

# **Отчёт по лабораторной работе 10**

**Работа с файлами средствами Nasm**

Зиборова Вероника Николаевна НММбд-02-24

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Задание для самостоятельной работы . . . . .	10
<b>3</b>	<b>Выводы</b>	<b>12</b>
<b>4</b>	<b>Ответы на вопросы</b>	<b>13</b>

## Список иллюстраций

2.1	Программа lab10-1.asm . . . . .	7
2.2	Запуск программы lab10-1.asm . . . . .	8
2.3	файл без возможности запуска . . . . .	8
2.4	файл asm с разрешением на выполнение . . . . .	9
2.5	установка прав . . . . .	9
2.6	Программа lab10-2.asm . . . . .	11
2.7	Запуск программы lab10-2.asm . . . . .	11

## Список таблиц

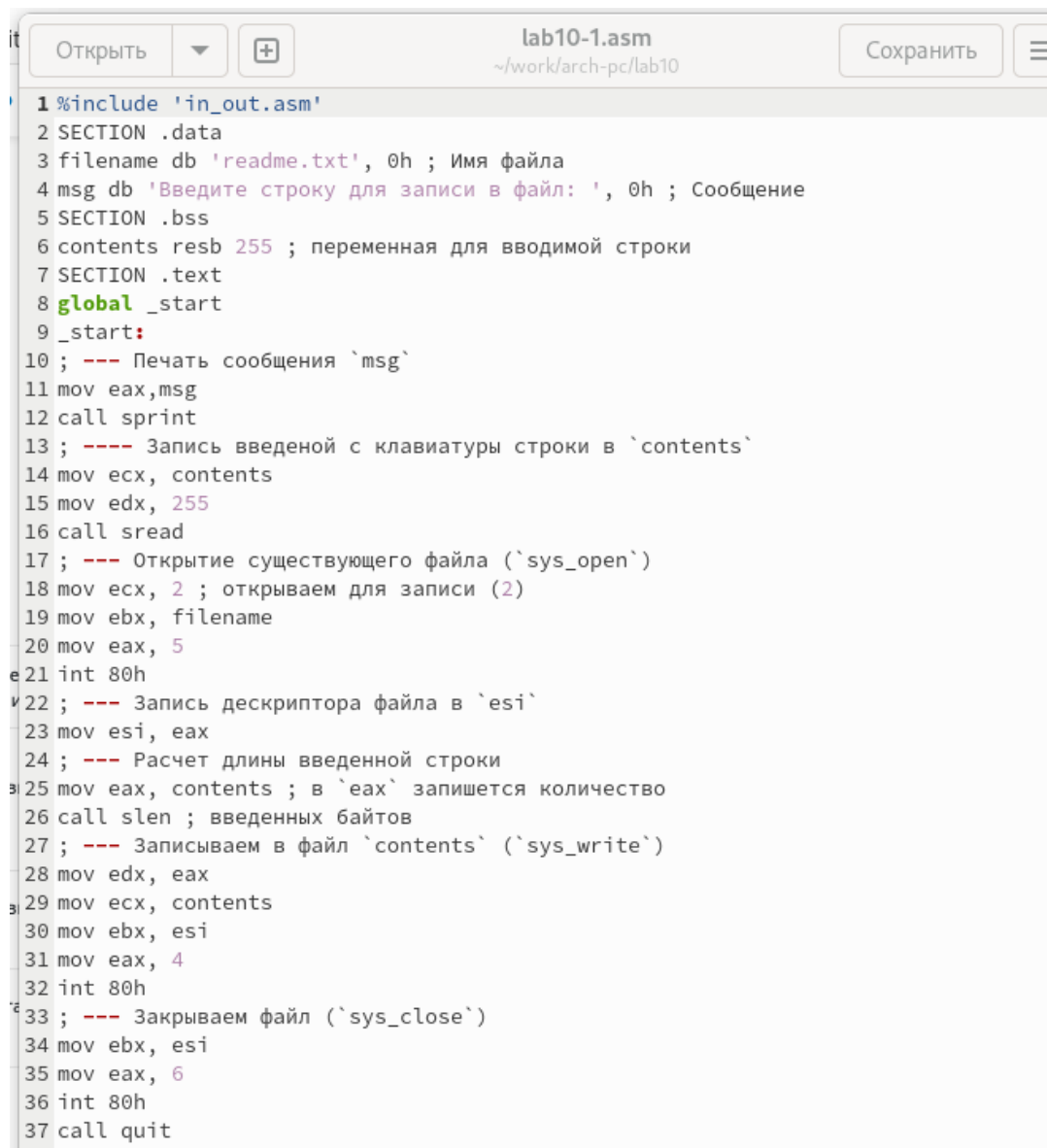
# 1 Цель работы

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

## 2 Выполнение лабораторной работы

Я создала каталог для программ лабораторной работы № 10, перешла в него и создала файлы `lab10-1.asm`, `readme-1.txt` и `readme-2.txt`.

