



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 ИУ, Информатика и системы управления

КАФЕДРА ИУ7, Программное обеспечение ЭВМ и информационные технологии

# ОТЧЕТ

## к лабораторной работе №2

*по курсу*

***“Операционные системы”***  
***На тему: “Файлы и каталоги”***

Студент      ИУ7-64Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)      **А.А. Андреев**  
(И.О.Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)      **Н.Ю. Рязанова**  
(И.О.Фамилия)

2022 г.

## **Оглавление**

Задание	3
Реализация	4
Результат работы программы	7

## Задание

1. Структурировать исходный код программы в листинге 4.7.
2. Изменить программу так, чтобы она выводила на экран дерево каталогов в наглядной форме при помощи символов \*, ../, | и тд.
3. Изменить функцию `myfwt()` так, чтобы она передавала функции `lstat()` не полный путь к файлу, а только его имя каждый раз, когда встречается каталог.

# Реализация

В листингах 1-3 приведен код программы.

## Листинг 1

```
1. #include <stdio.h>
2. #include <dirent.h> //opendir()/readdir()/closedir()
3. #include <sys/stat.h>
4. #include <string.h>
5. #include <unistd.h>
6. #include <errno.h>
7.
8. #define OK 0
9. #define ERROR -1
10. #define ERROR_STAT 1
11. #define ERROR_OPEN_DIR 2
12.
13. #define START_DEPTH 0
14.
15. // Рекурсивный проход по пути, его вывод и возврат ошибки
16. static int dopath(const char *filename, int
    recursion_depth);
17.
18. // Проверка наличия аргумента пути
19. int is_dir_in_arg(int argc);
20.
21. int main(int argc, char *argv[])
22. {
23.     int error_status;
24.
25.     // Параметр отсутствует
26.     if (is_dir_in_arg(argc) == OK)
27.         error_status = dopath(argv[1], START_DEPTH); //
    начиная с папки в параметре
28.     else
29.         error_status = dopath("./", 0); // начиная с текущей
    папки
30.
31.     return error_status;
32. }
33.
34. int is_dir_in_arg(int argc)
35. {
36.     if (argc == 2)
37.         return OK;
38.     return ERROR;
39. }
40.
41. static int dopath(const char *filename, int
    recursion_depth)
42. {
43.     struct stat statbuf;
44.     struct dirent *dirp;
45.     DIR *dp;
```

46.

## Листинг 2

```
47.  
48.     ///'.' - указатель на сам каталог  
49.     ///'..' - указатель на родительский каталог  
50.     if (strcmp(filename, ".") == 0 || strcmp(filename, "..")  
51.         == 0)  
52.         return OK;  
53.     //Вызов lstat() идентичен stat(), но в случае, если  
54.     filename является символьной ссылкой, то возвращается  
55.     информация о самой ссылке, а не о файле, на который она  
56.     указывает; возвращает информацию об указанном файле  
57.     if (lstat(filename, &statbuf) < 0) {  
58.         switch(errno)  
59.         {  
60.             case EBADF:  
61.                 printf("Неверный описатель файлового  
62.                 дескриптора.");  
63.                 break;  
64.             case ENOENT:  
65.                 printf("Компонент полного имени файла  
66.                 filename не существует или полное имя является пустой  
67.                 строкой.");  
68.                 break;  
69.             case ENOTDIR:  
70.                 printf("Компонент пути не является  
71.                 каталогом. ");  
72.                 break;  
73.             case ELOOP:  
74.                 printf("При поиске файла встретилась  
75.                 символьная ссылка.");  
76.                 break;  
77.             case EFAULT:  
78.                 printf("Некорректный адрес. ");  
79.                 break;  
80.             case EACCES:  
81.                 printf("Запрещен доступ. ");  
82.                 break;  
83.             case ENOMEM:  
84.                 printf("Недостаточно памяти в  
85.                 системе.");  
86.                 break;  
87.             case ENAMETOOLONG:  
88.                 printf("Слишком длинное название файла.  
89.                 ");  
90.                 break;  
91.             case EOVERFLOW:  
92.                 printf("Некоторые значения были слишком  
93.                 большими, чтобы быть представленными в возвращаемой  
94.                 структуре.s");  
95.                 break;  
96.         }  
97.     }
```

