

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

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 2

Дисциплина Операционные системы.
Тема Файлы и каталоги

Студент Куприй А. А.
Группа ИУ7-63Б
Преподаватель Рязанова Н.Ю.

Москва, 2020 г.

ВВЕДЕНИЕ

Задание:

- Структурировать исходный код программы в листинге 4.7.
- Изменить программу так, чтобы она выводила на экран дерево каталогов.
- 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("..")`.

1 Практическая часть

1.1 Задание

Ниже представлен листинг программы.

Листинг 1.1 — Текст программы

```
1 #include <sys/stat.h>
2 #include <sys/types.h>
3 #include <dirent.h>
4 #include <limits.h>
5 #include <stdio.h>
6 #include <unistd.h>
7 #include <string.h>
8 #include <errno.h>
9 #include <stdarg.h>
10 #include <stdlib.h>
11
12 #define FTW_F 1
13 #define FTW_D 2
14 #define FTW_DNR 3
15 #define FTW_NS 4
16 #define MAXLINE 4096
17
18 typedef int MyFunc(const char * ,const struct stat *, int);
19
20 static MyFunc myfunc;
21 static int myftw(char *, MyFunc*);
22 static int dopath(const char* filename, int depth, MyFunc*);
23
24 static long nreg, ndir, nblk, nchr, nfifo, nslink, nsock, ntot;
25
26 static void err_doit(int errnoflag, int error, const char *fmt, va_list ap);
27 err_quit(const char *fmt, ...);
28
29 int main(int argc, char *argv[])
30 {
31     int ret;
32
33     if (argc != 2)
34     {
35         err_quit("Usage: ./a.out <start_directory>\n");
36     }
37
38     ret = myftw(argv[1], myfunc);
39
```

```

40     ntot = nreg + ndir +  nblk + nchr +  nfifo + nslink + nsock;
41
42     if (ntot == 0)
43         ntot = 1;
44
45     printf("_____\n\n");
46     printf("Regular files:\t%7ld, %5.2f %%\n", nreg, nreg*100.0/ntot);
47     printf("Catalogs:\t%7ld, %5.2f %%\n", ndir, ndir*100.0/ntot);
48     printf("Block-devices:\t%7ld, %5.2f %%\n", nblk, nblk*100.0/ntot);
49     printf("Char-devices:\t%7ld, %5.2f %%\n", nchr, nchr*100.0/ntot);
50     printf("FIFOs:\t\t%7ld, %5.2f %%\n", nfifo, nfifo*100.0/ntot);
51     printf("Sym-links:\t%7ld, %5.2f %%\n", nslink, nslink*100.0/ntot);
52     printf("Sockets:\t%7ld, %5.2f %%\n\n", nsock, nsock*100.0/ntot);
53     printf("Total:\t%7ld\n", ntot);
54
55     exit(ret);
56 }
57
58 static void err_doit(int errnoflag, int error, const char *fmt, va_list ap)
59 {
60     char buf[MAXLINE];
61
62     vsnprintf(buf, MAXLINE-1, fmt, ap);
63     if (errnoflag)
64         snprintf(buf+strlen(buf), MAXLINE-strlen(buf)-1, ": %s",
65                 strerror(error));
66     strcat(buf, "\n");
67     fflush(stdout);      /* in case stdout and stderr are the same */
68     fputs(buf, stderr);
69     fflush(NULL);       /* flushes all stdio output streams */
70 }
71 err_quit(const char *fmt, ...)
72 {
73     va_list ap;
74
75     va_start(ap, fmt);
76     err_doit(0, 0, fmt, ap);
77     va_end(ap);
78     exit(1);
79 }
80
81 static int myftw(char * pathname, MyFunc *func)
82 {
83     return(dopath(pathname, 0, func));
84 }
85

```

```

86 static int dopath(const char* filename, int depth, MyFunc * func)
87 {
88     struct stat statbuf;
89     struct dirent * dirp;
90     DIR * dp;
91     int ret = 0;
92
93     if (lstat(filename, &statbuf) < 0)
94         return(func(filename, &statbuf, FTW_NS));
95
96     for (int i = 0; i < depth; ++i)
97         printf("|\\t");
98
99     if (S_ISDIR(statbuf.st_mode) == 0)
100         return(func(filename, &statbuf, FTW_F));
101
102     if ((ret = func(filename, &statbuf, FTW_D)) != 0)
103         return(ret);
104
105     if ((dp = opendir(filename)) == NULL)
106         return(func(filename, &statbuf, FTW_DNR));
107
108     chdir(filename);
109     while ((dirp = readdir(dp)) != NULL && ret == 0)
110     {
111         if (strcmp(dirp->d_name, ".") != 0 &&
112             strcmp(dirp->d_name, "..") != 0 )
113         {
114             ret = dopath(dirp->d_name, depth+1, func);
115         }
116     }
117
118     chdir("..");
119
120     if (closedir(dp) < 0)
121         perror("Unable to close directory");
122
123     return(ret);
124 }
125
126 static int myfunc(const char* pathame, const struct stat * statptr, int
127     type)
128 {
129     switch(type)
130     {
131         case FTW_F:

