



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет

*к лабораторной работе №2
«Дерево каталогов»*

Студент: Батбилэг Н.

Группа: ИУ7-71Б

Оценка (баллы) _____

Преподаватель: Рязанова Н.Ю.

Москва.
2021 г.

1. Задание

Написать программу, которая образует деревья каталогов через рекурсию и со стеком.

2. Реализация задания

В листинге 1 представлен код задания.

Листинг 1. prog.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <unistd.h>
5 #include <dirent.h>
6 #include <sys/stat.h>
7
8 #define FTW_F 1 //файл, не являющийся каталогом
9 #define FTW_D 2 //каталог
10 #define FTW_DNR 3 //каталог, недоступный для чтения
11 #define FTW_NS 4 //файл, информацию о котором нельзя получить с помощью stat
12
13 // тип функции, которая будет вызываться для каждого встреченного файла
14 typedef int MyFunc(const char * ,const struct stat * , int);
15
16 static MyFunc counter;
17 static int myftw(char * , MyFunc * );
18 static int dopath(const char* filename, int depth, MyFunc * );
19
20 static long nreg, ndir, nblk, nchr, nfifo, nlink, nsock, ntot;
21
22 int main(int argc, char * argv[]) {
23     int ret = -1;
24     if (argc != 2) {
25         fprintf(stderr, "Пример запуска: ./lab.out <каталог>\n");
26         return EXIT_FAILURE;
27     }
28
29     ret = myftw(argv[1], counter);
30
31     ntot = nreg + ndir + nblk + nchr + nfifo + nlink + nsock;
32
33     if (ntot == 0)
34         ntot = 1; //во избежание деления на 0; вывести 0 для всех счетчиков
35
36     printf("\нобычные файлы:\t%7ld, %5.2f %%\n", nreg, nreg*100.0/ntot);
37     printf("каталоги:\t%7ld, %5.2f %%\n", ndir, ndir*100.0/ntot);
38     printf("специальные файлы блочных устройств:\t%7ld, %5.2f %%\n", nblk,
39         nblk*100.0/ntot);
40     printf("специальные файлы символьных устройств:\t%7ld, %5.2f %%\n", nchr,
41         nchr*100.0/ntot);
42     printf("FIFO\t%7ld, %5.2f %%\n", nfifo, nfifo*100.0/ntot);
43     printf("символьные ссылки:\t%7ld, %5.2f %%\n", nlink, nlink*100.0/ntot);
44     printf("сокеты:\t%7ld, %5.2f %%\n\n", nsock, nsock*100.0/ntot);
45
46     return ret;
47 }
48
49 // Обходит дерево каталогов, начиная с pathname и применяя к каждому файлу func
50 static int myftw(char * pathname, MyFunc * func) {
51     return(dopath(pathname, 0, func));
52 }
```

```

53
54 static int dopath(const char * filename, int depth, MyFunc * func) {
55     struct stat statbuf;
56     struct dirent * dirp;
57     DIR * dp;
58     int ret = 0;
59
60     if (lstat(filename, &statbuf) == -1) // Ошибка вызова функции lstat
61         return(func(filename, &statbuf, FTW_NS));
62
63     for (int i = 0; i < depth; ++i) //
64         printf("|\\t");
65
66     if (S_ISDIR(statbuf.st_mode) == 0) // Файл не является каталогом
67         return(func(filename, &statbuf, FTW_F)); // Отобразить в дереве
68
69     if ((ret = func(filename, &statbuf, FTW_D)) != 0) // Каталог недоступен
70         return(ret);
71
72     if ((dp = opendir(filename)) == NULL) // не можем открыть дерикторию
73         return(func(filename, &statbuf, FTW_DNR));
74
75     chdir(filename);
76     while ((dirp = readdir(dp)) != NULL && ret == 0)
77     {
78         //Пропустить каталоги . И .. если попали в текущий или родительский каталог
79         if (strcmp(dirp->d_name, ".") != 0 && strcmp(dirp->d_name, "..") != 0)
80         {
81             ret = dopath(dirp->d_name, depth + 1, func); // рекурсии
82         }
83     }
84
85     chdir(".."); //
86
87     if (closedir(dp) < 0)
88         perror("Невозможно закрыть каталог");
89     return(ret);
90 }
91
92 static int counter(const char* filename, const struct stat * statptr, int type)
93 {
94     switch(type)
95     {
96         case FTW_F:
97             printf( "|--- %s\\n", filename);
98             switch(statptr->st_mode & S_IFMT)
99             {
100                 case S_IFREG: nreg++; break;
101                 case S_IFBLK: nblk++; break;
102                 case S_IFCHR: nchr++; break;
103                 case S_IFIFO: nfifo++; break;
104                 case S_IFLNK: nlink++; break;
105                 case S_IFSOCK: nsock++; break;
106                 case S_IFDIR:
107                     fprintf(stderr, "Каталог имеет тип FTW_F");
108                     return EXIT_FAILURE;
109             }
110             break;
111         case FTW_D:
112             printf( "|--- %s\\n", filename);

```

