

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. 1304

Новицкий М. Д.

Преподаватель

Чайка К. В.

Санкт-Петербург

2022

Цель работы.

Изучить рекурсивный способ обхода дерева.

Задание.

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.

Calculating - возвращающая значение long long int и принимающая на вход две переменные path – путь до директории или файла, обрабатываемый функцией, и last_dir, предыдущая директория, типа 3 char *. Внутри функции создаются две переменные типа long long int – res, которой присваивается значение 0, и tmp. Создается указатель dir_cp на тип char, который будет хранить в себе абсолютный путь до директории или файла. Затем указателю dir на DIR было присвоено значение функции opendir, в которую был передан path. Затем с помощью трех условных операторов проводится проверка на предыдущую папку.

Если предыдущей папкой была add, то переменной res присваивается значение 0 и объявляется указатель cur на структуру dirent, принимающий значение

функции `readdir`, которой на вход был подан указатель `dir`. Затем с помощью цикла `while` проведены проверки на имя считанного файла или директории.

- Если имя директории `add`, то указателю `dir_cp` передается адрес памяти, выделенной динамически в куче, с помощью функции `calloc`, которой на вход были поданы сумма длинны строки `path` и длинны строки `"/add"` и размер типа `char`. С помощью двух функций `strcpy` в `dir_cp` скопированы значения строк `path` и `"/add"`. К значению переменной `res` прибавляется значение функции `makeOperations`, которой на вход были поданы `dir_cp` и строка `"/add"`. После память, выделенная для `dir_cp`, очищается.

- Если имя директории `mul`, то указателю `dir_cp` передается адрес памяти, выделенной динамически в куче, с помощью функции `calloc`, которой на вход были поданы сумма длинны строки `path` и длинны строки `"/mul"` и размер типа `char`. С помощью двух функций `strcpy` в `dir_cp` скопированы значения строк `path` и `"/mul"`. К значению переменной `res` прибавляется значение функции `Calculating`, которой на вход были поданы `dir_cp` и строка `"/mul"`. После память, выделенная для `dir_cp`, очищается.

- Если в имени считанного элемента была подстрока `".txt"`, то указателю `dir_cp` передается адрес памяти, выделенной динамически в куче, с помощью функции `calloc`, которой на вход 4 были поданы сумма длины строки `path`, длины строки, хранящей в себе название файла, и числа 2 и размер типа `char`. С помощью двух функций `strcpy` в `dir_cp` скопированы значения строк `path` и `"/"`, а с помощью `strcat` совершена конкатенация строк `dir_cp` и название файла, который храниться в `cur->d_name`. С помощью функции `fopen` на чтение открывается файл `dir_cp` и указатель на тип `FILE` записывается в `file`. С помощью цикла `while` и функции `fscanf` считываются числа в файле и записываются в переменную `tmp`, которая прибавляется к значению переменной `res`. После окончания цикла файл закрывается, а память `dir_cp` очищается.

В переменную `cur` передается следующий считанный элемент. Если предыдущей папкой была `mul`, то переменной `res` присваивается значение 1 и объявляется указатель `cur` на структуру `dirent`, принимающий значение функции `readdir`, которой на вход был подан указатель `dir`. Затем с помощью цикла `while` проведены проверки на имя считанного файла или директории.

- Если имя директории `add`, то указателю `dir_cp` передается адрес памяти, выделенной динамически в куче, с помощью функции `calloc`, которой на вход были поданы сумма длинны строки `path` и длинны строки `"/add"` и размер типа `char`. С помощью двух функций `strcpy` в `dir_cp` скопированы значения строк `path` и `"/add"`. Значение переменной `res` умножается на значение функции `Calculating`, которой на вход были поданы `dir_cp` и строка `"/add"`. После память, выделенная для `dir_cp`, очищается.

- Если имя директории `mul`, то указателю `dir_cp` передается адрес памяти, выделенной динамически в куче, с помощью функции `calloc`, которой на вход

были поданы сумма длинны строки path и длинны строки “/mul” и размер типа char. С помощью двух функций strcpy в dir_cp скопированы значения строк path и “/mul”. Значение переменной res умножается на значение функции 5 Calculating, которой на вход были поданы dir_cp и строка “/add”. После память, выделенная для dir_cp, очищается.

- Если в имени считанного элемента была подстрока “.txt”, то указателю dir_cp передается адрес памяти, выделенной динамически в куче, с помощью функции calloc, которой на вход были поданы сумма длины строки path, длины строки, хранящей в себе название файла, и числа 2 и размер типа char. С помощью двух функций strcpy в dir_cp скопированы значения строк path и “/”, а с помощью strcat совершена конкатенация строк dir_cp и название файла, который храниться в sig->d_name. С помощью функции fopen на чтение открывается файл dir_cp и указатель на тип FILE записывается в file. С помощью цикла while и функции fscanf считываются числа в файле и записываются в переменную tmp. А значение res умножается на tmp и записывается в res. После окончания цикла файл закрывается, а память dir_cp очищается.

В переменную sig передается следующий считанный элемент. Если предыдущей папкой была tmp, то переменной res присваивается значение 0 и объявляется указатель sig на структуру dirent, принимающий значение функции readdir, которой на вход был подан указатель dir. Затем с помощью цикла while проведены проверки на имя считанного файла или директории.

- Если имя директории add, то указателю dir_cp передается адрес памяти, выделенной динамически в куче, с помощью функции calloc, которой на вход были поданы сумма длинны строки path и длинны строки “/add” и размер типа char. С помощью двух функций strcpy в dir_cp скопированы значения строк path и “/add”. К значению переменной res прибавляется значение функции Calculating, которой на вход были поданы dir_cp и строка “/add”. После память, выделенная для dir_cp, очищается.

