

# **Отчёт по лабораторной работе №10**

**2023**

Просина Ксения Максимовна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
3.1	Задание для самостоятельной работы . . . . .	10
<b>4</b>	<b>Выводы</b>	<b>13</b>
	<b>Список литературы</b>	<b>14</b>

# Список иллюстраций

2.1	Таблица 10.1. Двоичный, буквенный и восмеричный способ записи триады прав доступа . . . . .	7
3.1	Создание каталога и файлов . . . . .	8
3.2	Создание исполняемого файла и проверка . . . . .	8
3.3	Изменение прав . . . . .	9
3.4	Изменение прав . . . . .	9
3.5	Символьный вид . . . . .	10
3.6	Двоичный вид . . . . .	10
3.7	Код . . . . .	11
3.8	Компиляция . . . . .	12
3.9	Итог . . . . .	12

## Список таблиц

# **1 Цель работы**

Приобретение навыков написания программ для работы с файлами

## 2 Теоретическое введение

ОС GNU/Linux является многопользовательской операционной системой. И для обеспечения защиты данных одного пользователя от действий других пользователей существуют специальные механизмы разграничения доступа к файлам. Кроме ограничения доступа, данный механизм позволяет разрешить другим пользователям доступ данным для совместной работы. Права доступа определяют набор действий (чтение, запись, выполнение), разрешённых для выполнения пользователям системы над файлами. Для каждого файла пользователь может входить в одну из трех групп: владелец, член группы владельца, все остальные. Для каждой из этих групп может быть установлен свой набор прав доступа. Владелец файла является его создатель. Для предоставления прав доступа другому пользователю или другой группе командой `chown [ключи] [:новая_группа]` или `chgrp [ключи] < новая_группа >` Набор прав доступа задается тройками битов и состоит из прав на чтение, запись и исполнение файла. В символьном представлении он имеет вид строк `gwx`, где вместо любого символа может стоять дефис. Всего возможно 8 комбинаций, приведенных в таблице 10.1. Буква означает наличие права (установлен в единицу второй бит триады `r` — чтение, первый бит `w` — запись, нулевой бит `x` — исполнение), а дефис означает отсутствие права (нулевое значение соответствующего бита). Также права доступа могут быть представлены как восьмеричное число. Так, права доступа `rw`- (чтение и запись, без исполнения) понимаются как три двоичные цифры `110` или как восьмеричная цифра `6`

Двоичный	Буквенный	Восмеричный
111	rwx	7
110	rw-	6
101	r-x	5
100	r--	4
011	-wx	3
010	-w-	2
001	--x	1
000	---	0

Рис. 2.1: Таблица 10.1. Двоичный, буквенный и восмеричный способ записи триады прав доступа

Полная строка прав доступа в символьном представлении имеет вид: Так, например, права rwx r-x -x выглядят как двоичное число 111 101 001, или восьмеричное 751. Свойства (атрибуты) файлов и каталогов можно вывести на терминал с помощью команды ls с ключом -l. Так например, чтобы узнать права доступа к файлу README можно узнать с помощью следующей команды: \$ls -l /home/debugger/README -rwxr-xr- 1 debugger users 0 Feb 14 19:08 /home/debugger/README

### 3 Выполнение лабораторной работы

1. Создайте каталог для программ лабораторной работы № 10, перейдите в него и создайте файлы lab10-1.asm, readme-1.txt и readme-2.txt: mkdir ~/work/arch-pc/lab09 cd ~/work/arch-pc/lab09 touch lab10-1.asm readme-1.txt readme-2.txt

```
kmprosina@dk4n70 ~ $ mkdir ~/work/arch-pc/lab10
kmprosina@dk4n70 ~ $ cd ~/work/arch-pc/lab10
kmprosina@dk4n70 ~/work/arch-pc/lab10 $ touch lab10-1.asm readme-1.txt readme-2.txt
```

