Вывод:

В данной лабораторной работе была реализована программа, обходящая дерево каталогов и нахождения файла содержащего строку «Minotaur».