**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 |  | Ковынев М.В. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

**Содержание**

[**Цель работы 2**](#_Toc483142207)

[**Задание 3**](#_Toc483142208)

[**Содержание 3**](#_Toc483142209)

[**Вывод 6**](#_Toc483142210)

# **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**

/\*

\* Файл lab3\_progin\_c\_2sem.c

\* Обход дерева папок

\* Автор: kovinevmv

\* 21.05.2017

\*/

#include <stdio.h>

#include <dirent.h>

#include <string.h>

/\*=============================================================

Функция readFile получает в качестве аргументов путь к файлу

current\_path и имя папки folderName (add или mul), в которой

находится файл. Функция выполняет определенные арифметические

операции с числами из файла в зависимости от имени папки и

возвращает результат.

\*/

long long int readFile(char\* current\_path, char\* folderName);

//=============================================================

/\*=============================================================

Функция searchInDir получает в качестве аргументов текущий

путь и папку, в которой находится текущий каталог (add или mul)

Если в директории есть другая директория, тогда рекурсивно

вызывается функция searchInDir, которая добавляет или умножает

текущий результат на возвращаемое значение. Аналогичные действия

выполняются если в директории есть файл TXT, только вызывается

функция readFile.

\*/

long long int searchInDir(const char\* path, char\* prevFolderName);

//=============================================================

int main()

{

printf("\nTotal result: %lld\n\n", searchInDir(".", ""));

return 0;

}

long long int readFile(char\* current\_path, char\* folderName)

{

long long int result = 0; // Результат выполнения операций

int mathOperation; // Математическая операция, которая будет выполнена

// с числами из файла: 0 - сложение, 1 - умножение

// Распознавание математической операции

// mul - умножение, add - сложение

if (!strcmp(folderName, "mul"))

{

result = 1;

mathOperation = 1;

}

else if (!strcmp(folderName, "add"))

{

result = 0;

mathOperation = 0;

}

// Открытие файла

FILE\* fileTXT = fopen(current\_path, "r");

// Считываем каждое число пока файл не закончится

while (!feof(fileTXT))

{

int value;

fscanf(fileTXT, "%d ", &value);

// Выполнение соответствующей операции

if (mathOperation)

{

result \*= value;

}

else

{

result += value;

}

}

// Вывод результата выполнения действий в файле, имеющего путь current\_path

printf("Result: %lld\t File: %s\n", result, current\_path);

fclose(fileTXT);

return result;

}

long long int searchInDir(const char\* path, char\* prevFolderName)

{

long long int result = 0;

int mathOperation = 0;

// Распознавание математической операции

// mul - умножение, add - сложение

if (!strcmp(prevFolderName, "mul"))

{

result = 1;

mathOperation = 1;

}

else if (!strcmp(prevFolderName, "add"))

{

result = 0;

mathOperation = 0;

}

char current\_path[10000];

strcpy(current\_path, path);

// Открываем текущую директорию

DIR \*dir = opendir(current\_path);

struct dirent \*de = readdir(dir);

if (dir)

{

while (de)

{

// current\_path - полный путь к текущему файлу открытой директории

int path\_len = strlen(current\_path);

strcat(current\_path, "/");

strcat(current\_path, de->d\_name);

// Если это не текущий и не родительский каталог,

// то рекурсивный вызов функции обхода searchInDir

if (de->d\_type == DT\_DIR && strcmp(".", de->d\_name)

&& strcmp("..", de->d\_name))

{

if (mathOperation)

{

result \*= searchInDir(current\_path, de->d\_name);

}

else

{

result += searchInDir(current\_path, de->d\_name);

}

}

// Если это файл, оканчивающийся .txt, то

// вызов функции readFile

else if (strstr(de->d\_name, ".txt") && (de->d\_type == 8))

{

if (mathOperation)

{

result \*= readFile(current\_path, prevFolderName);

}

else

{

result += readFile(current\_path, prevFolderName);

}

}

// Добавление символа конца строки

current\_path[path\_len] = '\0';

// Считывание нового элемента директории

de = readdir(dir);

}

}

closedir(dir);

return result;

}

# **Вывод**

Выполнив данную лабораторную работу, было освоено и закреплено на практике написание программы, обходящей дерево директорий рекурсивным методом и выполняющей математический операции над числами текстовых документов текущей директории.