## Листинг 3

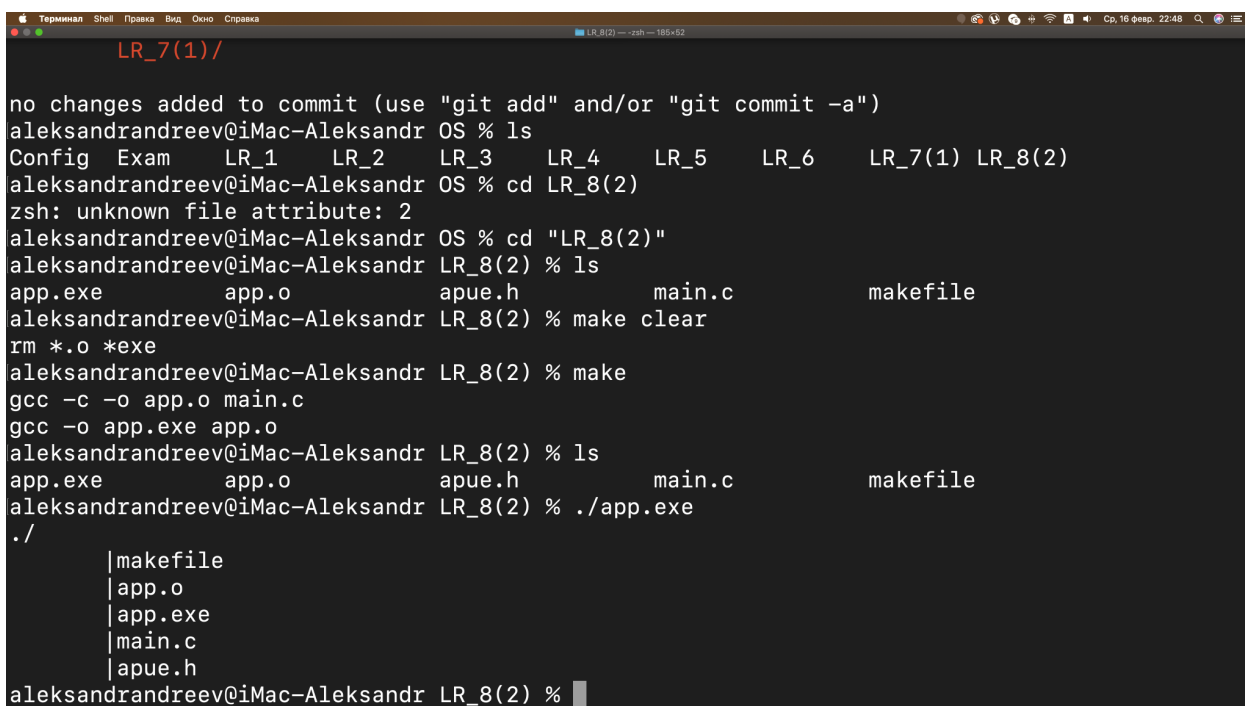
```

86.  printf("Ошибка вызова функции stat");
87.      return ERROR_STAT;
88.
89.  }
90.
91.      //печать отступов
92.      for (int i = 0; i < recursion_depth; ++i)
93.          printf("      |");
94.
95.
96.      /* печатаем имя файла */
97.      // s_isdir проверяет, является ли данный файл каталогом
98.      if (S_ISDIR(statbuf.st_mode) == 0) {
99.          printf("%s\n", filename);
100.         return OK;
101.     }
102.
103.     /* каталог */
104.     printf("%s\n", filename);
105.     if ((dp = opendir(filename)) == NULL) {
106.         printf("couldn't open directory '%s'\n", filename);
107.         return ERROR_OPEN_DIR;
108.     }
109.
110.     // chdir() изменяет текущий рабочий каталог вызывающего
    процесса на каталог, указанный в path; PATH — переменная
    окружения, представляющая собой набор каталогов, в которых
    расположены исполняемые файлы; нужен для коротких имен
111.
112.     chdir(filename);
113.
114.     // для каждого элемента каталога
115.     // Функция readdir() возвращает указатель на следующую
    запись каталога в структуре dirent, прочитанную из потока
    каталога. Каталог указан в dir. Функция возвращает NULL по
    достижении последней записи или если была обнаружена ошибка.
116.     while ((dirp = readdir(dp)) != NULL)
117.         dopath(dirp -> d_name, recursion_depth + 1);
118.     chdir("../");
119.
120.     //закрывает поток каталога и освобождает ресурсы,
    выделенные ему.
121.     //Она возвращает 0 в случае успеха и -1 при наличии
    ошибки.
122.     closedir(dp);
123.
124.     return OK;
125. }
126.

