

**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: Строки. Рекурсия, циклы, обход дерева.**

Студент гр. 1382

\_\_\_\_\_

Коренев Д.А.

Преподаватель

\_\_\_\_\_

Жангиров Т.Р.

Санкт-Петербург

2022

### **Цель работы.**

Освоение работы с файлами и обходом дерева, научиться использовать это с помощью рекурсии.

### **Задание (Вариант 2).**

Задана иерархия папок и файлов по следующим правилам:

- название папок может быть только "add" или "mul"
- В папках могут находиться другие вложенные папки и/или текстовые файлы
- Текстовые файлы имеют произвольное имя с расширением .txt
- Содержимое текстовых файлов представляет собой строку, в которой через пробел записано некоторое количество целых чисел

Требуется написать программу, которая, запускается в корневой директории, содержащей одну папку с именем "add" или "mul" и вычисляет и выводит на экран результат выражения состоящего из чисел в поддиректориях по следующим правилам:

- Если в папке находится один или несколько текстовых файлов, то математическая операция определяемая названием папки (add = сложение, mul = умножение) применяется ко всем числам всех файлов в этой папке
- Если в папке находится еще одна или несколько папок, то сначала вычисляются значения выражений, определяемые ими, а после используются уже эти значения

### **Выполнение работы.**

В функции main задаем в массив char'ов path и tmp название файла и путь директории соответственно. В открытый файл file

записывается значение возвращаемое функцией `readAddDir` с аргументом `tmp`.

Функция `readAddDir` принимает указатель на массив `char`'ов – путь к директории. С помощью функции `opendir()` открывает поток каталога советующий аргументу функции, указатель записывается в `dirArr` типа `DIR*`. Переменная `cnt` хранит значение сумм и будет возвращено по окончании работы функции. Если удалось открыть поток каталога, то с помощью функции `readdir()` записываем указатель на запись в каталоге в переменную структуры `dirent*`. Далее проверяем поле `d_name` переменной `dir`: Если оно соответствует “add” в переменную, которая хранит путь к директории добавляется “/add” и вызывается функция `readAddDir` от нее, значение суммируется с переменной `cnt`. Чтобы функция могла обработать последующие файлы и директории, добавленная ранее строка “/add” обрезается. Если поле `d_name` соответствует “/mul” происходит тоже самое что описано выше, но вместо функции `readAddDir` вызывается `readMulDir`, значение суммируется с переменной `cnt`. Если `d_name` не соответствует ничему из вышеописанного – в директории хранится файл, содержимое которого надо сложить и прибавить в переменную `cnt`. Когда работы с одной директорией/файлом была завершена, в переменную `dir` снова “читается” значение следующей директории/файла. Функция “закрывает” чтение и доступ к текущей директории и возвращает значение переменной `cnt`.

Функция `readMulDir` принимает указатель на массив `char`'ов – путь к директории. Переменная хранящая возвращаемый результат функции – `cnt` – равна 1. Аналогично вышеописанной функции `readAddDir` она “читает” все директории/файлы, но возвращаемое значение от вызова внутри этой функции других функций, а так же чтения файлов, умножается на переменную `cnt`.

## Тестирование.

Результаты тестирования представлены в таблице 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1	file.txt: 1 file1.txt: 1 file2.txt: 2 2 file3.txt: 7 file4.txt: 1 2 3 file5.txt: 3 -1 tmp->add->add->file.txt ->file1.txt ->mul->file2.txt ->file3.txt ->add->file4.txt ->file5.txt	226	Программа работает верно.
2	file.txt: 1 2 -4 -5 7 file1.txt: 3 0 file2.txt: 1 2 -3 file3.txt: 3 file4.txt: -10 tmp->mul->add->file.txt ->file1.txt ->mul->file2.txt ->file3.txt ->mul->file4.txt	720	Программа работает верно.

## Выводы.

Я научился работать с файлами с расширением .txt: читать информацию из них, делать обход дерева – директорий, используя рекурсивный подход.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название программы: PR\_Korenev\_Danil\_lb3.c

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

long long readMulDir(char* mul);

long long readAddDir(char* rootPath){
    //потенциальный файл с числами
    FILE *file;
    //открыли директорию
    DIR *dirArr = opendir(rootPath);
    long long cnt = 0;
    int c;
    long long value;

    if (dirArr){
        //взяли элемент директории
        struct dirent *dir = readdir(dirArr);
        while (dir){

            //если не назад
            if (dir->d_name[0] == '.'){
                dir = readdir(dirArr);
                continue;
            }

            //папка add
            if (strcmp(dir->d_name, "add") == 0) {
                strcat(rootPath, "/add");
                cnt += readAddDir(rootPath);
                rootPath[strlen(rootPath) - 4] = '\\0';
            }

            //папка mul
            else if (strcmp(dir->d_name, "mul") == 0){
                strcat(rootPath, "/mul");
                cnt += readMulDir(rootPath);
                rootPath[strlen(rootPath) - 4] = '\\0';
            }

            //файл в папке add -> надо прибавить в cnt сумму
            чисел файла
            else {
                strcat(rootPath, "/");
                strcat(rootPath, dir->d_name);
                file = fopen(rootPath, "r");
                while (c != EOF && (char)c != '\\n' &&
fscanf(file, "%lld", &value) != EOF){
                    cnt += value;
                }
                fclose(file);
            }
        }
    }
}
```

```

        rootPath[strlen(rootPath) - strlen(dir->d_name)
- 1] = '\\0';
    }

    //к следующей папке/файлу
    dir = readdir(dirArr);
}
}
closedir(dirArr);
return cnt;
}

long long readMulDir(char* rootPath){
    FILE *file;
    DIR *dirArr = opendir(rootPath);
    long long cnt = 1;
    int c;
    long long value;
    if(dirArr){
        struct dirent* dir = readdir(dirArr);
        while (dir){

            //если не назад
            if (dir->d_name[0] == '.'){
                dir = readdir(dirArr);
                continue;
            }

            //папка add
            if (strcmp(dir->d_name, "add") == 0) {
                strcat(rootPath, "/add");
                cnt *= readAddDir(rootPath);
                rootPath[strlen(rootPath) - 4] = '\\0';
            }

            //папка mul
            else if (strcmp(dir->d_name, "mul") == 0){
                strcat(rootPath, "/mul");
                cnt *= readMulDir(rootPath);
                rootPath[strlen(rootPath) - 4] = '\\0';
            }

            //файл в папке mul -> надо умножить cnt на
            //произведение чисел файла
            else {
                strcat(rootPath, "/");
                strcat(rootPath, dir->d_name);
                file = fopen(rootPath, "r");
                while (c != EOF && c != '\\n' && fscanf(file,
"%lld", &value) != EOF){
                    cnt *= value;
                }
                fclose(file);
                rootPath[strlen(rootPath) - strlen(dir->d_name)
- 1] = '\\0';
            }
        }
    }
}

```

```

        dir = readdir(dirArr);
    }

}
closedir(dirArr);
return cnt;
}

int main() {
    char path[1000] = "./result.txt";
    char tmp[1000] = "./tmp";

    FILE *file = fopen(path, "w");
    fprintf(file, "%lld", readAddDir(tmp));

    return 0;
}

```