

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. 7303

Юсковец А.В.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2018

Оглавление

Цель работы.....3

Ход работы.....4

 Директивы препроцессора.....4

 Определение структуры.....4

 createMusicalComposition.....5

 createMusicalCompositionList.....5

 push.....6

 removeEl.....6

 count.....7

 print_names.....7

Вывод.....8

Цель работы

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

Ход работы

▪ Директивы препроцессора

```
#include <stdio.h>
#include <dirent.h>
#include <string.h>
#include <limits.h>
#include <stdlib.h>
```

Подключаются соответствующие библиотеки для использования функций: работы с вводом-выводом, динамической памятью, константы, работы со строками и с директориями.

- Функция для обработки файла

```
int func_file(char const* path, char const* op) {
    FILE* file = fopen(path, "r");
    if (file) {
        int res;
        int number;

        if (strcmp(op, "add") == 0) {
            res = 0;
            while (fscanf(file, "%d", &number) == 1) {
                res += number;
            }

            return res;
        } else if (strcmp(op, "mul") == 0) {
            res = 1;
            while (fscanf(file, "%d", &number) == 1) {
                res *= number;
            }
            printf("func_file: res: %d\n", res);
            return res;
        }
    }
}
```

path – путь к файлу.

op – операция, применяемая к числам в файле.

Если файл открылся:

объявляется две переменные: res – результат, number – текущее число.

В зависимости от операции считается результат, считывая числа из файла.

- Функция для обработки директории

```
int func(char const* path, char const* op) {
    DIR* dir = opendir(path);
    if (dir) {
        char local_path[PATH_MAX + 1];
        strcpy(local_path, path);
        strcat(local_path, "/");

        char dirent_path[PATH_MAX + 1];

        struct dirent* dirent;
        int res = (strcmp(op, "add") == 0) ? 0 : 1;
        if (res == 0) {
            while (dirent = readdir(dir)) {
                if (strcmp(dirent->d_name, ".") != 0 && strcmp(dirent->d_name, "..") != 0) {
                    strcpy(dirent_path, local_path);
                    strcat(dirent_path, dirent->d_name);
                    if (dirent->d_type == DT_REG) {
                        res += func_file(dirent_path, "add");
                    } else {
                        res += func(dirent_path, dirent->d_name);
                    }
                }
            }
        } else {
            while (dirent = readdir(dir)) {
                if (strcmp(dirent->d_name, ".") != 0 && strcmp(dirent->d_name, "..") != 0) {
                    strcpy(dirent_path, local_path);
                    strcat(dirent_path, dirent->d_name);
                    if (dirent->d_type == DT_REG) {
                        res *= func_file(dirent_path, "mul");
                    } else {
                        res *= func(dirent_path, dirent->d_name);
                    }
                }
            }
        }

        return res;
    } else {
        return 0;
    }
}
```

dir — директория, путь к которой передан в функцию,

Если директория открылась:

Создается local_path для дальнейшего удобства (конкатенирование данного пути с названием текущего dirent.

Инициализируется переменная res: если операция добавления — 0, умножения — 1.

Далее, пока не будет достигнут конец директории результат readdir записывается в dirent.

Затем в зависимости от того является ли очередной элемент директорией или файлом вызываются соответствующие функции (для директории или для файла). В функцию передается путь, созданный ранее путем конкатенации заготовленного ранее local_path и dirent->d_name.

Когда все содержимое директории будет пройдено функции вернет вычисленные res.

▪ main

```
int main() {
    DIR* dir = opendir("tmp");
    struct dirent* dirent;
    while (dirent = readdir(dir)) {
        if (strcmp(dirent->d_name, ".") != 0 && strcmp(dirent->d_name,
"..") != 0) {
            char path[PATH_MAX + 1] = "tmp";
            strcat(path, "/");
            strcat(path, dirent->d_name);

            FILE* result = fopen("result.txt", "w");
            fprintf(result, "%d\n", func(path, dirent->d_name));
        }
    }

    return 0;
}
```

Открывается папка tmp. Далее пока не попадется директория с нужным названием в dirent записывается значение readdir. Когда встречается директория с нужным названием(add или mul). Ее путь конкатенируется с tmp. Далее в файл result записывается результат функции для обработки директорий, которой переданы: путь к первой директории и ее название (операция – что делать с содержимым данной дериктории).

Вывод

Была написана программа («файлово-директорный калькулятор» :). При написании были использована рекурсия и инструменты для работы с файловой системой. При компиляции и выполнении программы не возникает ошибок и предупреждений.