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. 9383 _____ Корсунов А.А.

Преподаватель _____ Размочаева Н.В.

Санкт-Петербург
2020

Цель работы.

Изучить работу с директориями и файлами путем изучения соответствующих библиотек, а также реализовать запрошенную программу с использованием рекурсии на языке СИ.

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

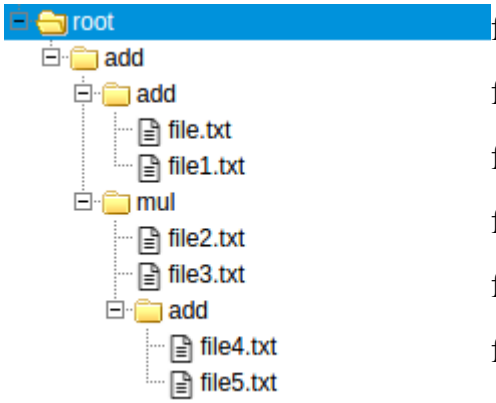
Требуется написать программу, которая, запускается в корневой директории, содержащей одну папку с именем "add" или "mul" и вычисляет и выводит на экран результат выражения состоящего из чисел в поддиректориях по следующим правилам:

●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

Пример

(Программа в момент запуска находится в директории root)



file.txt: 1

file1.txt: 1

file2.txt: 2 2

file3.txt: 7

file4.txt: 1 2 3

file5.txt: 3 -1

Решение:

226

Выражение в данном случае имеет вид: (((1+1))+((1+2+3+3+-1)*7*2*2))

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.

Описание функций:

1) **useFile(char *filename, char *condition);**

Данная функция считывает текст из очередного файла, разбивает его на части и производит соответствующую математическую операцию (в зависимости от аргумента *condition), после чего возвращает результат операции;

2) **useDir(char *direction, char *condition, int *outcome);**

Данная функция производит обход всех папок по очереди и считает необходимые по заданию значения с помощью вышеописанной функции. Коечный результат складывается в переменную *outcome. В конце производится возврат значения *outcome.

3) **main()**;

В данной функции производится самое первое открытие директории и вызов функции useDir,

Вывод.

В ходе проделанной работы была изучена работа с директориями и фалами, изучены соответствующие библиотеки. Разработана требуемая по заданию программа с использованием рекурсий.

Приложение А
Исходный код программ

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

int useFile(char *filename, char *condition){
        FILE *current_file = fopen(filename, "r");
        char message[100];
        fgets(message, 100, current_file);
        char *flow;
        int add = 0;
        int mul = 1;
        flow = strtok(message, " '\t''\n'");
   while(flow){
                if(strcmp(condition, "add") == 0){
          add = add + atoi(flow);
                }
                if(strcmp(condition, "mul") == 0){
                        mul = mul * atoi(flow);
                }
      flow = strtok(NULL, " '\t''\n'");
   }
        fclose(current_file);
        if(strcmp(condition, "add") == 0){
           return add;
        }
        if(strcmp(condition, "mul") == 0){
           return mul;
        }
}

int useDir(char *direction, char *condition, int *outcome){
        char direction_tmp[300] = "";
        strcpy(direction_tmp, direction);
        DIR *dir = opendir(direction);
        struct dirent *de = readdir(dir);
        while(de){
                while(de && strcmp(de->d_name, ".") && strcmp(de->d_name, "..")){
                        if(de->d_type == DT_DIR){
                                int count = 0;
```

```
                        if(strcmp(de->d_name, "mul") == 0){
                            count = 1;
                        }
                        int len = strlen(direction_tmp);
                        strcat(direction_tmp, "/");
                        strcat(direction_tmp,de->d_name);
                        if(strcmp(condition, "add") == 0){
                            *outcome = *outcome + useDir(direction_tmp, de->d_name, &count);
                        }
                        if(strcmp(condition, "mul") == 0){
                            *outcome = *outcome * useDir(direction_tmp, de->d_name, &count);
                        }
                        direction_tmp[len] = '\0';
                    }
                    if (de->d_type == DT_REG){
                        int len = strlen(direction_tmp);
                        strcat(direction_tmp, "/");
                        strcat(direction_tmp,de->d_name);
                        if(strcmp(condition, "add") == 0){
                            *outcome = *outcome + useFile(direction_tmp, condition);
                        }
                        if(strcmp(condition, "mul") == 0){
                            *outcome = *outcome * useFile(direction_tmp, condition);
                        }
                        direction_tmp[len] = '\0';
                    }
                    de = readdir(dir);
                }
                de = readdir(dir);
        }
        closedir(dir);
        return *outcome;
}

int main(){
        char direction[10000] = "./tmp";
        DIR *dir = opendir(direction);
        struct dirent *de;
        de = readdir(dir);
        while(strcmp(de->d_name, ".") == 0 || strcmp(de->d_name, "..") == 0){
                de = readdir(dir);
```

```
        };
        FILE *result = fopen("result.txt", "w");
        int outcome = 0;
        if(strcmp(de->d_name, "mul") == 0){
            outcome = 1;
        }
        fprintf(result, "%d\n", useDir(direction, de->d_name, &outcome));
        fclose(result);
        closedir(dir);
        return 0;
}
```