```

## Результат работы программы

На рисунках 1 и 2 приведен результат работы программы с аргументом пути к директории и без.



```
LR_7(1)/
no changes added to commit (use "git add" and/or "git commit -a")
aleksandrareev@iMac-Aleksandr OS % ls
Config Exam LR_1 LR_2 LR_3 LR_4 LR_5 LR_6 LR_7(1) LR_8(2)
aleksandrareev@iMac-Aleksandr OS % cd LR_8(2)
zsh: unknown file attribute: 2
aleksandrareev@iMac-Aleksandr OS % cd "LR_8(2)"
aleksandrareev@iMac-Aleksandr LR_8(2) % ls
app.exe app.o apue.h main.c makefile
aleksandrareev@iMac-Aleksandr LR_8(2) % make clear
rm *.o *.exe
aleksandrareev@iMac-Aleksandr LR_8(2) % make
gcc -c -o app.o main.c
gcc -o app.exe app.o
aleksandrareev@iMac-Aleksandr LR_8(2) % ls
app.exe app.o apue.h main.c makefile
aleksandrareev@iMac-Aleksandr LR_8(2) % ./app.exe
./
|makefile
|app.o
|app.exe
|main.c
|apue.h
aleksandrareev@iMac-Aleksandr LR_8(2) %
```

Рисунок 1: Результат работы программы без аргумента директории

```
Терминал Shell Правка Вид Окно Справка LR_8(2) — .zsh — 185x52
|
| |OS_LR_2_U17-548_Andreev_A.A.pdf
| |RL_PD_RL
| |main.asm
| |
aleksandr@aleksandr@Mac-Aleksandr LR_8(2) % ./app.exe ../
../
| .DS_Store
| Config
| |@ponosov.doc
| |na6_pa61_Int8h.doc
| |na6_pa61sew.doc
| |3y6xoa.pdf
| LR_1
| |OS_LR1_part2_Andreev_A.A_U07-548_report.pdf
| |Int8h_model.drawio.drawio
| |Int8h_model.drawio
| |sub_04_listing.asm
| |export_listing.asm
| |Int8h_model.drawio.html
| |OS_LR1_part1_report.pdf
| |Int_8h_listing.asm
| LR_6
| |docs
| | |Pixrep_windows-dlya-professionalov-sozdanie-effektivnyh-win32.pdf
| | |na6_pa6_6_wind_2021.docx
| | |src
| | | |example.cpp
| | | |Makefile
| | | |main.cpp
| LR_7(1)
| |main.o
| |makefile
| |app.o
| |app.exe
| |main.c
| |apue.h
| Exam
| |Theory.pages
| |Theory.pdf
| |Theory_compressed.pdf
| |Listings_compressed.pdf
| |Listings.pages
| |Listings.pdf
| |osl.pdf
| |exam.pdf
| |Lectures.pdf
| |Lectsls_seminary.pdf
| LR_3
| |pipeline
| |symlink
| |datlink
| |pipe
| |out
| | |msin.o
| |makefile
| |docs
| | |linux-lab1_2020_link.doc
| |readme.md
| |filename
```

Рисунок 2: Результат работы программы с аргументом директории, Часть 1



```
Терминал Shell Правка Вид Окно Справка
LR_8(2) ~ - ssh - 180x52

| makefile
| docs
| | linux-lab1_2020_link.doc
| readme.md
| filename
| filetype
| src
| | main.c
LR_8(2)
| makefile
| app.o
| app.exe
| main.c
| apue.h
LR_4
| out
| | task_4.o
| | task_3.o
| | task_1.o
| | task_5.o
| makefile
| meta
| | factorial
| | | out
| | | | first_thread.o
| | | | first_thread.out
| | | | a.out
| | | | factorial.c
| | | average
| | | | out
| | | | second_thread.out
| | | | second_thread.o
| | | | average.c
| | | | a.out
| OS_LR4_report.pdf
| Лаб2_Оптимизация_fork2021.doc
| /IP2_UNIX_OC_UI7-545_Андреев_Александр_записи_edited.pdf
| /IP2_UNIX_OC_UI7-545_Андреев_Александр_записи.pdf
| src
| | task_2.c
| | task_5_not_working_test.c
| | task_5.c
| | task_1.c
| | task_4.c
| | task_3.c
LR_5
| makefile
| p_2.c
| p_1.c
LR_2
| PD_RL_PD
| | main.asm
| | protac.asm
| docs
| | OS_LR_2_UI7-545_Андреев_А.А.pdf
| RL_PD_RL
| | main.asm
aleksandrdrzev@Mac-Aleksandr LR_8(2) %
```

Рисунок 3: Результат работы программы с аргументом директории, Часть 2