В файл `lab10-1.asm` я записала текст программы из листинга 10.1 (Программа записи в файл сообщения). Затем создала исполняемый файл и проверила его работу.



```
1 %include 'in_out.asm'
2 SECTION .data
3 filename db 'readme.txt', 0h ; Имя файла
4 msg db 'Введите строку для записи в файл: ', 0h ; Сообщение
5 SECTION .bss
6 contents resb 255 ; переменная для вводимой строки
7 SECTION .text
8 global _start
9 _start:
10 ; --- Печать сообщения `msg`
11 mov eax,msg
12 call sprint
13 ; ---- Запись введенной с клавиатуры строки в `contents`
14 mov ecx, contents
15 mov edx, 255
16 call sread
17 ; --- Открытие существующего файла (`sys_open`)
18 mov ecx, 2 ; открываем для записи (2)
19 mov ebx, filename
20 mov eax, 5
21 int 80h
22 ; --- Запись дескриптора файла в `esi`
23 mov esi, eax
24 ; ---- Расчет длины введенной строки
25 mov eax, contents ; в `eax` запишется количество
26 call slen ; введенных байтов
27 ; --- Записываем в файл `contents` (`sys_write`)
28 mov edx, eax
29 mov ecx, contents
30 mov ebx, esi
31 mov eax, 4
32 int 80h
33 ; --- Закрываем файл (`sys_close`)
34 mov ebx, esi
35 mov eax, 6
36 int 80h
37 call quit
```

Рис. 2.1: Программа lab10-1.asm

Программа запрашивает строку и перезаписывает её в файл `readme.txt`. Если файл не существует, строка не записывается.

```

vnziborova@fedora:~/work/arch-pc/lab10$ nasm -f elf lab10-1.asm
vnziborova@fedora:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-1 lab10-1.o
vnziborova@fedora:~/work/arch-pc/lab10$ ./lab10-1
Введите строку для записи в файл: test
vnziborova@fedora:~/work/arch-pc/lab10$ cat readme.txt
test
vnziborova@fedora:~/work/arch-pc/lab10$
vnziborova@fedora:~/work/arch-pc/lab10$

```

Рис. 2.2: Запуск программы lab10-1.asm

С помощью команды `chmod` я изменила права доступа к исполняемому файлу `lab10-1`, запретив его выполнение. Попыталась выполнить файл.

Файл не запускается, так как выполнение запрещено, атрибут “x” был снят во всех трех позициях.

```

vnziborova@fedora:~/work/arch-pc/lab10$ nasm -f elf lab10-1.asm
vnziborova@fedora:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-1 lab10-1.o
vnziborova@fedora:~/work/arch-pc/lab10$ ./lab10-1
Введите строку для записи в файл: test
vnziborova@fedora:~/work/arch-pc/lab10$ cat readme.txt
test
vnziborova@fedora:~/work/arch-pc/lab10$
vnziborova@fedora:~/work/arch-pc/lab10$
vnziborova@fedora:~/work/arch-pc/lab10$ chmod -x lab10-1
vnziborova@fedora:~/work/arch-pc/lab10$ ./lab10-1
bash: ./lab10-1: Отказано в доступе
vnziborova@fedora:~/work/arch-pc/lab10$
vnziborova@fedora:~/work/arch-pc/lab10$

```

Рис. 2.3: файл без возможности запуска

С помощью команды `chmod` я изменила права доступа к файлу `lab10-1.asm`, добавив права на выполнение. Попыталась снова выполнить его.

