

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. 3344

Пачев Д.К.

Преподаватель

Глазунов С.А.

Санкт-Петербург

2024

Цель работы

Написать программу на языке Си, в которой реализуется обход файловой системы.

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

Функция `count_numbers` принимает на вход путь до текстового файла и операцию, которую нужно произвести с числами в файле, возвращает функция результат выполнения операции.

Функция `check_first_directory` – функция для проверки первой операции, принимает на вход путь, и возвращает имя первой директории, то есть операцию.

Функция `traverse_directory` – принимает на вход путь и операцию. С помощью цикла `while` проходится по директории, если текущий файл является директорией, то вызывается рекурсивно функция `traverse_directory`, считает результат в этой директории. Если текущий файл - это текстовый файл, то открывается файл и выполняется операция над числами.

В функции `main` вызывается функция `check_first_directory`, далее в переменную `result` записывается результат функции `traverse_directory`, этот результат записывается в файл `result.txt`

Тестирование

Результаты тестирования представлены в Таблице 1

Таблица 1 - Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	Структура -tmp -add -add -file.txt (1) -file1.txt (1) -mul -file2.txt (2 2) -file3.txt (7) -add -file4.txt (1 2 3) -file5.txt (3 -1)	226	Верно

Выводы

В ходе лабораторной работы была написана программа на языке Си, которая позволяет обходить файловую систему и выполнять операции с числами в файлах.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

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

int count_numbers(char* path, char* operation){
    FILE *file;
    file = fopen(path, "r");
    if (file == NULL){
        printf("File not found\n");
        return 0;
    }

    if (strcmp(operation, "add") == 0){
        int sum = 0;
        int number;
        while (fscanf(file, "%d", &number) != EOF){
            sum += number;
        }
        return sum;
    }
    else if (strcmp(operation, "mul") == 0){
        int mul = 1;
        int number;
        while (fscanf(file, "%d", &number) != EOF){
            mul *= number;
        }
        return mul;
    }
}

char* check_first_directory(char* path){
    DIR* d;
    struct dirent *dir;
    d = opendir(path);
    if (d) {
        while ((dir = readdir(d)) != NULL) {
            if (dir->d_type == DT_DIR) {
                if (strcmp(dir->d_name, ".") != 0 && strcmp(dir->d_name,
"..") != 0 &&
                    strcmp(dir->d_name, ".DS_Store") != 0) {
                    return dir->d_name;
                }
            }
        }
    }
    closedir(d);
}
```

```

}

int traverse_directory(char* path, char* operation){
    int result;
    if (strcmp(operation, "add") == 0){
        result = 0;
    }
    else if (strcmp(operation, "mul") == 0){
        result = 1;
    }
    DIR *d;
    struct dirent *dir;
    d = opendir(path);
    if (d){
        while ((dir = readdir(d)) != NULL){
            if (dir->d_type == DT_DIR){
                if (strcmp(dir->d_name, ".") != 0 && strcmp(dir->d_name,
"..") != 0 && strcmp(dir->d_name, ".DS_Store") != 0){
                    char new_path[1000];
                    snprintf(new_path, sizeof(new_path), "%s/%s", path,
dir->d_name);

                    int subdir_result = traverse_directory(new_path, dir-
>d_name);

                    if (strcmp(operation, "add") == 0){
                        result += subdir_result;
                    }
                    else if (strcmp(operation, "mul") == 0){
                        result *= subdir_result;
                    }
                }
            }
        }
    }
    else{
        if (strstr(dir->d_name, ".txt")){
            char copy_path[1000];
            strcpy(copy_path, path);
            char *token = strtok(copy_path, "/");
            char *last_token = token;
            while (token != NULL){
                last_token = token;
                token = strtok(NULL, "/");
            }
            char new_path[1000];
            snprintf(new_path, sizeof(new_path), "%s/%s", path,
dir->d_name);

            if (strcmp(last_token, "add") == 0){
                int res = count_numbers(new_path, "add");
                result += res;
            }
            else if (strcmp(last_token, "mul") == 0){
                int res = count_numbers(new_path, "mul");
                result *= res;
            }
        }
    }
}

```



```

        }
    }
    closedir(d);
}
return result;
}

int main(void) {
    char* first_directory = check_first_directory("tmp");
    int result = traverse_directory("tmp",
                                   first_directory);
    FILE *output = fopen("result.txt", "w");
    fprintf(output, "%d\n", result);
    fclose(output);

    return 0;
}

```