

**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 ситемы**

Студент гр. 1304

\_\_\_\_\_

Басыров В.А.

Преподаватель

\_\_\_\_\_

Чайка К.В

Санкт-Петербург

2022

### **Цель работы.**

Изучение рекурсивного подхода к решению некоторых задач, а также изучение системы иерархии файлов и применение рекурсивных алгоритмов для решения задач обхода файловой системы.

### **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.

### **Основные теоретические положения.**

Рекурсия-подход к решению задач, в котором каждая подзадача выполняется путем самовывоза функции.

struct dirent-структура, содержащая информацию о папке.

opendir(Dir d)- открытие директории.

closedir(Dir d)-завершение работы с директорией.

readdir(Dir d)- получение очередного файла или папки.

### **Выполнение работы.**

Основная функция для работы с папкой- listdir, которая принимает первым аргументом слово, которое надо получить по условию задачи, вторым аргументом путь файла, третьим-путь корня и 4-файл, куда записывается результат.

Если объект есть файл типа txt, то проверяется, является очередной файл нужным, если да, то путь записывается в результирующий файл, иначе файл пропускается. Если текущий объект является директорией, то мы заходим рекурсивно в эту директорию.

Как только ответ будет найден, рекурсия силовым путем будет завершена.

### **Тестирование.**

Здесь результаты тестирования, которые помещаются на одну страницу.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	Дерево, которое дано в условии	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	Все работает верно в соответствии с заданием.

### **Выводы.**

Была исследована и изучена рекурсия, а также применено на практике взаимодействия с файлами. Применена рекурсия для обхода дерева файлов.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Сначала указываем имя файла, в котором код лежит в репозитории:

Название файла: main.cpp

```
#include <dirent.h>
#include <stdio.h>
#include <sys/types.h>
#include <string.h>
#include <stdlib.h>
void list_dir(char *s, char *dirPath, char *s2, FILE *f)
{
    DIR *dir = opendir(dirPath);
    if(dir) {
        struct dirent *de = readdir(dir);
        while (de)
            {if (strstr(de->d_name, ".txt"))
                {if (s[0]==de->d_name[0] && strlen(de-
>d_name)==5)
                    {
                        s++;
                        char
*s1=malloc(sizeof(char)*100);
                        strcpy(s1, dirPath);
                        strcat(s1, "/\0");
                        strcat(s1, de->d_name);
                        if (s)
                            fprintf(f, "%s\n", s1);
                        else
                            {fprintf(f, "%s", s1);
                                exit(1);}
                        //printf("%s\n", s2);
                        list_dir(s, s2, s2, f);
                        free(s1);
                    }
                }
            }
        else
            if (strcmp(de->d_name, ".\0") &&
strcmp(de->d_name, "..\0"))
                {
                    char
*s1=malloc(sizeof(char)*100);
                    strcpy(s1, dirPath);
                    strcat(s1, "/\0");
                    strcat(s1, de->d_name);
                    list_dir(s, s1, s2, f);
                    free(s1);
                }
        de=readdir(dir);
    }
    closedir(dir);
}
int main(int argc, char **argv){
    char *s=malloc(sizeof(char)*50);
    char *s1=malloc(sizeof(char)*100);
```

```
    fgets(s,100,stdin);
    FILE *f=fopen("result.txt","w");
    s1[0]='.';
    s1[1]='/';
    s1[2]='t';
    s1[3]='m';
    s1[4]='p';
    s1[5]='\0';
    list_dir(s,s1,s1,f);
    fclose(f);
    return 0;
}
```