Файл запускается, и терминал пытается выполнить его содержимое как консольные команды. Однако инструкции ассемблера не являются командами терминала, поэтому возникли ошибки. Тем не менее, если в такой файл написать команды терминала, его можно будет выполнить.



```

vnziborova@fedora:~/work/arch-pc/lab10$
vnziborova@fedora:~/work/arch-pc/lab10$ chmod +x lab10-1.asm
vnziborova@fedora:~/work/arch-pc/lab10$ ./lab10-1.asm
./lab10-1.asm: строка 1: fg: нет управления заданиями
./lab10-1.asm: строка 2: SECTION: команда не найдена
./lab10-1.asm: строка 3: filename: команда не найдена
./lab10-1.asm: строка 3: Имя: команда не найдена
./lab10-1.asm: строка 4: msg: команда не найдена
./lab10-1.asm: строка 4: Сообщение: команда не найдена
./lab10-1.asm: строка 5: SECTION: команда не найдена
./lab10-1.asm: строка 6: contents: команда не найдена
./lab10-1.asm: строка 6: переменная: команда не найдена
./lab10-1.asm: строка 7: SECTION: команда не найдена
./lab10-1.asm: строка 8: global: команда не найдена
./lab10-1.asm: строка 9: _start:: команда не найдена
./lab10-1.asm: строка 10: синтаксическая ошибка рядом с неожиданным маркером «;»
./lab10-1.asm: строка 10: `; --- Печать сообщения `msg`
vnziborova@fedora:~/work/arch-pc/lab10$

```

Рис. 2.4: файл asm с разрешением на выполнение

Предоставила права доступа к файлам readme в соответствии с вариантом из таблицы 10.4. Проверила правильность выполнения с помощью команды `ls -l`.

Для варианта 6: `-w- r-x -w-` и `011 001 111`

```

vnziborova@fedora:~/work/arch-pc/lab10$
vnziborova@fedora:~/work/arch-pc/lab10$ ls -l
итого 28
-rw-----. 1 vnziborova vnziborova 3773 дек  4 15:38 in_out.asm
-rw-r--r--. 1 vnziborova vnziborova 9164 дек  4 18:05 lab10-1
-rwx--x--x. 1 vnziborova vnziborova 1140 дек  4 15:38 lab10-1.asm
-rw-r--r--. 1 vnziborova vnziborova 1472 дек  4 18:05 lab10-1.o
-rw-----. 1 vnziborova vnziborova   0 дек  4 15:38 readme-2.txt
-rw-----. 1 vnziborova vnziborova   5 дек  4 18:06 readme.txt
vnziborova@fedora:~/work/arch-pc/lab10$ chmod 252 readme.txt
vnziborova@fedora:~/work/arch-pc/lab10$ chmod 317 readme-2.txt
vnziborova@fedora:~/work/arch-pc/lab10$ ls -l
итого 28
-rw-----. 1 vnziborova vnziborova 3773 дек  4 15:38 in_out.asm
-rw-r--r--. 1 vnziborova vnziborova 9164 дек  4 18:05 lab10-1
-rwx--x--x. 1 vnziborova vnziborova 1140 дек  4 15:38 lab10-1.asm
-rw-r--r--. 1 vnziborova vnziborova 1472 дек  4 18:05 lab10-1.o
--wx--xrw-. 1 vnziborova vnziborova   0 дек  4 15:38 readme-2.txt
--w-r-x-w-. 1 vnziborova vnziborova   5 дек  4 18:06 readme.txt
vnziborova@fedora:~/work/arch-pc/lab10$

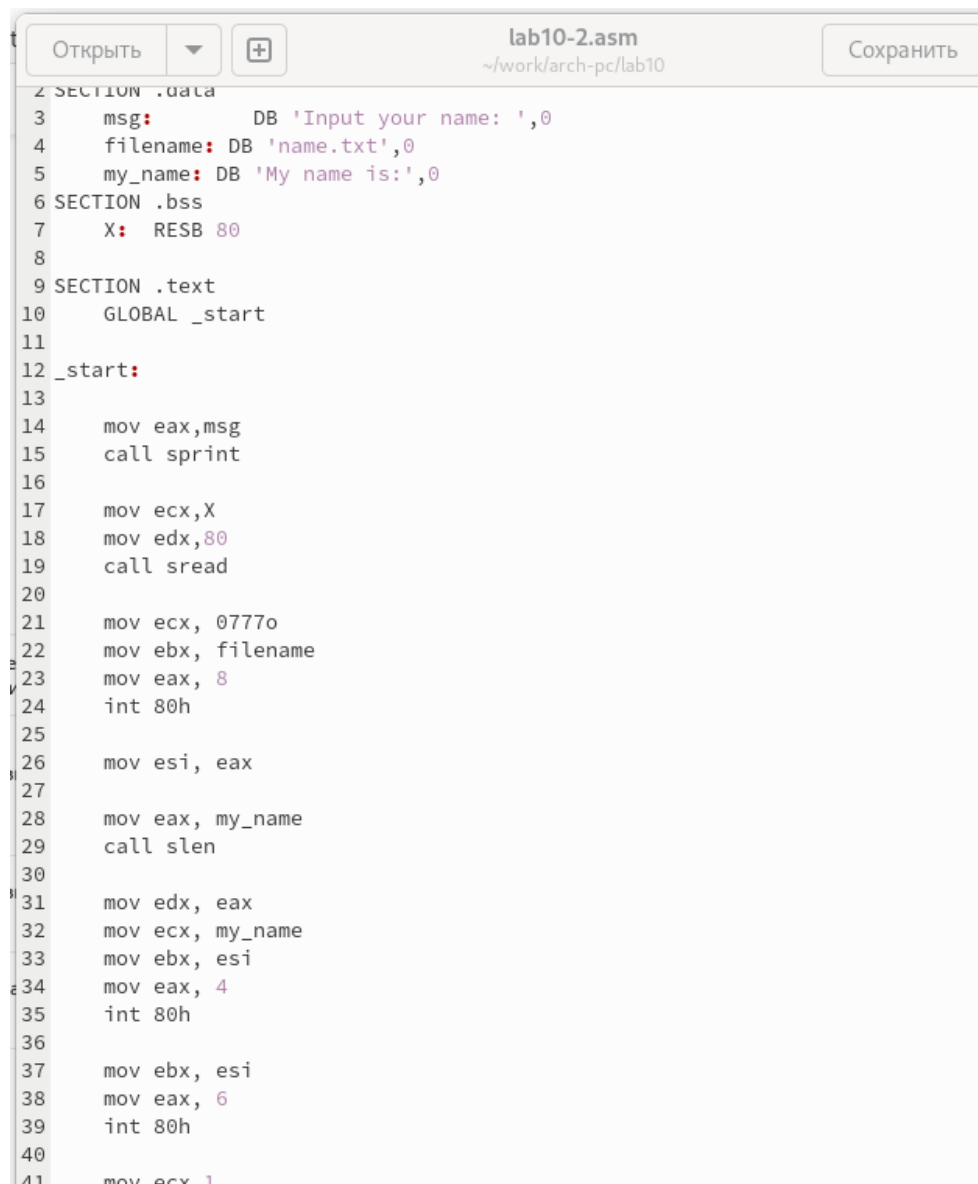
```