Рис. 3.1: Создание каталога и файлов

2. Введите в файл lab10-1.asm текст программы из листинга 10.1 (Программа записи в файл сообщения). Создайте исполняемый файл и проверьте его работу.

```
kmprosina@dk8n51 ~/work/arch-pc/lab10 $ nasm -f elf -g -l lab10-1.lst lab10-1.asm
kmprosina@dk8n51 ~/work/arch-pc/lab10 $ ld -m elf_i386 -o lab10-1 lab10-1.o
kmprosina@dk8n51 ~/work/arch-pc/lab10 $ ./lab10-1
Введите строку для записи в файл: Hello world!
kmprosina@dk8n51 ~/work/arch-pc/lab10 $ ls -l
итого 34
-rw-r--r-- 1 kmprosina studsci 3942 ноя  8 14:10 in_out.asm
-rwxr-xr-x 1 kmprosina studsci 9764 дек 15 14:34 lab10-1
-rw-r--r-- 1 kmprosina studsci 1287 дек 15 13:50 lab10-1.asm
-rw-r--r-- 1 kmprosina studsci 13713 дек 15 14:34 lab10-1.lst
-rw-r--r-- 1 kmprosina studsci 2544 дек 15 14:34 lab10-1.o
-rw-r--r-- 1 kmprosina studsci 13 дек 15 13:53 readme-1.txt
-rw-r--r-- 1 kmprosina studsci  0 дек 14 15:10 readme-2.txt
kmprosina@dk8n51 ~/work/arch-pc/lab10 $ cat readme-1.txt
Hello world!
```

Рис. 3.2: Создание исполняемого файла и проверка

3. С помощью команды chmod измените права доступа к исполняемому файлу



lab10-1, запретив его выполнение. Попробуйте выполнить файл. Объясните результат.

```
kmprosina@dk8n51 ~/work/arch-pc/lab10 $ chmod 000 lab10-1
kmprosina@dk8n51 ~/work/arch-pc/lab10 $ ls -l
итого 34
-rw-r--r-- 1 kmprosina studsci 3942 ноя  8 14:10 in_out.asm
----- 1 kmprosina studsci 9764 дек 15 14:34 lab10-1
-rw-r--r-- 1 kmprosina studsci 1287 дек 15 13:50 lab10-1.asm
-rw-r--r-- 1 kmprosina studsci 13713 дек 15 14:34 lab10-1.lst
-rw-r--r-- 1 kmprosina studsci 2544 дек 15 14:34 lab10-1.o
-rw-r--r-- 1 kmprosina studsci 13 дек 15 13:53 readme-1.txt
-rw-r--r-- 1 kmprosina studsci 0 дек 14 15:10 readme-2.txt
kmprosina@dk8n51 ~/work/arch-pc/lab10 $ ./lab10-1
bash: ./lab10-1: Отказано в доступе
```

Рис. 3.3: Изменение прав

С помощью `chmod 000 lab10-1` я запретила выполнение файла абсолютно для всех. Первый ноль запретил взаимодействие с файлом владельцу, второй - группе, а третий - другим. В итоге при попытке выполнения вышел текст, повествующий о том, что выполнить команду невозможно из-за отсутствия к нему доступа.

4. С помощью команды `chmod` измените права доступа к файлу `lab10-1.asm` с исходным текстом программы, добавив права на исполнение. Попробуйте выполнить его и объясните результат.

```
kmprosina@dk8n51 ~/work/arch-pc/lab10 $ chmod go+x lab10-1.asm
kmprosina@dk8n51 ~/work/arch-pc/lab10 $ ls -l lab10-1.asm
-rw-r-xr-x 1 kmprosina studsci 1287 дек 15 13:50 lab10-1.asm
```

Рис. 3.4: Изменение прав

С помощью `chmod go+x lab10-1.asm` я дала файлу возможность исполнения

5. В соответствии с вариантом в таблице предоставить права доступа к файлу `readme-1.txt` представленные в символьном виде, а для файла `readme-2.txt` – в двоичном виде. Проверить правильность выполнения с помощью команды `ls -l`

Мой вариант - 19. Следовательно, мне необходимо сделать rw- rwx r-x 111 111 001

```
kmprosina@dk4n60 ~/work/arch-pc/lab10 $ ls -l readme-1.txt
-rwxrwxrwx 1 kmprosina studsci 13 дек 15 13:53 readme-1.txt
kmprosina@dk4n60 ~/work/arch-pc/lab10 $ chmod u-x readme-1.txt
kmprosina@dk4n60 ~/work/arch-pc/lab10 $ chmod o-w readme-1.txt
kmprosina@dk4n60 ~/work/arch-pc/lab10 $ ls -l readme-1.txt
-rw-rwxr-x 1 kmprosina studsci 13 дек 15 13:53 readme-1.txt
kmprosina@dk4n60 ~/work/arch-pc/lab10 $
```

Рис. 3.5: Символьный вид

С помощью новых кодов я смогла выполнить задачу. и значит, что я редактирую права создателя, o - права других. x - право на исполнение, w - право на запись

```
kmprosina@dk4n60 ~/work/arch-pc/lab10 $ ls -l readme-2.txt
-rwxrwxrwx 1 kmprosina studsci 0 дек 14 15:10 readme-2.txt
kmprosina@dk4n60 ~/work/arch-pc/lab10 $ chmod 771 readme-2.txt
kmprosina@dk4n60 ~/work/arch-pc/lab10 $ ls -l readme-2.txt
-rwxrwx--x 1 kmprosina studsci 0 дек 14 15:10 readme-2.txt
kmprosina@dk4n60 ~/work/arch-pc/lab10 $
```

Рис. 3.6: Двоичный вид

Похожее есть и в двоичном коде. 7 значит 111 в двоичном коде => чтение, запись и исполнение доступны. 1 - 001 в двоичном коде. Он значит, что доступно только право на исполнение.

