

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

Старков 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 начинает считывание 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 <stdio.h>
#include <dirent.h>
#include <string.h>

#include <stdlib.h>

void read_numfile(char *filepath, int mode, int *res)
{
    FILE *f = fopen(filepath, "r");
    if (f == NULL)
    {
        return;
    }
    char string[256];
    fgets(string, 256, f);
    char *number = strtok(string, " ");
    while (number != NULL)
    {
        if (mode == 1)
        {
            (*res) += atoi(number);
        }
        if (mode == -1)
        {
            (*res) = (*res) * atoi(number);
        }

        number = strtok(NULL, " ");
    }
    fclose(f);
}

char *file_allocpath(char *path, char *name)
{
    int res_len = strlen(path) + strlen(name) + 2;
    char *res = calloc(res_len, sizeof(char));
    sprintf(res, "%s/%s", path, name);
    return res;
}

int dir_lookup(char *root, int mode)
{
    int tmp;
    int res = 0;
    if (mode == 1)
    {
        res = 0;
    }
    if (mode == -1)
    {

```

```

        res = 1;
    }
    DIR *root_dir = opendir(root);
    struct dirent *de;
    while ((de = readdir(root_dir)) != NULL)
    {
        char *new_path = file_allopath(root, de->d_name);
        if (strcmp(de->d_name, ".") == 0 || strcmp(de->d_name, "..")
== 0)
        {
            continue;
        } // когда открывается файл

        // когда открывается папка
        else if (de->d_type == 4)
        {

            // tmp = dir_lookup(:MODE = 1)
            if (strcmp(de->d_name, "add") == 0)
            {
                tmp = dir_lookup(new_path, 1);
            }
            else if (strcmp(de->d_name, "mul") == 0)
            {
                tmp = dir_lookup(new_path, -1);
            }

            if (mode == 1)
            {
                res = res + tmp;
            }
            else if (mode == -1)
            {
                res = res * tmp;
            }
            else if (mode == 0)
            {
                return tmp;
            }
        }
        else
        {

            read_numfile(new_path, mode, &res);
        }
    }
    closedir(root_dir);
    return res;
}

int main()
{
    char *dir = "./tmp";
    FILE *file = fopen("./result.txt", "w");
    fprintf(file, "%d\n", dir_lookup(dir, 0));
    fclose(file);
}

```

```
    printf("%d\n", dir_lookup(dir, 0));  
    return 0;  
}
```