Рис. 2.5: установка прав

## 2.1 Задание для самостоятельной работы

Написала программу, работающую по следующему алгоритму:

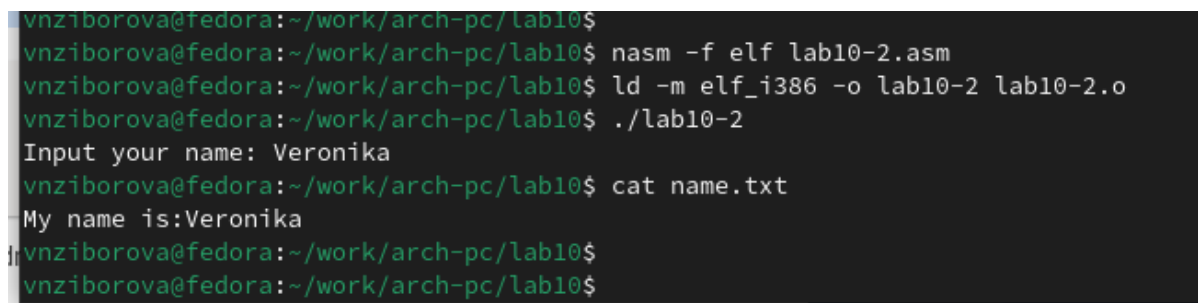
- Выводится приглашение: “Как Вас зовут?”
- Вводятся с клавиатуры фамилия и имя
- Создается файл с именем `name.txt`
- Записывается в файл сообщение “Меня зовут”
- Дописывается в файл строка, введенная с клавиатуры
- Файл закрывается



```
lab10-2.asm
~/work/arch-pc/lab10

2 SECTION .data
3     msg:      DB 'Input your name: ',0
4     filename: DB 'name.txt',0
5     my_name:  DB 'My name is:',0
6 SECTION .bss
7     X:  RESB 80
8
9 SECTION .text
10     GLOBAL _start
11
12 _start:
13
14     mov eax,msg
15     call sprint
16
17     mov ecx,X
18     mov edx,80
19     call sread
20
21     mov ecx, 0777o
22     mov ebx, filename
23     mov eax, 8
24     int 80h
25
26     mov esi, eax
27
28     mov eax, my_name
29     call slen
30
31     mov edx, eax
32     mov ecx, my_name
33     mov ebx, esi
34     mov eax, 4
35     int 80h
36
37     mov ebx, esi
38     mov eax, 6
39     int 80h
40
41     mov ecx, 1
```