- Если имя директории mul, то указателю dir_cp передается адрес памяти, выделенной динамически в куче, с помощью функции calloc, которой на вход были поданы сумма длинны строки path и длинны строки “/mul” и размер типа char. С помощью двух функций strcpy в dir_cp скопированы значения строк path и “/mul”. К значению переменной res прибавляется значение функции Calculating, которой на вход были поданы dir_cp и строка “/mul”. После память, выделенная для dir_cp, очищается.

В переменную sig передается следующий считанный элемент. Функция Calculating возвращает значение res. В функции main выделена пмять для указателя path на тип char для строки “tmp”. В переменную res типа long long int записан 0. В указатель resultf на FILE записывается адрес файла result.txt, который открыт на чтение. В переменную path копируется строка tmp с помощью функции strcpy. Переменной res присвоено значение функции

Calculating, которой на вход поданы две строки path и path. Затем в файл resultf записывается значение res и файл закрывается.

Результаты тестирования представлены в табл. 1. Таблица

1 – Результаты тестирования

№ теста	Вход	Выход	Комментарий
1	tmp/add/add/file1.txt - 1 tmp/add/add/file2.txt - 1 tmp/add/mul/file3.txt - 10 tmp/add/mul/file4.txt - 10 tmp/add/mul/add/file5.txt - 1 tmp/add/mul/add/file6.txt - 1	/result.txt - 202	Получен верный ответ

Выводы.

Изучен принцип работы с файлами и директориями с помощью рекурсивного обхода дерева в глубину. Написана программа, которая перебирает все файлы и считывает значения складывая или перемножая их в зависимости от названия директории, а затем записывает конечное значение в файл.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: solution.c

```
#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
#include <sys/types.h>
#include <string.h>

long long int Calculating(char *path, char *last_dir){
    printf("%s\n", path);
    long long int tmp, res = 0;
    char *dir_cp;
    DIR *dir = opendir(path);
    if(!strcmp(last_dir, "add")){
        res = 0;
        struct dirent *cur = readdir(dir);
        while(cur){
            if(strcmp(cur->d_name, "add") == 0){
                dir_cp = (char *)calloc(strlen(path) + 5,
sizeof(char));
                strcpy(dir_cp, path);
                strcat(dir_cp, "/add");
                res += Calculating(dir_cp, "add");
                free(dir_cp);
            }
            else if(strcmp(cur->d_name, "mul") == 0){
                dir_cp = (char *)calloc(strlen(path) + 5,
sizeof(char));
                strcpy(dir_cp, path);
                strcat(dir_cp, "/mul");
                res += Calculating(dir_cp, "mul");
                free(dir_cp);
            }
            else if(strstr(cur->d_name, ".txt") != NULL){
                dir_cp = (char *)calloc(strlen(path) + strlen(cur-
>d_name) + 2, sizeof(char));
                strcpy(dir_cp, path);
                strcat(dir_cp, "/");
                strcat(dir_cp, cur->d_name);
                FILE *file = fopen(dir_cp, "r");
                while(fscanf(file, "%lld", &tmp) != EOF){
                    res += tmp;
                }
                fclose(file);
                free(dir_cp);
            }
            cur = readdir(dir);
        }
    }
    else if(!strcmp(last_dir, "mul")){
        res = 1;
        struct dirent *cur = readdir(dir);
```

```

        while(cur){
            if(strcmp(cur->d_name, "add") == 0){
                dir_cp = (char *)calloc(strlen(path) + 5,
sizeof(char));
                strcpy(dir_cp, path);
                strcat(dir_cp, "/add");
                res += Calculating(dir_cp, "add");
                free(dir_cp);
            }
            else if(strcmp(cur->d_name, "mul") == 0){
                dir_cp = (char *)calloc(strlen(path) + 5,
sizeof(char));
                strcpy(dir_cp, path);
                strcat(dir_cp, "/mul");
                res += Calculating(dir_cp, "mul");
                free(dir_cp);
            }
            else if(strstr(cur->d_name, ".txt") != NULL){
                dir_cp = (char *)calloc(strlen(path) + strlen(cur-
>d_name) + 2, sizeof(char));
                strcpy(dir_cp, path);
                strcat(dir_cp, "/");
                strcat(dir_cp, cur->d_name);
                FILE *file = fopen(dir_cp, "r");
                while(fscanf(file, "%lld", &tmp) != EOF){
                    res += tmp;
                }
                fclose(file);
                free(dir_cp);
            }
            cur = readdir(dir);
        }
    }
    else if(!strcmp(last_dir, "tmp")){
        res = 0;
        struct dirent *cur = readdir(dir);
        while(cur){
            if(strcmp(cur->d_name, "add") == 0){
                dir_cp = (char *)calloc(strlen(path) + 5,
sizeof(char));
                strcpy(dir_cp, path);
                strcat(dir_cp, "/add");
                res += Calculating(dir_cp, "add");
                free(dir_cp);
            }
            else if(strcmp(cur->d_name, "mul") == 0){
                dir_cp = (char *)calloc(strlen(path) + 5,
sizeof(char));
                strcpy(dir_cp, path);
                strcat(dir_cp, "/mul");
                res += Calculating(dir_cp, "mul");
                free(dir_cp);
            }
            cur = readdir(dir);
        }
    }
    return res;
}

```

```
int main(int argc, char **argv){
    char *path = (char *)calloc(4, sizeof(char));
    long long int res = 0;
    FILE *resultf = fopen("result.txt", "w");
    strcpy(path, "tmp");
    res = Calculating(path, path);
    fprintf(resultf, "%lld", res);
    fclose(resultf);
    return 0;
}
```