# 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. 9303 |  | Жорже М.А |
| Преподаватель |  | Чайка К.В |

Санкт-Петербург 2020

**Цель работы.**

Изучить средства для работы с файловой системой и написать программу для рекурсивного обхода файлов.

**Задание.**

Задана иерархия папок и файлов по следующим правилам:

* название папок может быть только "add" или "mul"add"add" или "mul" или "add" или "mul"mul"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"add"add" или "mul" или "add" или "mul"mul"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 =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

**Основные теоретические положения.**

Основные функции для работы с деревом файловой системы, объявления которых находятся в заголовочном файле dirent.h (add =также, может понадобиться включить заголовочный файл sys/types.h)

Для того, чтобы получить доступ к содержимому некоторой директории можно использовать функцию

DIR \*opendir(add =const char \*dirname); Которая возвращает указатель на объект типа DIR с помощью которого можно из программы работать с заданной директорией. Тип DIR представляет собой поток содержимого директории. Для того, чтобы получить очередной элемент этого потока, используется функция struct dirent \*readdir(add =DIR \*dirp); Она возвращает указатель на объект

структуры dirent, в котором хранится информация о файле. Основной интерес представляют поля, хранящие имя и тип объекта в директории (add =это может быть не только "add" или "mul"файл"add" или "mul" и "add" или "mul"папка"add" или "mul").После завершения работы с содержимым директории, необходимо вызвать функцию int closedir(add =DIR \*dirp); Передав ей полученный функцией readdir(add =) ранее дескриптор.

**Выполнение работы.**

Программа открывает директорию ./tmp, затем смотрит какая содержится папка,в зависимости от этого вызывается та или иная функция,в которых рекурсивно обсчитывается результат. После результат сохраняется в файл result.txt.

Подробный код программы можно посмотреть в **Приложении A**. **Выводы.**

Была разработана программа, рекурсивно обходящая заданную директорию.Для этого были изучены функции по работе с файловой директорией.

# ПРИЛОЖЕНИЕ A ИСХОДНЫЙ КОД ПРОГРАММЫ

#include<stdio.h>

#include<string.h>

#include<dirent.h>

#include<sys/types.h>

int add(char\* path, char \*simb){

int res = 0, lenpath;

DIR \*dir = opendir(path);

struct dirent \*de;

FILE \*file;

int value;

if(simb == "mul")

res = 1;

while((de = readdir(dir)) != NULL){

lenpath = strlen(path);

if(strcmp(de->d\_name, "add") == 0){

strcat(path, "/");

strcat(path, de->d\_name);

if(simb == "add")

res += add(path, "add");

else if(simb == "mul")

res \*= add(path, "add");

path[lenpath] = '\0';

}

if(strcmp(de->d\_name, "mul") == 0){

strcat(path, "/");

strcat(path, de->d\_name);

if(simb == "add")

res += add(path, "mul");

else if(simb == "mul")

res \*= add(path, "mul");

path[lenpath] = '\0';

}

if(de->d\_name[strlen(de->d\_name) - 3] == 't' && de->d\_name[strlen(de->d\_name) - 2] == 'x' && de->d\_name[strlen(de->d\_name) - 1] == 't'){

strcat(path, "/");

strcat(path, de->d\_name);

file = fopen(path, "r");

if(simb == "add"){

while(fscanf(file, "%d", &value) != EOF)

res +=value;

}else if(simb == "mul"){

while(fscanf(file, "%d", &value) != EOF)

res \*=value;

}

path[lenpath] = '\0';

}

}

return res;

}

int get\_all\_files( DIR \* dir, char \* path){

struct dirent \*de;

int r = 0;

while((de = readdir(dir)) !=NULL)

{

if(strcmp(de->d\_name, "add") == 0){

strcat(path, "/");

strcat(path, de->d\_name);

r = add(path, "add");

}

if(strcmp(de->d\_name, "mul") == 0){

strcat(path, "/");

strcat(path, de->d\_name);

r = add(path, "mul");

}

}

return r;

}

int main(){

char caminho[100] = "root";

FILE \*result = fopen("result.txt", "w");

DIR \*dir;

int res;

if((dir = opendir(caminho)))

res = get\_all\_files(dir, caminho);

fprintf(result, "%d", res);

printf("%d\n", res);

closedir(dir);

fclose(result);

return 0;

}