

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. 9381

Колованов Р.А.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2020

Цель работы.

Изучение работы с файлами и деревом файловой системы; получение навыков работы с функциями библиотеки *dirent.h*, предназначенной для работы с файловой системой.

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

Ваше решение должно находиться в директории */home/box*, файл с решением должен называться *solution.c*. Результат работы программы должен быть записан в файл *result.txt*. Ваша программа должна обрабатывать директорию, которая называется *tmp*.

Выполнение работы.

Для решения поставленной задачи была разработана функция *recCalculateFiles()*, которая производит рекурсивный обход всех поддиректорий и вычисляет значения для каждой из них по правилам задания.

Функция *recCalculateFiles()*.

long long int recCalculateFiles(const char path, char operation);*

Принимает на вход путь к директории, с которой будет начинаться рекурсивный обход, и операцию, которую нужно применить к числам в файлах этой директории и к результатам поддиректорий, которые будут вычислены при помощи функции *recCalculateFiles()*. Возвращает результат операций, произведенных для файлов и поддиректорий, для переданной в качестве аргумента директории.

Принцип работы:

Для начала объявляется переменная *result*, которая будет содержать результат работы функции. Инициализируется значением 0 или 1 в зависимости от переданной в функцию операции (умножение или деление). Единица требуется для того, чтобы при перемножении результат не обнулялся. Далее при помощи функции *opendir()*, в которую передается путь к директории, получаем экземпляр структуры *DIR* для доступа к директории и записываем указатель на экземпляр в переменную *directory*. Если доступ был получен (*directory != NULL*), то при помощи цикла *while* производится обход содержимого данной директории. Далее объявляется указатель на экземпляр структуры *dirent*, получаемый из функции *readdir()*. В экземпляре хранится основная информация об одном содержимом директории *directory*, которая передается в функцию *readdir()* в качестве параметра. При повторном вызове функция вернет информацию о следующем содержимом директории. В цикле мы обрабатываем каждое содержимое директории *directory* до тех

пор, пока не пройдемся по всему содержимому. Создается строка *newDirectory*, которая содержит путь к данному содержимому, который получается конкатенацией пути директории, символа '/' и названия содержимого. Далее происходит определение типа содержимого: файл или папка. Если это файл, то происходит считывание последовательности чисел из него и для всех них выполняется операция, переданная в функцию. Полученное значение потом также умножается или суммируется с переменной *result*. Открытие файла происходит при помощи функции *fopen()*, после окончания работы файл закрывается при помощи функции *fclose()*. Если же содержимое является папкой, то для нее запускается та же функция *recCalculateFiles()*, результат которой потом суммируется или умножается с переменной *result*.

После происходит вызов функции *closedir()* для завершения работы с директорией, а переменная *result* возвращается из функции.

Так как внутри функции содержатся вызовы самой себя, функция является рекурсивной.

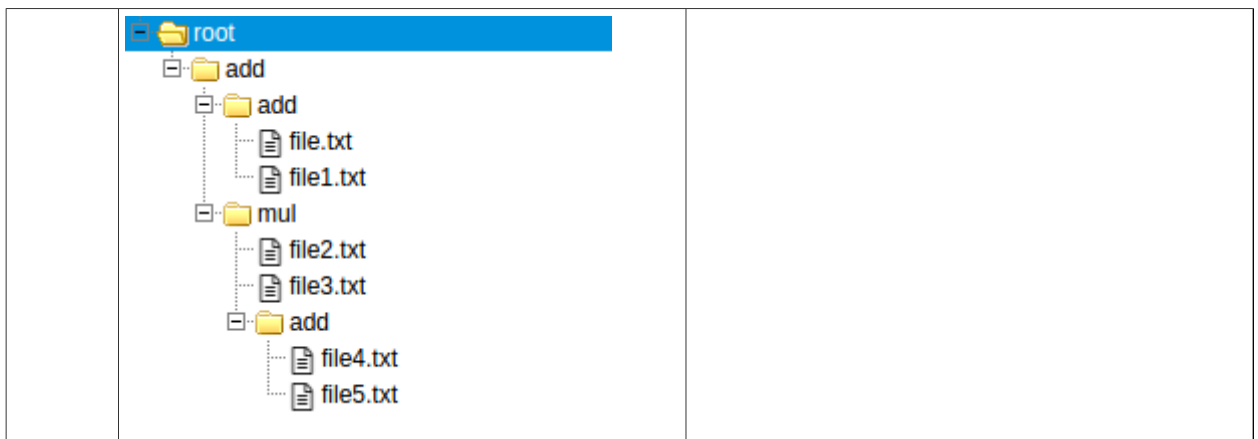
Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные
1.	file.txt: 1 file1.txt: 1 file2.txt: 2 2 file3.txt: 7 file4.txt: 1 2 3 file5.txt: 3 -1	226



Выводы.

Было изучена работа с файлами и деревом файловой системы в языке Си; получены навыки работы с функциями для работы с файловой системой библиотеки *dirent.h*.

Разработана программа, 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 определенным правилам.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

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

typedef struct dirent DirEnt;

long long int recCalculateFiles(const char* path, char operation) {
    long long int result = (operation == '*')? 1 : 0;
    DIR* directory = opendir(path);

    if (directory != NULL) {
        DirEnt* de = readdir(directory);

        while (de != NULL) {
            char* newDirectory = calloc(strlen(path) + 258,
sizeof(char));

            strcat(newDirectory, path);
            strcat(newDirectory, "/");
            strcat(newDirectory, de->d_name);

            if (de->d_type == DT_DIR) {
                char action = '?';

                if (strcmp(de->d_name, "add") == 0) {
                    action = '+';
                } else if (strcmp(de->d_name, "mul") == 0) {
                    action = '*';
                }

                if (action != '?') {
                    if (operation == '+') {
                        result += recCalculateFiles(newDirectory,
action);
                    } else if (operation == '*') {
                        result *= recCalculateFiles(newDirectory,
action);
                    }
                }
            }

            } else if (de->d_type == DT_REG) {
                FILE* file = fopen(newDirectory, "r");

                if (file != NULL) {
                    long long int value;

                    while (fscanf(file, "%lld", &value) == 1) {
                        if (operation == '+') {
                            result += value;
                        } else {
                            result *= value;
                        }
                    }
                }
            }
        }
    }
}
```

```

        }
    }
}

fclose(file);
}

free(newDirectory);
de = readdir(directory);
}

}

closedir(directory);

return result;
}

int main() {
    long long int result = recCalculateFiles("/home/box/tmp", '+');
    FILE* file = fopen("/home/box/result.txt", "w");

    if (file != NULL) {
        fprintf(file, "%lld", result);
        fclose(file);
    }

    return 0;
}

```