```

113         ndir++; break;
114     case FTW_DNR:
115         fprintf(stderr, "Закрыт доступ к одному из каталогов!");
116         return EXIT_FAILURE;
117     case FTW_NS:
118         fprintf(stderr, "Ошибка функции stat!");
119         return EXIT_FAILURE;
120     default:
121         fprintf(stderr, "Неизвестный тип файла!");
122         return EXIT_FAILURE;
123 }
124 return(0);
125

```

На рисунках 1-2 представлено дерево каталогов, начиная с директории /Documents/BMSTU/lab_02

```

dalai@dalai-Inspiron-5567:~/Documents/BMSTU/lab_02$ ./a.out .
--- ./
|--- a.out
|--- Отчет_лабораторная_2.odt
|--- prog.c
|--- Web-development-master/
|   |--- myphp/
|   |   |--- if.php
|   |   |--- index.php
|   |   |--- for.php
|   |   |--- arrays.php
|   |   |--- get.php
|   |   |--- post.php
|   |   |--- while.php
|   |--- CSS/
|   |   |--- exercise_position.css
|   |   |--- web_2.css
|   |   |--- font_family_size_style-text.html
|   |   |--- exercise_1.html
|   |   |--- exercise_2.html
|   |   |--- style_URL.html
|   |   |--- tasks/
|   |       |--- Screenshot_3.jpg
|   |       |--- Screenshot_5.jpg
|   |       |--- Screenshot_6.jpg
|   |       |--- Screenshot_4.jpg
|   |       |--- Screenshot_7.jpg
|   |       |--- Screenshot_1.jpg
|   |       |--- Screenshot_2.jpg
|   |   |--- float.html
|   |   |--- exercise_4.css
|   |   |--- advanced_selectors.css
|   |   |--- exercise_4.html
|   |   |--- external_css.css
|   |   |--- id_class.html
|   |   |--- exercise_4_b.css
|   |   |--- exercise_3.html
|   |   |--- image/

```

Рисунок 1. Результат работы программы (начало).

```

|--- exercise_weight.js
|--- switch_operator.js
|--- for_loop.js
|--- object.js
|--- methods.js
|--- arryas_object.js
|--- ternary_operator.js
|--- methods_array.js
|--- exercise_DOM.js
|--- app.html
|--- function_statements_expressions.js
|--- DOM_selectors.js
|--- function.js
|--- Note.txt
|--- exercise_cars_object.js
|--- exercise_dom.html
|--- exercise_cv.html
|--- change_content.js
|--- primitive.html
|--- app.js
|--- change_content_style.js
|--- DOM_events.js
|--- main_javascript.html
|--- CurrencyExchange/
|       |--- exchange.html
|       |--- javascript/
|               |--- exchange.js
|--- Primitive data types.txt

```

```

обычные файлы:      159, 83.25 %
каталоги:           32, 16.75 %
специальные файлы блочных устройств:      0,  0.00 %
специальные файлы символьных устройств:    0,  0.00 %
FIFO                0,  0.00 %
символьные ссылки:      0,  0.00 %
сокеты:             0,  0.00 %

```

```

dalai@dalai-Inspiron-5567:~/Documents/BMSTU/lab_02$ █

```

Рисунок 2. Результат работы программы (конец)

На рисунке 3 представлен результат действия команды “ls -al” из этого же каталога.

```

dalai@dalai-Inspiron-5567:~/Documents/BMSTU/lab_02$ ls -al
total 748
drwxrwxr-x  3 dalai dalai  4096 9-p cap  30 14:39 .
drwxrwxr-x 11 dalai dalai  4096 9-p cap  30 14:38 ..
-rwxrwxr-x  1 dalai dalai 17536 9-p cap  30 11:32 a.out
-rw-rw-r--  1 dalai dalai  4577 9-p cap  30 11:31 prog.c
drwxrwxr-x 10 dalai dalai  4096 7-p cap   5 16:35 Web-development-master
-rw-rw-r--  1 dalai dalai 724142 9-p cap  30 01:33 Отчет_лабораторная_2.odt
dalai@dalai-Inspiron-5567:~/Documents/BMSTU/lab_02$ █

```

Рисунок 3. Команда “ls -al”.