

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

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «Программирование»**  
**Тема: «Обход файловой системы». Вариант 2.**

Студент гр. 3343

Калиберов Н.И

Преподаватель

Государкин Я.С.

Санкт-Петербург

2024

## **Цель работы**

Научиться работать с файлами и директориями. Написать, с использованием полученных знаний, программу на языке Си по рекурсивному поиску нужных файлов среди директорий.

## Задание

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

Сначала считывается строка и открывается файл *result.txt* в режиме редактирования. Для каждой директории в директории *tmp* применяется функция *check\_dir*.

*check\_dir* – рекурсивная функция, принимающая в качестве аргументов путь до директории и *is\_add* — числовое значение обозначающее математическую операцию( 1 — сложение, 0 - умножение). Сперва открывается директория *dir* с помощью *readdir()* и начинает считывание содержимого всех файлов директории. Затем, в зависимости от названия папки, считает и записывает результат обработки файла в переменную *res*. Так рекурсивный алгоритм проходится по всему дереву, пока не находит директорию, в которой нет директории, но есть файлы. После обработки файлов, они закрываются. После окончания рекурсивных вызовов директория закрывается.

## **Выводы**

Во время выполнения лабораторной работы, мы ознакомились с синтаксисом языка Си по работе с директориями и файлами, а также написали программу по рекурсивной обработке файловой системы и поиску файлов в ней.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

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

int calculate_expression(const char* directory) {
    // Проверяем, является ли директория папк
ой "add" или "mul"
    if (strcmp(directory, "add") == 0 || strcmp(directory, "mul") ==
0) {
        int result = (strcmp(directory, "add") == 0) ? 0 : 1;
        int file_count = 0;
        int subdirectory_count = 0;

        // Открываем текущую директорию
        DIR* dir = opendir(directory);
        if (dir == NULL) {
            printf("Ошиб ка при отк рытии ди рек тор и
и: %s\n", directory);
            exit(1);
        }

        struct dirent* entry;

        // Обходим все файлы и поддиректории вн
у три текущей директории
        while ((entry = readdir(dir)) != NULL) {
            // Игнорируем "." и ".."
            if (strcmp(entry->d_name, ".") == 0 ||
strcmp(entry->d_name, "..") == 0) {
                continue;
            }

            // Формируем полный путь к текущему э
лементу
            char path[PATH_MAX];
            snprintf(path, PATH_MAX, "%s/%s", directory,
entry->d_name);

            if (entry->d_type == DT_REG && strstr(entry->d_name,
".txt") != NULL) {
                // Если текущий элемент - текстовы
й файл, выполняем операцию
                FILE* file = fopen(path, "r");
                if (file == NULL) {
                    printf("Ошиб ка при отк рытии ф а й л
а: %s\n", path);
                    exit(1);
                }
            }
        }
    }
}
```

```

        int value;
        while (fscanf(file, "%d", &value) == 1) {
            if (strcmp(directory, "add") == 0) {
                result += value;
            } else {
                result *= value;
            }
        }

        fclose(file);
        file_count++;
    } else if (entry->d_type == DT_DIR) {
        // Если текущий элемент - поддирек
        тория, рекурсивно вычисляем значение выражения
        внутри нее

        int sub_result = calculate_expression(path);
        if (sub_result == -1) {
            printf("Ошибка при вычислении выр
ажения в папке: %s\n", path);
            exit(1);
        }

        if (strcmp(directory, "add") == 0) {
            result += sub_result;
        } else {
            result *= sub_result;
        }

        subdirectory_count++;
    }
}

closedir(dir);

// Если в текущей директории нет файлов
и поддиректорий, возвращаем -1 (ошибка)
if (file_count == 0 && subdirectory_count == 0) {
    return -1;
}

return result;
}

// Если текущая директория не является па
пкой "add" или "mul", возвращаем -1 (ошибка)
return -1;
}

int main() {
    const char* root_directory = ".";
    int result = calculate_expression(root_directory);
    FILE *output = fopen("result.txt", "w");
    if(output){
        fprintf(output, printf("%d\n", result));
        fclose(output);
    }
}

```

```
    return 0;  
}
```