**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: Рекурсивный обход директорий

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Зыль С. Е. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

[**Цель работы** 3](#_Toc483344455)

[**Задание** 3](#_Toc483344456)

[**Содержание** 3](#_Toc483344457)

[**Исходный код** 4](#_Toc483344458)

[**Вывод** 5](#_Toc483344459)

# **Цель работы**

Реализация программы, которая вычисляет и выводит на экран результат выражения, вычисляемый математической операцией, определяемой названием поддиректории, над всеми числами всех файлов в этой папке.

# **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

# **Содержание**

Для решения данной задачи была использована рекурсивная функция **int recDir (const char\* start\_dir, int option),** которая получает путь к директории и модификатор функции:

* 0 - соответствует корневой директории
* 1 - соответствует add-директории
* 2 - соответствует mul-директории.

С помощью функций DIR\* opendir(const char\* path),

struct dirent\* readdir(DIR\* dir) и closedir(DIR\* dir) из библиотеки <dirent.h> выполняется открытие и закрытие потока каталога и чтения файлов из директории.

Если встречается текстовый файл, то программа открывает его и считывает данные, применяя к ним операцию согласно модификатору функции. Если функция readdir() вернёт имя вложенного каталога, то программа по имени каталога определяет новый модификатор функции и вызывает функцию reсDir() для этого каталога. Итоговое значение получается с помощью оператора **return.**

# **Исходный код**

|  |
| --- |
| #include <stdio.h> |
|  | #include <stdlib.h> |
|  | #include <string.h> |
|  | #include <dirent.h> |
|  | #include <sys/types.h> |
|  |  |
|  | int recDir(const char\* start\_dir, int option){ |
|  | char current\_path[1000]; |
|  | char string[200]; |
|  | strcpy(current\_path, start\_dir); |
|  | int result; |
|  | if (option == 1) |
|  | result = 0; |
|  | else if (option == 2) |
|  | result = 1; |
|  | DIR\* dir = opendir(current\_path); |
|  | struct dirent\* de = readdir(dir); |
|  |  |
|  | if(dir) |
|  | while(de){ |
|  | if((strstr(de->d\_name, ".txt")) && (option!=0)){ |
|  | int path\_len = strlen(current\_path); |
|  | strcat(current\_path, "/"); |
|  | strcat(current\_path, de->d\_name); |
|  | FILE\* file = fopen(current\_path, "r"); |
|  | current\_path[path\_len] = '\0'; |
|  | fgets(string, 200, file); |
|  | if(string[0] != '\n'){ |
|  | int digit; |
|  | char\* digits = strtok(string," \n"); |
|  | while(digits){ |
|  | sscanf(digits, "%d", &digit); |
|  | if(option == 1) |
|  | result+=digit; |
|  | else if(option == 2) |
|  | result\*=digit; |
|  | digits = strtok(NULL, " \n"); |
|  | } |
|  | } |
|  | fclose(file); |
|  | } |
|  |  |
|  | else{ |
|  | if(strcmp(".", de->d\_name)!=0 && strcmp("..", de->d\_name)!=0 && de->d\_type == DT\_DIR){ |
|  | int path\_len = strlen(current\_path); |
|  | strcat(current\_path, "/"); |
|  | strcat(current\_path, de->d\_name); |
|  | if(strcmp(de->d\_name, "add") == 0){ |
|  | if(option == 1) |
|  | result+=recDir(current\_path, 1); |
|  | else if(option == 2) |
|  | result\*=recDir(current\_path, 1); |
|  | else |
|  | result=recDir(current\_path, 1); |
|  | } |
|  | else if(strcmp(de->d\_name, "mul") == 0){ |
|  | if(option == 1) |
|  | result+=recDir(current\_path, 2); |
|  | else if(option == 2) |
|  | result\*=recDir(current\_path, 2); else |
|  | result=recDir(current\_path, 2); } |
|  | current\_path[path\_len] = '\0'; |
|  | } |
|  | } |
|  |  |
|  | de = readdir(dir); |
|  | } |
|  | closedir(dir); |
|  | return result; |
|  | } |
|  |  |
|  |  |
|  | int main(){ |
|  | printf("%d\n", recDir(".", 0)); |
|  | return 0; |
|  | } |

# **Вывод**

В ходе данной лабораторной работы были получены знания по реализации алгоритма обхода директорий с помощью рекурсивной функции.