**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 |  | Никишин С.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 walk\_directory.

*walk\_directory* – рекурсивная функция, принимающая в качестве аргументов путь до директории и mode — числовое значение обозначающее математическую операцию( 1 — сложение, -1 - умножение). Сперва открывается директория *dir* с помощью *readdir()* и начинаетсчитывание содержимого всех файлов директории. Затем, в зависимости от названия папки, считает и записывает результат обработки файла в переменную res. 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 <dirent.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

char\* pathcat(const char\* path1, const char\* path2) {

int res\_path\_len =

strlen(path1) + strlen(path2) +

2; // определение длины новой строки с учетом символов / и символа конца строки

char\* res\_path =

malloc(res\_path\_len \* sizeof(char)); // выделение памяти под новую строку

sprintf(res\_path, "%s/%s", path1,

path2); // форматный вывод данных в строку return res\_path;

return res\_path;

}

int walk\_directory(const char\* path, int mode) {

int temp;

int res = 0;

if (mode == -1)

res = 1;

DIR\* dir = opendir(path);

struct dirent\* entry;

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

if (strcmp(entry->d\_name, ".") == 0 || strcmp(entry->d\_name, "..") == 0) {

continue;

} else if (entry->d\_type == DT\_REG) {

char\* path\_file = pathcat(path, entry->d\_name);

FILE\* file = fopen(path\_file, "r");

if (file) {

char buffer[256];

fgets(buffer, 256, file);

char\* number = strtok(buffer, " ");

while (number != NULL) {

if (mode == 1) {

res += atoi(number);

} else if (mode == -1) {

res \*= atoi(number);

}

number = strtok(NULL, " ");

}

}

fclose(file);

} else if (entry->d\_type == DT\_DIR) {

char\* path\_file = pathcat(path, entry->d\_name);

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

temp = walk\_directory(path\_file, 1);

} else if (strcmp(entry->d\_name, "mul") == 0) {

temp = walk\_directory(path\_file, -1);

}

if (mode == 1) {

res += temp;

} else if (mode == -1) {

res \*= temp;

} else if (mode == 0) {

return temp;

}

}

}

closedir(dir);

return res;

}

int main() {

char\* path = "./tmp";

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

fprintf(file, "%d\n", walk\_directory(path, 0));

fclose(file);

printf("%d\n", walk\_directory(path, 0));

return 0;

}