

# **Лабораторная работы №10**

**Работа с файлами средствами Nasm**

Сырцева Анастасия Романовна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>10</b>
<b>5</b>	<b>Самостоятельная работа</b>	<b>14</b>
<b>6</b>	<b>Выводы</b>	<b>17</b>

# Список иллюстраций

3.1	Отображение прав доступа . . . . .	8
4.1	Создание каталога и файлов . . . . .	10
4.2	Программа записи в файл сообщения . . . . .	11
4.3	Копирование внешнего файла . . . . .	12
4.4	Запуск программы . . . . .	12
4.5	Изменение прав доступа файла lab10-1 . . . . .	12
4.6	Запуск файла . . . . .	12
4.7	Изменение прав доступа файла lab10-1.asm . . . . .	12
4.8	Запуск программы . . . . .	12
4.9	Вариант №1 . . . . .	13
4.10	Предоставление прав доступа в символьном виде . . . . .	13
4.11	предоставление прав доступа в двоичном виде . . . . .	13
4.12	Проверка правильности выполнения команд . . . . .	13
5.1	Текст программы, работающей по алгоритму из задания . . . . .	15
5.2	Запуск программы и проверка её работы . . . . .	16

# Список таблиц

3.1	Двоичный, буквенный и восьмеричный способ записи триады прав доступа . . . . .	7
3.2	Возможные значения аргументов команды <code>chmod</code> . . . . .	8

# 1 Цель работы

Целью лабораторной работы является приобретение навыков написания программ для работы с файлами.

## 2 Задание

- Ввести в файл lab10-1.asm текст программы записи в файл сообщения, проверить его работу;
- Изменить права доступа к исполняемому файлу lab10-1, запретив его выполнение;
- С помощью команды `chmod` измените права доступа к файлу lab10-1.asm с исходным текстом программы, добавив права на исполнение;
- В соответствии с вариантом из лабораторной №7 предоставить права доступа к файлу `readme-1.txt` представленные в символьном виде, а для файла `readme-2.txt` – в двоичном виде.

### 3 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. 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 3.1. Буква означает наличие права, а дефис означает отсутствие права. Также права доступа могут быть представлены как восьмеричное число.

Таблица 3.1: Двоичный, буквенный и восьмеричный способ записи триады прав доступа

Двоич- ный	Буквенный	Восьмеричный
111	<code>gwx</code>	7
110	<code>gw-</code>	6

Двоич- ный	Буквенный	Восьмеричный
101	r-x	5
100	r-	4
011	-wx	3
010	-w-	2
001	-x	1
000	-	0

Свойства (атрибуты) файлов и каталогов можно вывести на терминал с помощью команды `ls` с ключом `-l`. В первой колонке показаны текущие права доступа, далее указан владелец файла и группа (рис. 3.1):

```

-      rwx      r-x      r-x      l debugger user
|      |      |      |
тип    владелец группа  остальные
type  owner  group  others

```

Рис. 3.1: Отображение прав доступа

Тип файла определяется первой позицией, это может быть: каталог — `d`, обычный файл — дефис (`-`) или символическая ссылка на другой файл — `l`. Следующие 3 набора по 3 символа определяют конкретные права для конкретных групп: `r` — разрешено чтение файла, `w` — разрешена запись в файл; `x` — разрешено исполнение файла и дефис (`-`) — право не дано. Для изменения прав доступа служит команда `chmod`, которая понимает как символическое, так и числовое указание прав. возможные значения аргументов команды представлены в таблице 3.2.

Таблица 3.2: Возможные значения аргументов команды `chmod`

Категория	Обозначение	Значение
Принадлежность	<code>u</code>	Владелец



Категория	Обозначение	Значение
Действие	g	Группа владельца
	o	Прочие пользователи
	a	Все пользователи
	+	Добавить набор прав
	-	Отменить набор прав
	=	назначить набор прав
Право	r	Право на чтение
	w	Право на запись
	x	Право на исполнение

## 4 Выполнение лабораторной работы

Создаю рабочий каталог и файлы в нём для выполнения данной лабораторной работы (рис. 4.1).

```
arsihrceva@dk6n57 ~/work/arch-pc $ mkdir lab10  
arsihrceva@dk6n57 ~/work/arch-pc $ cd lab10  
arsihrceva@dk6n57 ~/work/arch-pc/lab10 $ touch lab10-1.asm readme-1.txt readme-2.txt
```

