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»

Дисциплина: «Операционные системы»

Лабораторная работа №2

Тема: «Дерево каталогов»

Студент: Амиян Э. Г.
Группа: ИУ7-64Б
Преподаватель: Рязанова Н.Ю.

2020 г.

Задание:

1. Структурировать исходный код программы в листинге 4.7.

2. Изменить программу так, чтобы она выводила на экран дерево каталогов.

3. Изменить функцию myftw() так, чтобы каждый раз, когда встречается каталог, функции lstat() передавался не полный путь к файлу, а только его имя. Для этого после обработки всех файлов в каталоге вызовите chdir("..").

Листинг 1: lab..c

```
1. #include <dirent.h>
2. #include <limits.h>
3. #include <stdio.h>
4. #include <stdlib.h>
5. #include <errno.h>
6. #include <unistd.h>
7. #include <string.h>
8. #include <sys/types.h>
9. #include <sys/stat.h>
10.
11.#define FTW_F 1 //файл, не являющийся каталогом
12.#define FTW_D 2 //каталог
13.#define FTW_DNR 3 //каталог, недоступный для чтения
14.#define FTW_NS 4 //файл, информацию о котором нельзя получить с помощью stat
15.
16.// тип функции, которая будет вызываться для каждого встреченного файла
17.typedef int MyFunc(const char * ,const struct stat *, int);
18.
19.static MyFunc counter;
20.static int myftw(char *, MyFunc * );
21.static int dopath(const char* filename, int depth, MyFunc * );
22.
23.static long nreg, ndir, nblk, nchr, nfifo, nslink, nsock, ntot;
24.
25.int main(int argc, char * argv[]) {
26.    int ret = -1;
27.    if (argc != 2) {
28.       fprintf(stderr, "Пример запуска: ./lab.out <каталог>\n");
29.       return EXIT_FAILURE;
30.    }
31.
32.    ret = myftw(argv[1], counter);
33.    ntot = nreg + ndir +  nblk + nchr +  nfifo + nslink + nsock;
34.
35.     if (ntot == 0)
36.             ntot = 1; //во избежание деления на 0; вывести 0 для всех
   счетчиков
```

```
37.
38.       printf("\nобычные файлы:                          %7ld, %5.2f %%\n",
39.      nreg, nreg*100.0/ntot);
40.       printf("каталоги:                                %7ld, %5.2f %%\n",
41.      ndir, ndir*100.0/ntot);
42.       printf("специальные файлы блочных устройств:     %7ld, %5.2f %%\n",
43.      nblk, nblk*100.0/ntot);
44.       printf("специальные файлы сисвольных устройств: %7ld, %5.2f %%\n",
45.      nchr, nchr*100.0/ntot);
46.       printf("FIFO:                                    %7ld, %5.2f %%\n",
47.      nfifo, nfifo*100.0/ntot);
48.       printf("символьные ссылки:                       %7ld, %5.2f %%\n",
49.      nslink, nslink*100.0/ntot);
50.       printf("сокеты:                                  %7ld, %5.2f %%\n\n",
51.      nsock, nsock*100.0/ntot);
52.
53.     return ret;
54. }
55.
56. // Обходит дерево каталогов, начиная с pathname и применяя к каждому файлу func
57. static int myftw(char * pathname, MyFunc * func) {
58.     return(dopath(pathname, 0, func));
59. }
60.
61. static int dopath(const char* filename, int depth, MyFunc * func)
62. {
63.     struct stat statbuf;
64.     struct dirent * dirp;
65.     DIR * dp;
66.     int ret = 0;
67.
68.     if (lstat(filename, &statbuf) == -1) // Ошибка вызова функции lstat
69.         return(func(filename, &statbuf, FTW_NS));
70.
71.     for (int i = 0; i < depth; ++i)
72.              printf("|\t");
73.
74.     if (S_ISDIR(statbuf.st_mode) == 0) // Не каталог
75.         return(func(filename, &statbuf, FTW_F)); // Отобразить в дереве
76.
77.     if ((ret = func(filename, &statbuf, FTW_D)) != 0)
78.         return(ret);
79.
80.     if ((dp = opendir(filename)) == NULL) // Каталог недоступен
81.         return(func(filename, &statbuf, FTW_DNR));
82.
83.     chdir(filename);
84.     while ((dirp = readdir(dp)) != NULL && ret == 0)
85.     {
86.         if (strcmp(dirp->d_name, ".") != 0 &&
```

```
87.             strcmp(dirp->d_name, "..") != 0 ) // Пропустить каталоги . и ..
88.         {
89.             ret = dopath(dirp->d_name, depth + 1, func);
90.         }
91.     }
92.
93.     chdir("..");
94.
95.     if (closedir(dp) < 0)
96.         perror("Невозможно закрыть каталог");
97.
98.     return(ret);
99. }
100.
101.      static int counter(const char* filename, const struct stat * statptr, int type)
102.      {
103.          switch(type)
104.          {
105.              case FTW_F:
106.                  printf( "|--- %s\n", filename);
107.                  switch(statptr->st_mode & S_IFMT)
108.                  {
109.                      case S_IFREG: nreg++; break;
110.                      case S_IFBLK: nblk++; break;
111.                      case S_IFCHR: nchr++; break;
112.                      case S_IFIFO: nfifo++; break;
113.                      case S_IFLNK: nslink++; break;
114.                      case S_IFSOCK: nsock++; break;
115.                      case S_IFDIR:
116.                          fprintf(stderr, "Каталог типа FTW_F");
117.                          return EXIT_FAILURE;
118.                  }
119.                  break;
120.              case FTW_D:
121.                  printf( "|--- %s/\n", filename);
122.                  ndir++; break;
123.              case FTW_DNR:
124.                  fprintf(stderr, "Закрыт доступ к одному из каталогов!");
125.                  return EXIT_FAILURE;
126.              case FTW_NS:
127.                  fprintf(stderr, "Ошибка функции stat!");
128.                  return EXIT_FAILURE;
129.              default:
130.                  fprintf(stderr, "Неизвестый тип файла!");
131.                  return EXIT_FAILURE;
132.          }
133.
134.          return(0);
135.      }
```

