

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

Лихацкий В.Р.

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 *calculate*.

`int add(FILE *file)` – возвращает сумму чисел в файле.

`int mul(FILE *file)` - возвращает произведение чисел в файле.

`int addValues(int *values, int len)` - возвращает сумму чисел в массиве.

`int mulValues(int *values, int len)` - возвращает произведение чисел в массиве.

`int calc(const char *path)` - рекурсивная функция, принимающая в качестве аргументов путь до директории. Сначала открывается директория *dir* с помощью *readdir()* и считывается содержимое всех файлов директории. Затем, в зависимости от названия папки, считает и записывает результат обработки файла одну из ячеек массива значений *values*. 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 <string.h>
#include <dirent.h>

int add(FILE *file)
{
    int res = 0;
    int n;
    while (fscanf(file, "%d", &n) == 1) res += n;
    return res;
}

int mul(FILE *file)
{
    int res = 1;
    int n;
    while (fscanf(file, "%d", &n) == 1) res *= n;
    return res;
}

int addValues(int *values, int len)
{
    int res = 0;
    for (int i = 0; i < len; i++) res += values[i];
    return res;
}

int mulValues(int *values, int len)
{
    int res = 1;
    for (int i = 0; i < len; i++) res *= values[i];
    return res;
}

int calc(const char *root)
{
    DIR *dir = opendir(root);
    FILE *file;
    struct dirent *entry;
    char filePath[1000];
    int values[100];
    int len = 0;

    if (dir)
    {
        while ((entry = readdir(dir)) != NULL)
        {
            sprintf(filePath, "%s/%s", root, entry->d_name);
            if (entry->d_type == DT_DIR)
            {

```

```

        if (strcmp(entry->d_name, ".") != 0 &&
strcmp(entry->d_name, "..") != 0)
            values[len++] = calc(filePath);
    }
    else
    {
        file = fopen(filePath, "r");

        if (strcmp(root + strlen(root) - 3, "add") == 0)
            values[len++] = add(file);
        else
            values[len++] = mul(file);

        fclose(file);
    }
}

int res = values[0];

if (strcmp(root + strlen(root) - 3, "add") == 0)
    res = addValues(values, len);
else
    res = mulValues(values, len);

return res;
}

int main()
{
    char *dirPath = "./tmp";
    FILE *result = fopen("result.txt", "w+");
    int n = calc(dirPath);
    fprintf(result, "%d", n);
    fclose(result);
    return 0;
}

```