### 3.1 Задание для самостоятельной работы

- 1) Напишите программу, работающую по следующему алгоритму

```

4 %include 'in_out.asm'
5 SECTION .data
6 filename db 'name.txt', 0h ; Имя файла
7 msg db 'Как вас зовут?: ', 0h ; Сообщение
8 msg2 db 'Меня зовут ', 0h ; Сообщение
9 SECTION .bss
10 contents resb 255 ; переменная для вводимой строки
11 SECTION .text
12 global _start
13 _start:
14 ; --- Печать сообщения `msg`
15 mov eax,msg
16 call sprint
17 ; ---- Запись введенной с клавиатуры строки в `contents`
18 mov ecx, contents
19 mov edx, 255
20 call sread
21 ; --- Открытие существующего файла (`sys_open`)
22 mov ecx, 0777o ; открываем для записи (2)
23 mov ebx, filename
24 mov eax, 5
25 int 80h
26 ; --- Запись дескриптора файла в `esi`
27 mov esi, eax
28 ; --- Расчет длины введенной строки
29 mov eax, contents ; в `eax` запишется количество
30 call slen ; введенных байтов
31 ; --- Записываем в файл `contents` (`sys_write`)
32 mov edx, eax
33 mov ecx, msg2
34 mov ebx, esi
35 mov eax, 4
36 int 80h
37
38 mov edx, eax
39 mov ecx, contents
40 mov ebx, esi
41 mov eax, 4
42 int 80h
43 ; --- Закрываем файл (`sys_close`)
44 mov ebx, esi
45 mov eax, 6
46 int 80h
47 call quit

```

Рис. 3.7: Код

Создать исполняемый файл и проверить работу. Проверить наличие файла и его содержимое с помощью `ls` и `cat`

```

kmprosina@dk4n60 ~/work/arch-pc/lab10 $ nasm -f elf -g -l lab10-2.lst lab10-2.asm
kmprosina@dk4n60 ~/work/arch-pc/lab10 $ ld -m elf_i386 -o lab10-2 lab10-2.o
kmprosina@dk4n60 ~/work/arch-pc/lab10 $ ./lab10-2
Как вас зовут?: Просина Ксения
kmprosina@dk4n60 ~/work/arch-pc/lab10 $ ls -l
итого 63
-rw-r--r-- 1 kmprosina studsci 3942 ноя  8 14:10 in_out.asm
----- 1 kmprosina studsci 9764 дек 15 14:34 lab10-1
-rw-r-xr-x 1 kmprosina studsci 1287 дек 15 13:50 lab10-1.asm
-rw-r--r-- 1 kmprosina studsci 13713 дек 15 14:34 lab10-1.lst
-rw-r--r-- 1 kmprosina studsci 2544 дек 15 14:34 lab10-1.o
-rwxr-xr-x 1 kmprosina studsci 9776 дек 15 17:39 lab10-2
-rw-r--r-- 1 kmprosina studsci 1371 дек 15 17:39 lab10-2.asm
-rw-r--r-- 1 kmprosina studsci 14005 дек 15 17:39 lab10-2.lst
-rw-r--r-- 1 kmprosina studsci 2576 дек 15 17:39 lab10-2.o
-rwxrwxrwx 1 kmprosina studsci  0 дек 15 17:33 name.txt
-rw-rwxr-x 1 kmprosina studsci  13 дек 15 13:53 readme-1.txt
-rwxrwx--x 1 kmprosina studsci  0 дек 14 15:10 readme-2.txt

```

Рис. 3.8: Компиляция

```

kmprosina@dk4n60 ~/work/arch-pc/lab10 $ cat name.txt
Меня зовут Просина Ксения
kmprosina@dk4n60 ~/work/arch-pc/lab10 $

```

Рис. 3.9: Итог

## **4 Выводы**

В ходе работы были приобретены навыки написания программ для работы с файлами и изменение их прав

## **Список литературы**