```

```

131         printf( "— %s\n", pathame);
132     switch( statptr->st_mode & S_IFMT)
133     {
134         case S_IFREG: nreg++; break;
135         case S_IFBLK: nblk++; break;
136         case S_IFCHR: nchr++; break;
137         case S_IFIFO: nfifo++; break;
138         case S_IFLNK: nslink++; break;
139         case S_IFSOCK: nsock++; break;
140         case S_IFDIR:
141             perror("The directory is of type FTW_F"); return(-1);
142     }
143     break;
144 case FTW_D:
145     printf( "— %s/\n", pathame);
146     ndir++; break;
147 case FTW_DNR:
148     perror("Blocked access to one of the directories!"); return(-1);
149 case FTW_NS:
150     perror("Function error stat!"); return(-1);
151 default:
152     perror("Unknown file type!"); return(-1);
153 }
154
155 return(0);
156 }

```

1.2 Результат

Демонстрация запуска программы с неправильными аргументами:

```
san_sanchez@LEX > ~/workspace/0C/sem_2/lab_02/source/test > ./a
Usage: ./a.out <start_directory>

✖ san_sanchez@LEX > ~/workspace/0C/sem_2/lab_02/source/test > ./a error
Function error stat!: No such file or directory
```

Рисунок 1.1 — Скришот запуска программы.

Демонстрация корректного запуска программы:

```
san_sanchez@LEX > ~/workspace/0C/sem_2/lab_02/source/test > ./a ~/ | head -30
-- /home/san_sanchez//
-- pycharm/
--   pycharm-edu-2019.3.2/
--     license/
--       xmlrpc_license.txt
--       LICENSE.txt
--       junit_license.txt
--       winp_license.txt
--       eclipse_license.txt
--       kryo-license.txt
--       jaxen_license.txt
--       XStream_license.txt
--       third-party-libraries.html
--       imgscalr_license.txt
--       ant_license.txt
--       swingx_license.txt
--       trove4j_license.txt
--       log4j_license.txt
--       saxon-conditions.html
--       jgoodies_forms_license.txt
--       eclipse_license2.txt
--       jdom_license.txt
--       miglayout_swing_license.txt
--       NOTICE.txt
--       jaxb_license.txt
--       javahelp_license.txt
--       javolution_license.txt
--       nanoxml_license.txt
--       picoContainer_license.txt
--       gson_license.txt
```

Рисунок 1.2 — Скришот вывода программы.

```

san_sanchez@LEX ~/workspace/OC/sem_2/lab_02/source > ./a ~/workspace/OC/sem_2/lab_02
-- /home/san_sanchez/workspace/OC/sem_2/lab_02/
|
|   -- report/
|   |
|   |   -- theory.tex
|   |   -- data/
|   |   |
|   |   |   -- image/
|   |   |   |
|   |   |   |   -- error.png
|   |   |   |   -- work_1.png
|   |   |   -- graph/
|   |   |   -- pdf/
|   |   -- practical.aux
|   |   -- missfont.log
|   |   -- theory.fdb_latexmk
|   |   -- main.log
|   |   -- main.aux
|   |   -- main.pdf
|   |   -- theory.flx
|   |   -- pdflatex17394.flx
|   |   -- concl.tex
|   |   -- titul.tex
|   |   -- main.tex
|   |   -- theory.aux
|   |   -- intro.aux
|   |   -- main.nlo
|   |   -- title.aux
|   |   -- titul.aux
|   |   -- main.fdb_latexmk
|   |   -- intro.tex
|   |   -- concl.aux
|   |   -- main.synctex.gz
|   |   -- practical.tex
|   |   -- main.flx
|   |   -- main.out
|   |   -- theory.log
|
|   -- source/
|   |
|   |   -- lab_2.c
|   |   -- a
|
-----

```

Рисунок 1.3 — Скришот результата работы программы (часть 1).

```

-----
Regular files:      29, 80.56 %
Catalogs:          7, 19.44 %
Block-devices:     0, 0.00 %
Char-devices:      0, 0.00 %
FIFOs:             0, 0.00 %
Sym-links:         0, 0.00 %
Sockets:           0, 0.00 %

Total:             36

```

Рисунок 1.4 — Скришот результата работы программы (часть 2).