

**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. 0382

Сергеев Д.А.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2021

### **Цель работы.**

Изучение основных принципов работы с файловыми директориями операционной системы Linux на языке программирования Си. Реализация рекурсивного обхода файлового дерева.

### **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 *<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.**

Работа программы начинается с того, что считывается входная строка, после этого создаётся экземпляр структуры `searchLink` `ex` (в структуре `searchLink` 3 поля, массив строк `letter`, каждая строка – буква латинского алфавита и расширение `“.txt”`, массив строк `path` – в каждой строке записывается путь до файла с именем, соответствующим строке `letter` с этим же индексом, целое число `n` – количество элементов в массиве). Далее заполняется поле `letter` у `ex`. Затем вызывается функция `listDir`, в которую передаётся имя папки, по которой будет совершаться обход и адрес структуры `ex`.

В функции `listDir` создаётся строка `next`, которая содержит нынешний путь, далее переменной `DIR* dir` присваивается дескриптор директории, название которой передано в функцию. Далее происходит непосредственно обход директории в ходе которого происходит сравнение имени объекта обхода со строками из поля `letter` структуры `ex`, в случае совпадения в соответствующую строку `ex->path[i]` записывается путь до файла. Затем если объект обхода является папкой и не ведет в вышестоящую папку, то в переменную `len`

записывается текущая длина пути next, далее к строке next добавляется строка de->d\_name и вызывается функция listDir с аргументами next и ex, после работы функции выполняется функция closedir(dir).

Далее в функции main нашей программы вызывается функция createOutput() с аргументами “result.txt” и адрес структуры ex. Функция createOutput() в случае отсутствия файла с переданным именем создаёт данный файл с разрешением на редактирование и далее записывает в файл строки, содержащиеся в поле структуры ex->path. После этого заканчивается работа с этим файлом с помощью функции fclose(f).

Разработанный программный код см. в приложении А.

### Тестирование.

Для тестирования была создана директория hello\_world\_test(рис. 1)

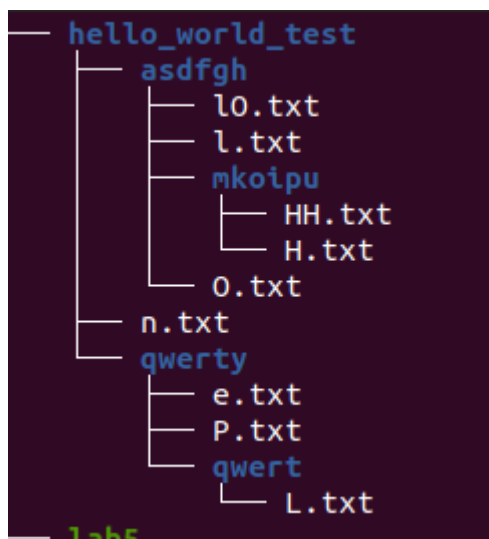


Рисунок 1 – Вид директории

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	HelLO	hello_world_test/asdfgh/mkoipu/H.txt hello_world_test/qwerty/e.txt hello_world_test/asdfgh/l.txt hello_world_test/qwerty/qwert/L.txt hello_world_test/asdfgh/O.txt	Программа работает правильно

## **Выводы.**

В ходе работы были изучены основные принципы работы с файловыми директориями операционной системы Linux на языке Си. Реализован поиск определённых файлов в директории при помощи рекурсивного обхода каталога. Была создана программа, которая с помощью функции `listDir()` совершает обход директории, а с помощью функции `createOutput()` происходит запись данных в файл.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab3.c

```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <stdlib.h>
#include <dirent.h>

typedef struct searchLink
{
    char** letter;
    char** path;
    int n;
} searchLink;

void listDir(const char* path, searchLink* ex)
{
    char next[200]={0};
    strcpy(next,path);
    strcat(next,"/");
    DIR* dir = opendir(path);
    if (!dir)
        return;
    struct dirent* de=readdir(dir);
    while (de)
    {
        for (int i=0;i<ex->n;i++)
        {
            if (strcmp(de->d_name,ex->letter[i])==0)
            {
                ex->path[i]=calloc(100,sizeof(char));
                strcpy(ex->path[i],next);
                strcat(ex->path[i],ex->letter[i]);
                break;
            }
        }
        if (de->d_type == DT_DIR && strcmp(de->d_name, ".")!=0 &&
strcmp(de->d_name, "..")!=0)
        {
            int len=strlen(next);
            strcat(next,de->d_name);
            listDir(next,ex);
            next[len]='\0';
        }
        de=readdir(dir);
    }
    closedir(dir);
}

void createOutput(char* filename,searchLink* ex)
{
    FILE* f= fopen(filename,"w");
    for (int i=0;i<ex->n;i++)
    {
```

```

        fprintf(f,"%s\n",ex->path[i]);
    }
    fclose(f);
}

int main()
{
    char* str=calloc(100,sizeof(char));
    fgets(str,100,stdin);
    searchLink ex;
    ex.n= strlen(str)-1;
    ex.letter= calloc(strlen(str),sizeof(char*));
    ex.path= calloc(strlen(str),sizeof(char*));
    for (int i=0;i<strlen(str)-1;i++)
    {
        ex.letter[i]=calloc(100,sizeof(char));
        ex.letter[i][0]=str[i];
        strcat(ex.letter[i],".txt\0");
    }
    listDir("./tmp",&ex);
    createOutput("result.txt",&ex);
    return 0;
}

```