**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе№3**

**по дисциплине «Программирование»**

Тема: «Обход файловой системы»

|  |  |  |
| --- | --- | --- |
| Студент гр. 3343 |  | Атоян М.А. |
| Преподаватель |  | Государкин Я.С. |

Санкт-Петербург

2024

# Цель работы

Научиться работать с файлами и директориями. Написать, с использованием полученных знаний, программу на языке Си по рекурсивному поиску нужных файлов среди директорий.

# Задание

Вариант 2.

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

## Выполнение работы

Сначала считывается строка и открывается файл *result.txt* в режиме редактирования. Для каждой директории в директории *tmp* применяется функция calculate.

int sumFile(FILE \*file) – возвращает сумму чисел в файле.

int mulFile(FILE \*file) - возвращает произведение чисел в файле.

int sumValues(int \*values, int valuesLen) - возвращает сумму чисел в массиве.

int mulValues(int \*values, int valuesLen) - возвращает произведение чисел в массиве.

int calculate(const char \*dirPath) - рекурсивная функция, принимающая в качестве аргументов путь до директории. Сначала открывается директория *dir* с помощью *readdir()* и считывается содержимое всех файлов директории. Затем, в зависимости от названия папки, считает и записывает результат обработки файла одну из ячеек массива значений values. Так рекурсивный алгоритм проходится по всему дереву, пока не находит директорию, в которой нет директории, но есть файлы. После обработки файлов, они закрываются. После окончания рекурсивных вызовов директория закрывается.

# Выводы

Во время выполнения лабораторной работы, мы ознакомились с синтаксисом языка Си по работе с директориями и файлами, а также написали программу по рекурсивной обработке файловой системы и поиску файлов в ней.

ПРИЛОЖЕНИЕ А  
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

#include <stdio.h>

#include <string.h>

#include <dirent.h>

int sumFile(FILE \*file) //возвращает сумму чисел в файле

{

int sum = 0;

int n;

while (fscanf(file, "%d", &n) == 1)

{

sum += n;

}

return sum;

}

int mulFile(FILE \*file) //возвращает произведение чисел в файле

{

int prod = 1;

int n;

while (fscanf(file, "%d", &n) == 1)

{

prod \*= n;

}

return prod;

}

int sumValues(int \*values, int valuesLen) //возвращает сумму чисел в массиве

{

int sum = 0;

for (int i = 0; i < valuesLen; i++)

{

sum += values[i];

}

return sum;

}

int mulValues(int \*values, int valuesLen) //возвращает произведение чисел в массиве

{

int prod = 1;

for (int i = 0; i < valuesLen; i++)

{

prod \*= values[i];

}

return prod;

}

int calculate(const char \*dirPath)

{

DIR \*dir = opendir(dirPath);

FILE \*file;

struct dirent \*entry;

char filePath[1000];

int values[100]; //для хранения результатов вычислений в файлах текущей директории

int valuesLen = 0;

if (dir)

{

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

{

sprintf(filePath, "%s/%s", dirPath, entry->d\_name);

if (entry->d\_type == DT\_DIR) //проверяем, является ли файл директорией

{

if (strcmp(entry->d\_name, ".") != 0 && strcmp(entry->d\_name, "..") != 0)

values[valuesLen++] = calculate(filePath);

}

else //иначе обрабатываем текстовый файл

{

file = fopen(filePath, "r");

if (strcmp(dirPath + strlen(dirPath) - 3, "add") == 0) //если директория называется add, то складываем числа в файле

values[valuesLen++] = sumFile(file);

else // иначе перемножаем

values[valuesLen++] = mulFile(file);

fclose(file);

}

}

}

int res = values[0];

if (strcmp(dirPath + strlen(dirPath) - 3, "add") == 0) //если директория называется add, то складываем полученные в ней значения

res = sumValues(values, valuesLen);

else //иначе перемножаем

res = mulValues(values, valuesLen);

return res;

}

int main()

{

char \*dirPath = "./tmp";

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

int n = calculate(dirPath);

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

fclose(result);

return 0;

}