Рис. 4.1: Создание каталога и файлов

Ввожу в файл lab10-1.asm программу записи в файл сообщения(рис. 4.2).

```

1 %include 'in_out.asm'
2
3 SECTION .data
4 filename db 'readme.txt', 0h
5 msg db 'Введите строку для записи в файл: ', 0h
6
7 SECTION .bss
8 contents resb 255
9
10 SECTION .text
11     global _start
12 _start:
13
14 mov eax,msg
15 call sprint
16
17 mov ecx, contents
18 mov edx, 255
19 call sread
20
21 mov ecx, 2
22 mov ebx, filename
23 mov eax, 5
24 int 80h
25
26 mov esi, eax
27
28 mov eax, contents
29 call slen
30
31 mov edx, eax
32 mov ecx, contents
33 mov ebx, esi
34 mov eax, 4
35 int 80h
36
37 mov ebx, esi
38 mov eax, 6
39 int 80h
40 call quit

```

Рис. 4.2: Программа записи в файл сообщения

Для корректной работы программы копирую в рабочую папку внешний файл in\_out.asm(рис. 4.3).

```
arsihrceva@dk6n57 ~/work/arch-pc/lab10 $ cp ~/work/in_out.asm ~/work/arch-pc/lab10/in_out.asm
```

Рис. 4.3: Копирование внешнего файла

Создаю и запускаю исполняемый файл, проверяю его работу(рис. 4.4).

```
arsihrceva@dk6n57 ~/work/arch-pc/lab10 $ nasm -f elf lab10-1.asm
arsihrceva@dk6n57 ~/work/arch-pc/lab10 $ ld -m elf_i386 -o lab10-1 lab10-1.o
arsihrceva@dk6n57 ~/work/arch-pc/lab10 $ ./lab10-1
Введите строку для записи в файл: Hello world!
```

Рис. 4.4: Запуск программы

С помощью команды `chmod` запрещаю выполнение файла `lab10-1`(рис. 4.5).

```
arsihrceva@dk2n21 ~/work/arch-pc/lab10 $ chmod u-x lab10-1
```

Рис. 4.5: Изменение прав доступа файла `lab10-1`

Запускаю файл с изменёнными правами доступа(рис. 4.6).

```
arsihrceva@dk2n21 ~/work/arch-pc/lab10 $ ./lab10-1
bash: ./lab10-1: Отказано в доступе
```

Рис. 4.6: Запуск файла

Выдаётся отказ, так как я запретила владельцу, себе, запуск файла. Разрешаю запуск файла `lab10-1.asm` с помощью той же команды(рис. 4.7).

```
arsihrceva@dk2n21 ~/work/arch-pc/lab10 $ chmod u+x lab10-1.asm
```

Рис. 4.7: Изменение прав доступа файла `lab10-1.asm`

Создаю и запускаю исполняемый файл(рис. 4.8).

```
arsihrceva@dk2n21 ~/work/arch-pc/lab10 $ nasm -f elf lab10-1.asm
arsihrceva@dk2n21 ~/work/arch-pc/lab10 $ ld -m elf_i386 -o lab10-1 lab10-1.o
arsihrceva@dk2n21 ~/work/arch-pc/lab10 $ ./lab10-1
Введите строку для записи в файл: Hallow world!
```

Рис. 4.8: Запуск программы

Программа запустилась, так как исполняемый файл был перекомпилирован и не имеет запретов. Для выполнения задания нахожу свой вариант в соответствии с вариантом в лабораторной работе №7(рис. 4.9).

Номер варианта	В символьном виде	В двоичной системе
1	--x -wx rwx	000 110 010

Рис. 4.9: Вариант №1

В соответствии с этим вариантов предоставляю права доступа в символьном виде к файлу readme-1.txt. Проверяю правильность с помощью команды ls -l(рис. 4.10).

```

arsihrcева@dk2n21 ~/work/arch-pc/lab10 $ chmod u=x readme-1.txt
arsihrcева@dk2n21 ~/work/arch-pc/lab10 $ chmod u+wx readme-1.txt
arsihrcева@dk2n21 ~/work/arch-pc/lab10 $ chmod u+rwx readme-1.txt
arsihrcева@dk2n21 ~/work/arch-pc/lab10 $ ls -l
итого 16
-rw-r--r-- 1 arsihrceva studsci 3942 дек 12 11:48 in_out.asm
-rwxr-xr-x 1 arsihrceva studsci 9164 дек 13 15:15 lab10-1
-rwxr--r-- 1 arsihrceva studsci 478 дек 13 15:14 lab10-1.asm
-rw-r--r-- 1 arsihrceva studsci 1472 дек 13 15:15 lab10-1.o
-rwxrw-rw- 1 arsihrceva studsci 0 дек 12 11:40 readme-1.txt

