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: Строки. Рекурсия, циклы, обход дерева

Студент гр. 3344 _____ Коршунов П.И.

Преподаватель _____ Глазунов С.А.

Санкт-Петербург

2024

Цель работы

Освоение работы с рекурсией на языке Си на примере использующей ее программы.

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

Были подключены *<stdio.h>*, *<stdlib.h>*, *<string.h>* и *<dirent.h>* для работы с файлами и директориями. Была создана рекурсивная функция *int listdir(const char *name, char *last_dir)* для обхода дерева файлов и подсчета значений в соответствии с условием *int c*;. На вход функция принимает путь к текущей и предыдущей директории. В функции создаются специальные структуры *DIR *dir; struct dirent *entry;* для работы с директориями. Также создается переменная счетчик, для подсчета значений в текущей директории и происходит проверка на успешное открытие директории. Далее запускается цикл *while ((entry = readdir(dir)) != NULL)* для прохода по всем файлам в текущей директории. Если текущий файл является директорией, то мы формируем к ней путь и вызываем нашу функцию рекурсивно, возвращаемое значение сохраняем. Также происходит исключение путей “.” и “..”, чтобы исключить бесконечную рекурсию. Возвращаемое значение прибавляется или умножается на переменную счетчик, чтобы просчитать всю директорию. Если текущий файл является файлом, то формируется путь к нему, открывается поток к этому файлу и вызывается функция *int count = counter(fp, last_dir);* для подсчета значений в файле с учетом директории, которой он находится. После поток закрывается. Возвращаемое значение прибавляется или умножается на переменную счетчик, чтобы просчитать всю директорию. После цикла закрывается поток на директорию и возвращается переменная счетчик. В функции для подсчета данных внутри файла происходит сканирование подсчет данных с помощью *while(fscanf(fp, "%d ", &c) == 1) overall += c;*. В *main* запускается функция для просчета всех подкаталогов и происходит запись результата в текстовый файл с помощью *FILE* fp = fopen("result.txt", "w"); fprintf(fp, "%d", overall); fclose(fp);*

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	file.txt: 1 file1.txt: 1 file2.txt: 2 2 file3.txt: 7 file4.txt: 1 2 3 file5.txt: 3 -1 root/add/add/file.txt root/add/add/file1.txt root/add/mul/file2.txt root/add/mul/file3.txt root/add/mul/add/file4.txt root/add/mul/add/file5.txt	236 result.txt	-

Выводы

Была освоена работа с рекурсивными функциями на языке Си на примере использующей их программы.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Korshunov_Petr_lb3.c

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

int counter(FILE *fp, char* operation){
    if(fp == NULL) {
        return -1;
    }
    if(strcmp(operation, "add")==0){

        int overall = 0;
        int c;
        while(fscanf(fp, "%d ", &c) == 1)
            overall += c;
        return overall;
    }else{
        int overall = 1;
        int c;
        while(fscanf(fp, "%d ", &c) == 1)
            overall *= c;
        return overall;
    }
}


int listdir(const char *name, char *last_dir)
{
    DIR *dir;
    struct dirent *entry;

    if (!(dir = opendir(name)))
        exit(-1);

    int c;
    if(strcmp(last_dir, "add")==0){
        c = 0;
    }else if (strcmp(last_dir, "mul")==0)
    {
        c = 1;
    }
    while ((entry = readdir(dir)) != NULL) {
        if (entry->d_type == DT_DIR) {
            if (strcmp(entry->d_name, ".") == 0 || strcmp(entry->d_name,
"..") == 0)
                continue;
            char* path = malloc(strlen(name) + strlen(entry->d_name) + 2);
            snprintf(path,  strlen(name)  +  strlen(entry->d_name)  +  2,
"%s/%s", name, entry->d_name);
            int count = listdir(path, entry->d_name);
```

```
            if(strcmp(last_dir, "add")==0){
                c += count;
            }else if (strcmp(last_dir, "mul")==0)
            {
                c *= count;
            }
            else{
                closedir(dir);
                return count;
            }

        } else {
            char* temp = malloc(strlen(name) + strlen(entry->d_name) + 2);
            snprintf(temp,  strlen(name)  +  strlen(entry->d_name)  +  2,
"%s/%s", name, entry->d_name);
            FILE* fp = fopen(temp, "r");
            int count = counter(fp, last_dir);
            fclose(fp);
            if(strcmp(last_dir, "add")==0){
                c += count;
            }else if (strcmp(last_dir, "mul")==0)
            {
                c *= count;
            }
        }
    }
    closedir(dir);
    return c;
}

int main(void) {
    int overall = 0;
    overall = listdir("./tmp", ".");
    FILE* fp = fopen("result.txt", "w");
    if(fp == NULL) {
        return 1;
    }
    fprintf(fp, "%d", overall);
    fclose(fp);
    return 0;
}
```