**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: Рекурсивный обход директорий

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Смотрова А.А. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

Содержание

[**Цель работы** 3](#_Toc483876933)

[**Задача** 3](#_Toc483876934)

[**Ход выполнения задачи** 3](#_Toc483876935)

[**Вывод** 5](#_Toc483876936)

[**Исходный код программы** 5](#_Toc483876937)

# **Цель работы**

Написать программу, которая находит и выводит последовательность полных путей файлов, имена которых образуют заданную пользователем строку

# **Задача**

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. В качестве имени файла используется символ латинского алфавита.

На вход программе подается строка. Требуется найти и вывести последовательность полных путей файлов, имена которых образуют эту строку.

# **Ход выполнения задачи**

Для решения данной задачи была реализована рекурсивная функция check\_Dr, которая принимает на вход указатель на имя директории, указатель на путь директории и ее длину. Код функции представлен ниже.

int check\_Dr (char \*Name\_Dir, character \*Ch\_Path, int len) *//проверка директории*

{

char Current\_Dir[100];

strcpy(Current\_Dir, Name\_Dir);*//скопировали имя директории в новый массив*

DIR \*dir = opendir(Current\_Dir);

struct dirent \*content = readdir(dir);*//создаём структуру типа dirent*

if(dir)

while(content)

{

if (content->d\_type == 8)

{*//если это файл*

char \*fileName = strtok(content->d\_name, ".");*//имя файла до точки*

check\_Ch(Ch\_Path, Current\_Dir, fileName, len);*//проверяем первую букву*

if (is\_name\_full(Ch\_Path, len))*//если имя полное,  выходим*

return 0;

}

if ((content->d\_type == 4) && strcmp(".", content->d\_name) && strcmp("..", content->d\_name))*//если это директория*

{

int Current\_Dir\_len = strlen(Current\_Dir);*//длина этой директории*

strcat(Current\_Dir, "/");*//добавил файловый разделитель*

strcat(Current\_Dir, content->d\_name);*//добавили имя файла*

check\_Dr(Current\_Dir, Ch\_Path, len);*//проверки директории*

\*(Current\_Dir + Current\_Dir\_len) = '**\0**';*//ставим конец строки*

}

content = readdir(dir);

}

closedir(dir);

return 0;

}

Вдобавок к этой функции были реализованы несколько дополнительных функций, важных для выполнения поставленной задачи.

Функция Check\_ch проверяет, начинается ли имя файла в данной директории с заданной в строке буквы.

int check\_Dr (char \*Name\_Dir, character \*Ch\_Path, int len) *//проверка директории*

{

char Current\_Dir[100];

strcpy(Current\_Dir, Name\_Dir);*//скопировали имя директории в новый массив*

DIR \*dir = opendir(Current\_Dir);

struct dirent \*content = readdir(dir);*//создаём структуру типа dirent*

if(dir)

while(content)

{

if (content->d\_type == 8)

{*//если это файл*

char \*fileName = strtok(content->d\_name, ".");*//имя файла до точки*

check\_Ch(Ch\_Path, Current\_Dir, fileName, len);*//проверяем первую букву*

if (is\_name\_full(Ch\_Path, len))*//если имя полное,  выходим*

return 0;

}

if ((content->d\_type == 4) && strcmp(".", content->d\_name) && strcmp("..", content->d\_name))*//если это директория*

{

int Current\_Dir\_len = strlen(Current\_Dir);*//длина этой директории*

strcat(Current\_Dir, "/");*//добавил файловый разделитель*

strcat(Current\_Dir, content->d\_name);*//добавили имя файла*

check\_Dr(Current\_Dir, Ch\_Path, len);*//проверки директории*

\*(Current\_Dir + Current\_Dir\_len) = '**\0**';*//ставим конец строки*

}

content = readdir(dir);

}

closedir(dir);

return 0;

}

Функция is\_name\_full проверяет, готово ли «слово» из букв заданной строки

int is\_name\_full (character \*Ch\_Path, int len) *//функция для проверки готовности слова*

{

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

if (!(Ch\_Path + i)->trigger)

return 0;

return 1;

}

Для реализации данной программы была введена специальная структура character:

typedef struct character *//структура для хранения:*

{

char symbol; *//буквы*

char path[PATHLEN]; *//директории*

int trigger; *//триггерa есть-нет*

} character;

# **Вывод**

В ходе данной лабораторной работы была создана программа, на основе алгоритма обхода директорий с помощью рекурсивной функции.

# **Исходный код программы**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/types.h> //различные типы данных

#include <dirent.h> //Открытие и вывод каталогов

#define PATHLEN 1000

typedef struct character *//структура для хранения:*

{

char symbol; *//буквы*

char path[PATHLEN]; *//директории*

int trigger; *//триггерa есть-нет*

} character;

int is\_name\_full (character \*Ch\_Path, int len) *//функция для проверки готовности слова*

{

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

if (!(Ch\_Path + i)->trigger)

return 0;

return 1;

}

void check\_Ch(character \*Ch\_Path, char \*Current\_Dir, char \*fileName, int len) *//функция ищет букву в имени файла*

{

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

{

if (((Ch\_Path + i)->symbol == \*fileName) && !((Ch\_Path + i)->trigger))

{

\*(fileName + strlen(fileName)) = '.';

int Current\_Dir\_len = strlen(Current\_Dir);

strcat(Current\_Dir, "/");

strcpy((Ch\_Path + i)->path, strcat(Current\_Dir, fileName));

(Ch\_Path + i)->trigger = 1;

\*(Current\_Dir + Current\_Dir\_len) = '**\0**';

**break**;

}

}

}

int check\_Dr (char \*Name\_Dir, character \*Ch\_Path, int len) *//проверка директории*

{

char Current\_Dir[100];

strcpy(Current\_Dir, Name\_Dir);*//скопировали имя директории в новый массив*

DIR \*dir = opendir(Current\_Dir);

struct dirent \*content = readdir(dir);*//создаём структуру типа dirent*

if(dir)

while(content)

{

if (content->d\_type == 8)

{*//если это файл*

char \*fileName = strtok(content->d\_name, ".");*//имя файла до точки*

check\_Ch(Ch\_Path, Current\_Dir, fileName, len);*//проверяем первую букву*

if (is\_name\_full(Ch\_Path, len))*//если имя полное,  выходим*

return 0;

}

if ((content->d\_type == 4) && strcmp(".", content->d\_name) && strcmp("..", content->d\_name))*//если это директория*

{

int Current\_Dir\_len = strlen(Current\_Dir);*//длина этой директории*

strcat(Current\_Dir, "/");*//добавил файловый разделитель*

strcat(Current\_Dir, content->d\_name);*//добавили имя файла*

check\_Dr(Current\_Dir, Ch\_Path, len);*//проверки директории*

\*(Current\_Dir + Current\_Dir\_len) = '**\0**';*//ставим конец строки*

}

content = readdir(dir);

}

closedir(dir);

return 0;

}

int main ()

{

char Name\_Dir[100] = {'.', '**\0**'};

char s[100];

fgets(s, (102), stdin); *//читаем строку, в которой записана последовательность букв*

int len;

len= (strlen(s) - 1);

character \*Ch\_Path = (character\*)malloc(len \* sizeof(character));

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

{

(Ch\_Path + i)->symbol = s[i];

(Ch\_Path + i)->trigger = 0;

}

check\_Dr(Name\_Dir, Ch\_Path, len);

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

printf("%s**\n**", (Ch\_Path + i)->path);

return 0;

}