**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. 2381 |  | Богатов И.А. |
| --- | --- | --- |
| Преподаватель |  | Тиняков С.А. |

Санкт-Петербург

2023

## Цель работы

Изучить принцип работы с рекурсивными алгоритмами и файловой системой на языке программирования Си.

## 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

Ваше решение должно находиться в директории **/home/box**, файл с решением должен называться **solution.c**. Результат работы программы должен быть записан в файл **result.txt**. Ваша программа должна обрабатывать директорию, которая называется **tmp**.

## Выполнение работы

Подключаются заголовочные файлы *stdio.h, stdlib.h, string.h, dirent.h, sys/types.h, sys/stat.h, unistd.h.*

### Структура dirent

Эта структура представляет собой структуру, используемую для хранения информации о директории и файлах внутри нее. Она определена в заголовочном файле dirent.h. Структура содержит следующие поля:

*d\_ino* - номер inode файла

*d\_off* - смещение структуры dirent в каталоге

*d\_reclen* - длина записи d\_type - тип файла (*S\_IFIFO, S\_IFCHR, S\_IFDIR, S\_IFBLK, S\_IFREG, S\_IFLNK, S\_IFSOCK*)

*d\_name* - имя файла

### Структура *stat*

Эта структура представляет собой структуру, используемую для хранения информации о файле. Она определена в заголовочном файле sys/stat.h. Структура содержит следующие поля:

st\_dev - идентификатор устройства, на котором находится файл

st\_ino - номер inode файла

st\_mode - режим доступа к файлу (маска прав доступа, определяющая тип файла и права доступа)

st\_nlink - количество ссылок на файл

st\_uid - идентификатор пользователя-владельца файла

st\_gid - идентификатор группы-владельца файла

st\_rdev - идентификатор устройства, если файл является устройством

st\_size - размер файла в байтах

st\_blksize - размер блока ввода-вывода

st\_blocks - количество выделенных блоков файлу

st\_atime - время последнего доступа к файлу

st\_mtime - время последней модификации файла

st\_ctime - время последнего изменения статуса файла

### Функция process\_directory

Рекурсивно обходит директорию, переданную ей в качестве аргумента path, и выполняет операцию, переданную ей в аргументе operation, над содержимым директории и всех ее поддиректорий. Если аргумент operation равен NULL, то функция по умолчанию выполняет операцию нахождения суммы чисел из файлов с расширением ".txt". Если operation равен "add", то функция вычисляет сумму чисел, а если operation равен "mul", то функция вычисляет произведение чисел.

### Функция snprintf

Формирует путь для каждого файла в директории.

### Функция main

Является точкой входа в программу. Она вызывает функцию process\_directory() для обработки директории "tmp" и передает ей второй аргумент NULL, что означает выполнение операции по умолчанию (получение суммы чисел из файлов с расширением ".txt" в директории и всех ее поддиректориях). Затем функция main() открывает файл "result.txt" для записи результата и записывает в него значение, возвращенное функцией process\_directory(). И наконец, функция main() закрывает файл и завершает программу.

Разработанный программный код см. в приложении А.

## Выводы

Была изучена библиотека dirent.h, принцип работы с рекурсивными алгоритмами и файловой системой на языке программирования Си. Была написана программа, которая находит пути до файлов, образующих заданное слово, и записывает результат работы в файл result.txt.

# ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <dirent.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <unistd.h>

int process\_directory(const char \*path, const char \*operation);

int main() {

int result = process\_directory("tmp", NULL);

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

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

fclose(output);

return 0;

}

int process\_directory(const char \*path, const char \*operation) {

if (!dir) {

perror("opendir");

exit(1);

}

struct dirent \*entry;

int result = (operation && strcmp(operation, "mul") == 0) ? 1 : 0;

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

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

continue;

}

char subpath[1024];

snprintf(subpath, sizeof(subpath), "%s/%s", path, entry->d\_name);

struct stat st;

if (stat(subpath, &st) == -1) {

perror("stat");

exit(1);

}

if (S\_ISDIR(st.st\_mode)) {

int subdir\_result = process\_directory(subpath, entry->d\_name);

if (operation) {

if (strcmp(operation, "add") == 0) {

result += subdir\_result;

} else if (strcmp(operation, "mul") == 0) {

result \*= subdir\_result;

}

} else {

result = subdir\_result;

}

} else if (S\_ISREG(st.st\_mode) && strstr(entry->d\_name, ".txt")) {

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

    if (!file) {

perror("fopen");

exit(1);

}

int number;

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

if (operation) {

if (strcmp(operation, "add") == 0) {

result += number;

} else if (strcmp(operation, "mul") == 0) {

result \*= number;

}

} else {

result = number;

}

}

fclose(file);

}

}

closedir(dir);

return result;

}