

**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. 3344

Ханнанов А.Ф.

Преподаватель

Глазунов С.А.

Санкт-Петербург

2024

## **Цель работы**

Ознакомиться с методами работы с файлами на языке Си. Научиться создавать функцию для рекурсивного обхода файлового дерева.

**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 *<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.

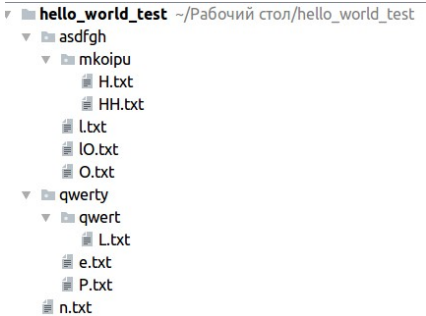
## **Выполнение работы**

В функции `main` происходит открытие/создание файла `result.txt`, в который будет записываться результат работы программы. После этого на вход принимается входная строка. По символам этой строки начинается обход с помощью цикла. Для каждого символа вызывается функция `listFilesRecursively`. Эта функция принимает на вход файл, путь и символ. Она записывает в переменную `path` обновлённый путь, если следом идущий элемент – директория, иначе проверяет название файла и в случае, если он удовлетворяет условию, записывает в файл. С обновлённым путём она снова запускается. В конце работы происходит закрытие директории.

## Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	 <p>hello_world_test ~/Рабочий стол/hello_world_test</p> <ul style="list-style-type: none"><li>asdfgh<ul style="list-style-type: none"><li>mkoipu<ul style="list-style-type: none"><li>H.txt</li><li>HH.txt</li><li>L.txt</li><li>IO.txt</li><li>O.txt</li></ul></li><li>qwerty<ul style="list-style-type: none"><li>qwert<ul style="list-style-type: none"><li>L.txt</li><li>e.txt</li><li>P.txt</li><li>n.txt</li></ul></li></ul></li></ul></li></ul>	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	Выходные данные находятся в файле result.txt

## **Выводы**

Были получены навыки работы с файловой системой на языке Си с помощью библиотеки `dirent.h`. Изучены способы обхода файлового дерева.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Khannanov\_Artem\_lb3.c

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

void listFilesRecursively(FILE *inp, char *base_path, char letter) {

    DIR *dir = opendir(base_path);

    if (!dir) return;

    struct dirent *dp = readdir(dir);
    char *path;

    while (dp != NULL) {

        if (dp->d_type == DT_DIR && strcmp(dp->d_name, ".") != 0 &&
            strcmp(dp->d_name, "..") != 0) {

            path = (char *)malloc(sizeof(char) * (strlen(base_path) +
            strlen(dp->d_name) + 2));
            path[0] = '\0';

            strcat(path, base_path);
            strcat(path, "/");
            strcat(path, dp->d_name);
            listFilesRecursively(inp, path, letter);

            free(path);

        } else {

            if (dp->d_type == DT_REG && strlen(dp->d_name) == 5 && dp->
            d_name[0] == letter && dp->d_name[1] == '.' &&
            dp->d_name[2] == 't' && dp->d_name[3] == 'x' && dp->d_name[4]
            == 't') {

                fprintf(inp, "%s/%s\n", base_path, dp->d_name);

            }

        }

        dp = readdir(dir);

    }

    closedir(dir);
}

int main() {
```

```
char *base_path = "./tmp";
FILE *file = fopen("./result.txt", "w");

char *inp_string = (char *)malloc(sizeof(char) * 1000);
scanf("%s", inp_string);

    for (size_t i = 0; i < strlen(inp_string); i++)
listFilesRecursively(file, base_path, inp_string[i]);

fclose(file);
free(inp_string);

return 0;
}
```