

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Программирование»**  
**ТЕМА: Обход файловой системы**

Студент гр. 1304

Поршнеv Р.А.

Преподаватель

Чайка К.В.

Санкт-Петербург

2022

### **Цель работы.**

Исследование рекурсивного обхода файловой системы.

### **Задание.**

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.

Вариант 4.

### **Основные теоретические положения.**

В данной работе использовались такие библиотеки, как `<string.h>`, `<dirent.h>`, `<stdlib.h>`, `<stdio.h>`. Использование библиотек `<stdlib.h>` и `<stdio.h>` достаточно очевидно, в то время как использование библиотеки `<string.h>` оправдано записью пути к файлам. Библиотека `<dirent.h>` нужна для работы с файлами и директориями.

### **Выполнение работы.**

В функции `main()` происходит открытие файла `result.txt`, в который будет записываться путь к файлу. Объявляется переменная `string` типа указатель на `char`, в которую будет записана исходная строка. Так же инициализируется

переменная *head\_path* типа указатель на *char*, в которую будет записываться путь к файлам.

В функции *readstring()* происходит посимвольное считывание исходной строки. Функции возвращается считанная строка типа указатель на *char*.

В функции *main()* запускается цикл, в котором каждая буква отправляется в функцию *list\_dir*, которая принимает начало пути, саму букву и указатель на файл для записи.

В функции *list\_dir* происходит открытие директории. Если открытие произошло успешно, то происходит считывание директории. Далее происходит перебор всех файлов и директорий. Если название директории не равно “.” и “..”, то инициализируется переменная *s*, в которую будет записываться текущий путь. Затем открывается файл или директория, хранящиеся в пути *s*. Если значение функции *opendir* равно NULL, то это файл. В таком случае следует проверка, является ли данный файл тем файлом, которую нужно отыскать. Если он таковым является, то он записывается в файл *result.txt* и происходит выход из функции. Если значение функции *opendir* не равно NULL, то добавляется слеш и происходит углубление в файловую систему. После обхода файлов и директорий в текущей директории происходит её закрытие.

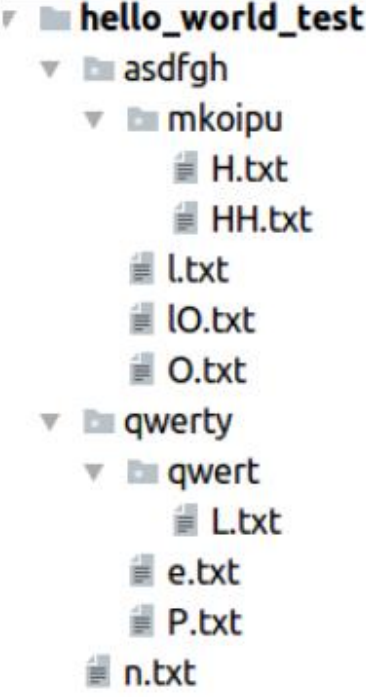
В функции *main()* после нахождения всех путей производится очистка памяти под строку *string* и закрывается файл *result.txt*.

### **Тестирование.**

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
-------	----------------	-----------------	-------------

1.		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	<p>Ответ правильный</p>
----	---	--	-------------------------

### Выводы.

Была изучена рекурсия и применена для обхода файловой системы.

Разработана программа, которая ищет файлы с определённым названием и записывает в текстовый файл их полные пути.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

```
#include <string.h>
#include <dirent.h>
#include <stdlib.h>
#include <stdio.h>
void list_dir(const char *dirPath, char c, FILE* result){

    char *name;
    DIR *dir = opendir(dirPath);
    if(dir){
        struct dirent *de = readdir(dir);
        while (de){
            if (strcmp(de->d_name, ".") && strcmp(de->d_name,
"..")){
                char s[255];
                s[0] = '\0';
                strcat(s, dirPath);
                strcat(s, de->d_name);
                if (opendir(s) == NULL){
                    name = de->d_name;
                    if ((name[0] == c) && (name[1] == '.') &&
(name[2] == 't') && (name[3] == 'x') && (name[4] == 't') && (strlen(name)
== 5)){
                        fprintf(result, "%s\n", s);
                        return;
                    }
                }else{
                    strcat(s, "/");
                    list_dir(s, c, result);
                }
            }
            de = readdir(dir);
        }
        closedir(dir);
    }
    char* readstring(){
        char c;
        char* string = NULL;
        string = malloc(sizeof(char));
        int symbols = 0;
        do{
            scanf("%c", &c);
            string = realloc(string, (symbols+1) * sizeof(char));
            string[symbols] = c;
            symbols += 1;
        }while (c != '\n');
        string[symbols-1] = '\0';
        return string;
    }
    int main(){
        FILE* result;
        result = fopen("result.txt", "w");
```

```
int i;
char *string = NULL;
char* head_path = "./tmp/";
string = readstring();
for(i = 0; i<strlen(string); i++){
    list_dir(head_path, string[i], result);
}
free(string);
fclose(result);
}
```