```

Рис. 4.10: Предоставление прав доступа в символьном виде

Также в соответствии с вариантом предоставляю права доступа в двоичном виде к файлу readme-2.txt(рис. 4.11), проверяю правильность выполнения команд(рис. 4.12).

```

arsihrcева@dk2n21 ~/work/arch-pc/lab10 $ chmod 000 readme-2.txt
arsihrcева@dk2n21 ~/work/arch-pc/lab10 $ chmod 110 readme-2.txt
arsihrcева@dk2n21 ~/work/arch-pc/lab10 $ chmod 010 readme-2.txt

```

Рис. 4.11: предоставление прав доступа в двоичном виде

```

arsihrcева@dk2n21 ~/work/arch-pc/lab10 $ ls -l
итого 16
-rw-r--r-- 1 arsihrceva studsci 3942 дек 12 11:48 in_out.asm
-rwxr-xr-x 1 arsihrceva studsci 9164 дек 13 14:14 lab10-1
-rwxr--r-- 1 arsihrceva studsci 503 дек 12 11:47 lab10-1.asm
-rw-r--r-- 1 arsihrceva studsci 1472 дек 13 14:14 lab10-1.o
-rwxr--r-- 1 arsihrceva studsci 0 дек 12 11:40 readme-1.txt
-----x--- 1 arsihrceva studsci 0 дек 12 11:40 readme-2.txt

```

Рис. 4.12: Проверка правильности выполнения команд

## 5 Самостоятельная работа

Создаю файл lab10-1.asm для выполнения задания. Ввожу в него программу (рис. 5.1), которая будет работать по следующему алгоритму: 1. Вывод приглашения “Как Вас зовут?” 2. Ввод фамилии и имени с клавиатуры 3. создание файла name.txt 4. Запись в файл сообщения “Меня зовут” 5. Запись в файл строки, введённой с клавиатуры 6. Закрытие файла

```

3 SECTION .data
4 file db 'name.txt', 0h
5 msg1 db 'Как Вас зовут? ', 0h
6 msg2 DB 'Меня зовут:', 0
7
8 SECTION .bss
9 contens resb 2550
10
11 SECTION .text
12 global _start
13 _start:
14
15 mov eax, msg1
16 call sprint
17
18 mov ecx, contens
19 mov edx, 2550
20 call sread
21
22 mov ecx, 0777o
23 mov ebx, file
24 mov eax, 8
25 int 80h
26
27 mov esi, eax
28
29 mov eax, msg2
30 call slen
31
32 mov edx, eax
33 mov ecx, msg2
34 mov ebx, esi
35 mov eax, 4
36 int 80h
37
38 mov eax, contens
39 call slen
40
41 mov edx, eax
42 mov ecx, contens
43 mov ebx, esi
44 mov eax, 4
45 int 80h
46
47 mov ebx, esi
48 mov eax, 6
49 int 80h

```

Рис. 5.1: Текст программы, работающей по алгоритму из задания

Создаю и запускаю исполняемый файл, на запрос программы ввожу свои фамилию и имя. С помощью команды `ls -l` проверяю правильность выполнения(рис. 5.2).

```
arsihrcева@dk2n21 ~/work/arch-pc/lab10 $ nasm -f elf lab10-2.asm
arsihrcева@dk2n21 ~/work/arch-pc/lab10 $ ld -m elf_i386 -o lab10-2 lab10-2.o
arsihrcева@dk2n21 ~/work/arch-pc/lab10 $ ./lab10-2
Как Вас зовут? Сырцева Анастасия
arsihrcева@dk2n21 ~/work/arch-pc/lab10 $ ls -l
итого 29
-rw-r--r-- 1 arsihrceva studsci 3942 дек 12 11:48 in_out.asm
-rwxr-xr-x 1 arsihrceva studsci 9164 дек 13 15:15 lab10-1
-rwxr--r-- 1 arsihrceva studsci 478 дек 13 15:14 lab10-1.asm
-rw-r--r-- 1 arsihrceva studsci 1472 дек 13 15:15 lab10-1.o
-rwxr-xr-x 1 arsihrceva studsci 9164 дек 13 15:40 lab10-2
-rw-r--r-- 1 arsihrceva studsci 537 дек 13 15:40 lab10-2.asm
-rw-r--r-- 1 arsihrceva studsci 1520 дек 13 15:40 lab10-2.o
-rwxr-xr-x 1 arsihrceva studsci 54 дек 13 15:40 name.txt
-rwxrw-rw- 1 arsihrceva studsci 0 дек 12 11:40 readme-1.txt
-----x--- 1 arsihrceva studsci 0 дек 12 11:40 readme-2.txt
arsihrcева@dk2n21 ~/work/arch-pc/lab10 $ cat name.txt
Меня зовут:Сырцева Анастасия
```

Рис. 5.2: Запуск программы и проверка её работы



## **6 Выводы**

Приобретены навыки написания программ для работы с файлами.