Рис. 2.6: Программа lab10-2.asm



```
vnziborova@fedora:~/work/arch-pc/lab10$
vnziborova@fedora:~/work/arch-pc/lab10$ nasm -f elf lab10-2.asm
vnziborova@fedora:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-2 lab10-2.o
vnziborova@fedora:~/work/arch-pc/lab10$ ./lab10-2
Input your name: Veronika
vnziborova@fedora:~/work/arch-pc/lab10$ cat name.txt
My name is:Veronika
vnziborova@fedora:~/work/arch-pc/lab10$
vnziborova@fedora:~/work/arch-pc/lab10$
```

Рис. 2.7: Запуск программы lab10-2.asm

## **3 Выводы**

Я освоила работу с файлами и правами доступа.

## 4 Ответы на вопросы

### 1. Каким образом в Unix-подобных ОС определяются права доступа к файлу?

В Unix-подобных операционных системах права доступа к файлу определяются с помощью трёх типов разрешений для трёх категорий пользователей. Эти разрешения включают:

- **Чтение (r)**: разрешает читать содержимое файла.
- **Запись (w)**: разрешает изменять содержимое файла.
- **Исполнение (x)**: разрешает запускать файл как программу.

Права доступа определяются для трёх категорий пользователей:

- **Владелец (user)**: пользователь, который создал файл.
- **Группа (group)**: группа пользователей, к которой принадлежит владелец.
- **Остальные (others)**: все остальные пользователи.

### 2. Как ОС определяет, является ли файл исполняемым? Как регулировать права на чтение и запись?

ОС определяет, что файл является исполняемым, по наличию права “x” (исполнение) для соответствующей категории пользователей. Это означает, что файл можно запустить как программу.

Для регулирования прав на чтение и запись используются команды `chmod`:

- Для добавления права на чтение: `chmod +r <файл>`

- Для добавления права на запись: `chmod +w <файл>`
- Для добавления права на исполнение: `chmod +x <файл>`

Уровни доступа могут быть настроены для владельца, группы и других пользователей.

### 3. Как разграничить права доступа для различных категорий пользователей?

Права доступа для различных категорий пользователей разграничиваются с помощью команд `chmod`, где можно указать, какие права имеют владелец, группа и другие пользователи:

- Владелец: `u` (user)
- Группа: `g` (group)
- Остальные пользователи: `o` (others)

Например:

- `chmod u+x <файл>` — добавляет право на исполнение для владельца.
- `chmod g-w <файл>` — убирает право на запись для группы.
- `chmod o=r <файл>` — даёт право на чтение только для остальных пользователей.

### 4. Какой номер имеют системные вызовы `sys_read`, `sys_write`, `sys_open`, `sys_close`, `sys_creat`?

Номера системных вызовов в Linux (для x86) следующие:

- **`sys_read`** — номер 0
- **`sys_write`** — номер 1
- **`sys_open`** — номер 5
- **`sys_close`** — номер 6
- **`sys_creat`** — номер 8

**5. Какие регистры и как используют системные вызовы `sys_read`, `sys_write`, `sys_open`, `sys_close`, `sys_creat`?**

В Linux для архитектуры x86, регистры используются следующим образом:

- **`sys_read`:**
  - `eax`: номер системного вызова (0)
  - `ebx`: дескриптор файла
  - `ecx`: указатель на буфер для чтения
  - `edx`: количество байт для чтения
- **`sys_write`:**
  - `eax`: номер системного вызова (1)
  - `ebx`: дескриптор файла
  - `ecx`: указатель на данные для записи
  - `edx`: количество байт для записи
- **`sys_open`:**
  - `eax`: номер системного вызова (5)
  - `ebx`: имя файла
  - `ecx`: флаги (например, `O_RDONLY`, `O_WRONLY`)
  - `edx`: режим доступа (например, 0666)
- **`sys_close`:**
  - `eax`: номер системного вызова (6)
  - `ebx`: дескриптор файла
- **`sys_creat`:**
  - `eax`: номер системного вызова (8)
  - `ebx`: имя файла
  - `ecx`: режим доступа

**6. Что такое дескриптор файла?**

Дескриптор файла — это целое число, которое используется операционной системой для обозначения открытого файла в процессе. Это индекс в

таблице открытых файлов, которая хранит информацию о каждом открытом файле. Дескриптор файла предоставляется при успешном открытии файла с помощью системных вызовов, таких как `open`, и используется для дальнейших операций с файлом (чтение, запись, закрытие и т.д.).