

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: Обход файловой системы

Студент гр. 1304

Заика Т.П

Преподаватель

Чайка К.В.

Санкт-Петербург

2022

Цель работы.

Рекурсионный обход дерева файловой системы для решения поставленной задачи (в зависимости от варианта).

Задание.

Вариант №4.

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 .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.

! Регистрозависимость

! Могут встречаться файлы, в имени которых есть несколько букв и эти файлы использовать нельзя.

! Одна буква может встречаться один раз.

Ваше решение должно находиться в директории /home/box, файл с решением должен называться solution.c. Результат работы программы должен быть записан в файл result.txt. Ваша программа должна обрабатывать директорию, которая называется tmp.

Основные теоретические положения.

Рекурсия

Работа с деревом файловой системы

Выполнение работы.

В ходе работы для решения поставленной задачи было принято создать программу, осуществляющую рекурсивный обход дерева файловой системы. Для начала нужно было создать файл на запись, после осуществить ввод с клавиатуры искомого слова, далее необходимо задать директорию, в которой будет осуществляться обход, а также необходимо задать изначальный шаблон для поиска файлов с заданным наименованием, в который будет подставляться каждая буква из введенного слова. Данное сформированное название файла будет передаваться рекурсивной функции совместно с указанием пути исходной директории. Внутри рекурсии происходит обход всего дерева, включая внутренние папки и их содержимое. При нахождении файла с заданным наименованием вызывается функция, осуществляющая запись в созданный ранее текстовый файл полного пути до найденного файла с заданным названием. В конечном итоге, результатом выполнения программы является текстовый файл со всеми путями до файлов заданного наименования, которые побуквенно составляют введенное пользователем слово.

Переменные:

`FILE *f_out` — указатель на объект, содержащий информацию для управления потоком. Используется для создания на запись и дополнения файла, указанного в функции `fopen()`.

`char input_line[100]` — переменная для хранения введенного пользователем с клавиатуры значения.

`char path[300]` — переменная для хранения исходного пути, по которому будет осуществляться рекурсивный обход файлового дерева.

`char cur_file[6]` — переменная для хранения шаблона названия файла для поиска по файловой системе.

`DIR *dir` - указатель на тип, представляющий поток каталога. Используется для открытия текущей директории.

struct dirent *cur — указатель на структуру, содержащую информацию о текущей директории. Используется для чтения открытой директории и получении информации о ней.

int slen — переменная для хранения длины текущего пути до директории.

char* curfilename — переменная для хранения текущего пути до файла.

Функции:

void dirtravel(char* dirname, char* filename) — функция, осуществляющая рекурсивный обход файлового дерева. Принимает путь исходной директории и название искомого файла.


void writeWayToFile(char* way, char* outfilename) — функция, осуществляющая дозапись значений в файл. Принимает путь до найденного при помощи рекурсивного обхода файла и название файла для дозаписи в него значений.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	 <pre> hello_world_test ~/Рабочий стол/hello_world_test ├── asdfgh │ ├── mkoipu │ │ ├── H.txt │ │ └── HH.txt │ ├── L.txt │ ├── IO.txt │ └── O.txt ├── qwerty │ ├── qwert │ │ ├── L.txt │ │ ├── e.txt │ │ ├── P.txt │ │ └── n.txt </pre>	hello_world_test/asdfgh/ mkoipu/H.txt hello_world_test/qwerty/ e.txt hello_world_test/qwerty/ qwert/L.txt hello_world_test/asdfgh/ l.txt hello_world_test/asdfgh/ O.txt	Успешный тест

Выводы.

Была исследована, изучена рекурсия и ее применение для обхода дерева файловой системы.

Разработана программа, выполняющая рекурсивный обход файловой системы для поиска файлов с названием, удовлетворяющем данному условию, и записывающая полный пути до таких файлов в выходной текстовый файл.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
#include <string.h>

void writeWayToFile(char* way, char* outfilename){
    FILE *f_out = fopen(outfilename, "a");
    fprintf(f_out, "%s\n", way);
    fclose(f_out);
}

void dirtravel(char* dirname, char* filename){
    DIR *dir = opendir(dirname); // dir != NULL
    if(!dir) return;
    struct dirent *cur;
    while(cur = readdir(dir)){
        if(cur->d_type == DT_DIR){
            if(strcmp(cur->d_name, ".") && strcmp(cur->d_name,
"..")){
                int slen = strlen(dirname);
                strcat(dirname, "/");
                strcat(dirname, cur->d_name);
                dirtravel(dirname, filename);
                dirname[slen] = '\0';
            }
        }
        if(cur->d_type == DT_REG){
            if(strcmp(cur->d_name, filename) == 0){
                char* curfilename = dirname;
                strcat(curfilename, "/");
                strcat(curfilename, cur->d_name);
                writeWayToFile(curfilename, "result.txt");
                break;
            }
        }
    }
    closedir(dir);
}

int main(){
    FILE *f_out = fopen("result.txt", "w");
    fclose(f_out);

    char input_line[100];
    scanf("%s", input_line);

    char path[300] = "./tmp";

    char cur_file[6] = "_.txt";
```

```
    for(int i=0; i<strlen(input_line); i++){
        cur_file[0] = input_line[i];
        dirtravel(path, cur_file);
    }

    return 0;
}
```