Демонстрация работы программы

На рисунках 1-2 отображена работа программы, можно наблюдать дерево каталогов, начиная с директории ***os_labs.***

```
amiyan_ed@Lenovo-IdeaPad-L340:~/os_labs$ ./lab2.out .
|--- ./
|       |--- lab.c
|       |--- lab_8/
|       |       |--- image
|       |       |--- myfs.mod
|       |       |--- myfs.ko
|       |       |--- Makefile
|       |       |--- dir/
|       |       |--- myfs.c
|       |--- lab2.out
|       |--- lab_6/
|       |       |--- part_2/
|       |       |       |--- server.c
|       |       |       |--- client.c
|       |       |--- Pic_png/
|       |       |       |--- 1_2.png
|       |       |       |--- 2_5.png
|       |       |       |--- 1_1.png
|       |       |       |--- 2_6.png
|       |       |--- part_1/
|       |       |       |--- server.c
|       |       |       |--- client.c
|       |--- lab_05/
|       |       |--- main.c
|       |       |--- Makefile
|       |--- lab_04/
|       |       |--- fortune.c
|       |       |--- proc.c
|       |       |--- fortune.ko
|       |       |--- Makefile
|       |       |--- fortune.mod
|       |--- lab_03/
|       |       |--- md1.mod
|       |       |--- md3.ko
|       |       |--- md3.mod
|       |       |--- md2.mod
|       |       |--- part_2/
|       |       |       |--- md1.mod
|       |       |       |--- md.h
|       |       |       |--- md2.c
|       |       |       |--- md3.ko
```

Рисунок 1: Дерево каталогов (начало)

```
|       |       |       |--- md2.ko
|       |       |       |--- md1.ko
|       |       |--- screens/
|       |       |       |--- 1_.png
|       |       |       |--- 2_1.png
|       |       |       |--- 2_3.png
|       |       |       |--- 1_1.png
|       |       |       |--- 2_2.png
|       |       |--- md2.ko
|       |       |--- md1.ko
|       |       |--- part_1/
|       |       |       |--- md.mod
|       |       |       |--- md.ko
|       |       |       |--- Makefile
|       |       |       |--- md.c
|       |--- lab_9/
|       |       |--- snapshots/
|       |       |       |--- 2_1.png
|       |       |       |--- 4_2.png
|       |       |       |--- 4.png
|       |       |       |--- 1.png
|       |       |       |--- 4_1.png
|       |       |--- tasklet/
|       |       |       |--- tasklet.mod
|       |       |       |--- tasklet.ko
|       |       |       |--- Makefile
|       |       |       |--- tasklet.c
|       |       |--- queue/
|       |       |       |--- Makefile
|       |       |       |--- queue.ko
|       |       |       |--- queue.mod
|       |       |       |--- queue.c
```

```
обычные файлы:                                  61, 78.21 %
каталоги:                                       17, 21.79 %
специальные файлы блочных устройств:             0,  0.00 %
специальные файлы сисвольных устройств:          0,  0.00 %
FIFO:                                            0,  0.00 %
символьные ссылки:                               0,  0.00 %
сокеты:                                          0,  0.00 %

amiyan_ed@Lenovo-IdeaPad-L340:~/os_labs$ ls -al
итого 56
drwxr-xr-x  8 amiyan_ed amiyan_ed  4096 июн  5 01:51 .
drwxr-xr-x 25 amiyan_ed amiyan_ed  4096 июн  4 19:23 ..
drwxr-xr-x  5 amiyan_ed amiyan_ed  4096 июн  1 16:52 lab_03
drwxr-xr-x  2 amiyan_ed amiyan_ed  4096 мар 28 17:29 lab_04
drwxr-xr-x  2 amiyan_ed amiyan_ed  4096 июн  5 01:48 lab_05
-rwxr-xr-x  1 amiyan_ed amiyan_ed 13232 июн  5 01:51 lab2.out
drwxr-xr-x  5 amiyan_ed amiyan_ed  4096 апр 28 04:27 lab_6
drwxr-xr-x  3 amiyan_ed amiyan_ed  4096 мая 19 02:01 lab_8
drwxr-xr-x  5 amiyan_ed amiyan_ed  4096 мая 24 03:11 lab_9
-rw-r--r--  1 amiyan_ed amiyan_ed  4724 июн  5 01:43 lab.c
amiyan_ed@Lenovo-IdeaPad-L340:~/os_labs$ man ls
```

Рисунок 2: Дерево каталогов (конец) и демонстрация содержимого каталога с помощью команды ***ls -al***