**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 максимального количества файлов и максимальной длины пути

#**define** MAXPATH 200

#**define** MAXLENGTH 1000

* Подключение библиотек для работы с директориями, ввода/вывода, работы со строками, а также основных функций

#**include** <stdio.h>

#**include** <dirent.h>

#**include** <string.h>

#**include** <stdlib.h>

* Создание прототипов функций, отвечающих за чтение информации из файла, а также поиск пути к файлу.

char\* read(char \*cpath, int \*fsize);

int filefinder(char \*npath, char \*startdir, char\* file, char \*\*paths);

* Основное тело программы в котором происходит выделение памяти под коннечный массив путей, вызов функции, их находящей, а также вывод и очистка памяти

int main()

{

char \*\*paths=(char\*\*)**malloc**(MAXPATH\***sizeof**(char\*));

int depth=0;

depth=**filefinder**(".", ".", "file**.txt**", paths);

for(int i=depth-1;i>=0;i--)

{

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

free(paths[i]);

}

free(paths);

return 0;

}

* Функция для считывания информации из файла, а также определения его размера путём, вычисления разницы между начальным и конечным положением указателя.

char \*read(char \*cpath, int \*fsize)

{

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

fseek(file,0, SEEK\_END);

\*fsize = ftell(file);

rewind(file);

char \*str=(char\*)**malloc**(**sizeof**(char)\*(\*fsize));

fread(str, sizeof(char), \*fsize, file);

fclose(file);

return str;

}

* Функция поиска файлов.

int filefinder(char \*npath, char \*startdir, char\* file, char \*\*paths)

{

char cpath[MAXLENGTH];

int depth = 0;

strcpy (cpath, npath);

* Происходит объявление указателя на файловый поток и его открытие

DIR \*dir = opendir(cpath);

struct dirent \*cat = readdir(dir);

* До тех пор, пока есть файлы внутри каталога происходит их считывание

while(cat)

{

int path\_len = strlen(cpath);

strcat(cpath, "/");

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

* Изначально файл проверяется на совпадение его имени с искомым, а также то, что он является txt-файлом. При выполнении этих условий происходит считывание информации из файла.

if(cat->d\_type == 8 && !**strcmp**(cat->d\_name, file) && strstr(cat->d\_name,"**.txt**"))

{

int j = 0;

int fsize = 0;

char \*str=**read**(cpath,&fsize);

* При нахождениее «Минотавра» в файле происходит запись пути этого файла в массив и возвращение глубины смещения.

if(**strstr**(str, "Minotaur"))

{

paths[depth]=(char\*)**malloc**(**sizeof**(char)\*1000);

strcpy(paths[depth],cpath);

depth=1;

return depth;

}

* При нахождении «Дедлока» происходит возвращение глубины смещения – нуля.

if(**strstr**(str,"Deadlock")) return depth;

* В противном случае текст разбивается на лексемы и по лексемам, включающим в себя строку «.txt» происходит поиск новых файлов.

char tokens[MAXPATH][fsize];

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

while(item!=NULL)

{

strcpy(tokens[j++],item);

if(**strstr**(tokens[j-1], "**.txt**"))

{

depth=**filefinder**(startdir, startdir, tokens[j-1], paths);

* Проверяется значение глубины, при значении отличном от нуля происходит запись пути в массив, иначе путь обнуляется.

if (depth)

{

paths[depth]=(char\*)**malloc**(**sizeof**(char)\*1000);

strcpy(paths[depth++],cpath);

return depth;

}

}

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

}

}

* Проверяется наличие директорий отличных от родительской и домашней, при наличии вызывается поиск файла внутри них.

if (cat->d\_type == 4 && strcmp(".",cat->d\_name) && strcmp("..",cat->d\_name))

{

depth = filefinder(cpath, startdir, file, paths);

}

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

cat = readdir(dir);

}

closedir(dir);

return depth;

}

Вывод:

В данной лабораторной работе была реализована программа поиска файла, содержащего строку «Minotaur», путём обхода дерева каталогов.