**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе №3**

**по дисциплине «Программирование»**

Тема: [**Строки. Рекурсия, циклы, обход дерева**](https://stepik.org/course/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5-(%D0%9B%D0%AD%D0%A2%D0%98)-1096/syllabus?module=12)

|  |  |  |
| --- | --- | --- |
| Студент гр. 6303 |  | Матюшина М.Е. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

Цель работы: 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

Иехархия папок и файлов задана 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

**Ход работы.**

1. **Рекурсивная функция int dirTraveler(const char \*startDir, int flag);**

Принимает в качестве ***аргуменов*** строку, содержащую полный путь из исходной директории, и флаг, который сигнализирует о том, из какого рода директории: add (flag =0) или mul (flag =1) вызывается функия.

***Возвращаемое значение***: целое число, результат операций в директории.

1. **Вспомогательные функии.**

* **int takeAdd(const char \*path)**

Функция принимает в качестве ***аргумента*** строку, содержащую полный путь из исходной директории до текстого файла, открывает этот файл и считывает его данные в строку, далее с помощью функии strtok, исходная строка разбивается на лексемы, а функция atoi интерпретерует их как целые числа. Файл закрывается.

***Возвращаемое значение***: целое число, сумма чисел, записанных в файле.

int takeAdd(const char \*path) {

char\* s=malloc(100\*sizeof(char));

FILE \*f = fopen(path, "r");

int count;

char\* part;

count=0;

if(f) {

fgets(s,sizeof(s)/sizeof(char),f) ;

part=strtok(s, " ");

while (part!=NULL){

count=count+atoi(part);

part = strtok(NULL, " ");

}

fclose(f);

}

free(s);

return count;

}

* **int takeMul(const char \*path)**

Функция аналогична функции **takeAdd.**

***Возвращаемое значение***: целое число, произведение чисел, записанных в файле.

* **void remake( char \*path, char \* new) {**

Функция дописывает путь из исходной директории, добавляя в изначальный путь **path** новую папку  **/new**

void remake( char \*path, char \* new) {

strcat(path, "/");

strcat(path, new);

}

1. **Функция main()**

В функции main происходят открытие самой первой папки mul/add и первый вызов рекурсивной функции **dirTraveler.**

Результат выводится на печать.

1. **Полный код программы.**

#include <stdio.h>

#include <string.h>

#include <sys/types.h>

#include <dirent.h>

#include <stdlib.h>

int takeAdd(const char \*path) {

char\* s = malloc(100\*sizeof(char));

FILE \*f = fopen(path, "r");

int count;

char\* part;

count=0;

if(f) {

fgets(s,sizeof(s)/sizeof(char),f) ;

part=strtok(s, " ");

while (part!=NULL){

count=count+atoi(part);

part = strtok(NULL, " ");

}

fclose(f);

}

free(s);

return count;

}

int takeMul(const char \*path) {

char\* s=malloc(100\*sizeof(char));

FILE \*f = fopen(path, "r");

int count;

char\* part;

count=1;

if(f) {

fgets(s,sizeof(s)/sizeof(char),f);

part=strtok(s, " ");

while (part!=NULL){

count=count\*atoi(part);

part = strtok(NULL, " ");

}

fclose(f);

}

free(s);

return count;

}

void remake( char \*path, char \* new) {

strcat(path, "/");

strcat(path, new);

}

int dirTraveler(const char \*startDir, int flag) {

char\* path=malloc(1000\*sizeof(char));

strcpy(path, startDir);

int path\_len, path\_len1;

int result;

if (flag==0){

result=0;

remake(path, "add");

}

if (flag==1){

result=1;

remake(path, "mul");

}

DIR \*dir=opendir(path);

struct dirent \*de;

while(de = readdir(dir)) {

if(de->d\_type == DT\_REG && strstr(de->d\_name,".txt")) {

path\_len=strlen(path);

remake(path, de->d\_name);

if (flag==0){

result=result+takeAdd(path);

}

if (flag==1){

result=result\*takeMul(path);

}

path[path\_len]='\0';

}

if(de->d\_type == DT\_DIR && strcmp("add", de->d\_name)==0 ) {

if (flag==0) {

result=result+dirTraveler(path, 0);

}

else {

result=result\*dirTraveler(path, 0);

}

}

if(de->d\_type == DT\_DIR && strcmp("mul", de->d\_name)==0 ) {

if (flag==0) {

result=result+dirTraveler(path, 1);

}

else {

result=result\*dirTraveler(path, 1);

}

}

}

closedir(dir);

free(path);

return result;

}

int main() {

char\* path=malloc(1000\*sizeof(char));

strcpy(path, ".");

DIR \*dir=opendir(path);

struct dirent \*de;

while (de = readdir(dir)) {

if (strcmp("add", de->d\_name)==0)

printf("%d\n", dirTraveler(path,0));

if (strcmp("mul", de->d\_name)==0)

printf("%d\n", dirTraveler(path,1));

}

closedir(dir);

free(path);

return 0;

}

1. **Загрузка файла на GitHub.**

* В локальном репозитории была создана новая ветка matyushina\_2sem\_lr3

git checkout -b matyushina\_2sem\_lr3

* Создана директория matyushina\_2sem\_lr3

mkdir matyushina\_2sem\_lr3

* В вышеуказанную папку был перемещен файл LAB3.c
* Исходный файл добавлен в локальный репозиторий

git add LAB3.c

* Командой <git commit –m> были зафиксированы изменения в проекте

git commit –m «matyushina\_2sem\_lr3»

* Командой <git push> последний commit был отправлен на удаленный репозиторий

git push origin matyushina\_2sem\_lr3

**Вывод:** в ходе работы была написана программа, которая, запускается в корневой директории, содержащей одну папку с именем "add" или "mul" и выполняет действия над файлами и папками, содержащимися в ней. Код и отчет по работе были загружены в репозиторий.