**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: Рекурсия, циклы, обход дерева

|  |  |  |
| --- | --- | --- |
| Студент гр. 6303 |  | Амбарцумян А.В. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

**Цель работы:**

**Задание**:

Задана иехархия 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 дирректории, 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

**Ход работы:**

1. Подключены необходимые для работы заголовочные файлы

 #include <stdio.h>

 #include <stdlib.h>

 #include <string.h>

 #include <dirent.h>

 #include <sys/types.h>

1. В функции main выделяется память для хранения пути и вызывается функция dir\_res для обхода директорий

 char \*current\_path=(char \*)[malloc](http://www.opengroup.org/onlinepubs/009695399/functions/malloc.html)(200);

 [strcpy](http://www.opengroup.org/onlinepubs/009695399/functions/strcpy.html)(current\_path,".");

 dir\_res(current\_path, "smth");

 [free](http://www.opengroup.org/onlinepubs/009695399/functions/free.html)(current\_path);

 return 0;

1. Создана функция dir\_res, выполняющая рекурсивный обход директорий. Если в директория обнаруживается файл с расширением “.txt”, то вызывается функция count., вычисляющая результат в соответствии с названием директории. Если найдена директория с названием add или mul, то рекурсивно вызывается функция dir\_res.

 if (de->d\_type == DT\_REG && [strstr](http://www.opengroup.org/onlinepubs/009695399/functions/strstr.html)(de->d\_name, ".txt") )

 {

 int len = [strlen](http://www.opengroup.org/onlinepubs/009695399/functions/strlen.html)(current\_path);

 [strcat](http://www.opengroup.org/onlinepubs/009695399/functions/strcat.html)([strcat](http://www.opengroup.org/onlinepubs/009695399/functions/strcat.html)(current\_path, "/"), de->d\_name);

 int result = count(current\_path, operation);

 if ([strcmp](http://www.opengroup.org/onlinepubs/009695399/functions/strcmp.html)(operation, "add")) file\_result\*=result;

 else file\_result+=result;

 if (de->d\_type == DT\_DIR && [strcmp](http://www.opengroup.org/onlinepubs/009695399/functions/strcmp.html)(de->d\_name, ".")!=0 && [strcmp](http://www.opengroup.org/onlinepubs/009695399/functions/strcmp.html)(de->d\_name, "..")!=0 )

 {

 int len = [strlen](http://www.opengroup.org/onlinepubs/009695399/functions/strlen.html)(current\_path);

 [strcat](http://www.opengroup.org/onlinepubs/009695399/functions/strcat.html)([strcat](http://www.opengroup.org/onlinepubs/009695399/functions/strcat.html)(current\_path, "/"), de->d\_name);

 if (![strcmp](http://www.opengroup.org/onlinepubs/009695399/functions/strcmp.html)(de->d\_name, "mul")) file\_result+=dir\_res(current\_path, de->d\_name);

 else file\_result\*=dir\_res(current\_path, de->d\_name);

 current\_path[len] = '\0';

1. В функции count открывается файл по пути, переданному из dir\_res. Выполняется проверка открытия файла. В случае, если файл открыть не удалось, выводится сообщение об ошибке и функция возвращает 0. В обратном случае, выполняются вычисления в соответствии с переданным из dir\_res названием. Результат возвращается в dir\_res.

 FILE \*f = [fopen](http://www.opengroup.org/onlinepubs/009695399/functions/fopen.html)(current\_path, "r");

 if (f == NULL)

 {

 [printf](http://www.opengroup.org/onlinepubs/009695399/functions/printf.html)("File haven`t open");

 return 0;

 }

 if (![strcmp](http://www.opengroup.org/onlinepubs/009695399/functions/strcmp.html)(operation, "add"))

 {

 result = 0;

 while(token)

 {

 result+=[atoi](http://www.opengroup.org/onlinepubs/009695399/functions/atoi.html)(token);

 token = [strtok](http://www.opengroup.org/onlinepubs/009695399/functions/strtok.html)(NULL, " \n");

 }

 }

Затем файл lab3.c с кодом написанной программы и отчёт о проведенной работе был загружен на github c помощью консоли:

* Создана новая ветка Ambartsumyan\_lr\_3

git checkout –b Ambartsumyan\_lr\_3

* Создана новая директория Ambartsumyan\_lr\_2 и файл lab2.c в ней, куда помещен код написанной программы

mkdir Ambartsumyan\_lr\_3

cd Ambartsumyan\_lr\_3

nano lab3.c

* Созданные файлы добавлены для загрузки

git add Ambartsumyan\_lr\_3

* Добавлен комментарий

git commit –m “3 Lab”

* Файлы загружены на репозиторий Git

git push origin Ambartsumyan\_lr\_3

**Вывод:** в ходе работы создана функция для рекурсивного обхода директорий и поиска файлов и вычисления. В работе использованы циклы, рекурсия, а также динамическая память.

**Исходный код**

1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <string.h>
4. #include <dirent.h>
5. #include <sys/types.h>
7. int dir\_res(char \*, char \*);
8. int count(char \*, char \*);
10. int main()
11. {
12. char \*current\_path=(char \*)malloc(200);
13. strcpy(current\_path,".");
14. dir\_res(current\_path, "smth");
15. free(current\_path);
16. return 0;
17. }
19. int dir\_res(char \*current\_path, char \*operation)
20. {
21. DIR \*dir = opendir(current\_path);
22. struct dirent \*de = readdir(dir);
23. int file\_result = 0;
24. int final = 0;
25. if (strcmp(operation, "mul"))
26. file\_result = 0;
27. else
28. file\_result = 1;
29. if (dir)
30. while (de)
31. {
32. if (de->d\_type == DT\_REG && strstr(de->d\_name, ".txt") )
33. {
34. int len = strlen(current\_path);
35. strcat(strcat(current\_path, "/"), de->d\_name);
36. int result = count(current\_path, operation);
37. if (strcmp(operation, "add")) file\_result\*=result;
38. else file\_result+=result;
39. final = file\_result;
40. current\_path[len] = '**\0**';
41. }
42. if (de->d\_type == DT\_DIR && strcmp(de->d\_name, ".")!=0 && strcmp(de->d\_name, "..")!=0 )
43. {
44. int len = strlen(current\_path);
45. strcat(strcat(current\_path, "/"), de->d\_name);
46. if (!strcmp(de->d\_name, "mul")) file\_result+=dir\_res(current\_path, de->d\_name);
47. else file\_result\*=dir\_res(current\_path, de->d\_name);
48. current\_path[len] = '**\0**';
49. final = file\_result;
50. }
51. de = readdir(dir);
52. }
53. closedir(dir);
54. if (file\_result != 0)
55. {
56. final = file\_result;
57. printf("result %d**\n**", final);
59. }
60. return final;
61. }
63. int count(char \*current\_path, char \*operation)
64. {
65. FILE \*f = fopen(current\_path, "r");
66. if (f == NULL)
67. {
68. printf("File haven`t open")
69. return 0;
70. }
71. char str[101];
72. fgets(str, 100, f);
73. char \*token = strtok(str, " **\n**");
74. int result;
75. int a;
76. if (!strcmp(operation, "add"))
77. {
78. result = 0;
79. while(token)
80. {
81. result+=atoi(token);
82. token = strtok(NULL, " **\n**");
83. }
84. }
85. else
86. {
87. if (!strcmp(operation, "mul"))
88. {
89. result = 1;
90. while(token)
91. {
92. result\*=atoi(token);
93. token = strtok(NULL, " **\n**");
94. }
95. }
96. }
97. fclose(f);
98. return result;
99. }