

# **Отчёт по лабораторной работе №10**

**Дисциплина: Архитектура компьютера**

Ларина Наталья Денисовна

# 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>9</b>
4.1	Написание программ для работы с файлами . . . . .	9
4.2	Задание для самостоятельной работы . . . . .	11
<b>5</b>	<b>Выводы</b>	<b>13</b>
<b>6</b>	<b>Список литературы</b>	<b>14</b>

## **Список таблиц**

## Список иллюстраций

4.1	Создание файлов для лабораторной работы . . . . .	9
4.2	Ввод текста программы из листинга 10.1 . . . . .	10
4.3	Запуск исполняемого файла . . . . .	10
4.4	Запрет на выполнение файла . . . . .	10
4.5	Добавление прав на исполнение . . . . .	11
4.6	Предоставление прав доступа в символьном и двоичном виде . .	11
4.7	Написание текста программы . . . . .	12
4.8	Запуск исполняемого файла и проверка его работы . . . . .	12

# 1 Цель работы

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

## 2 Задание

1. Написание программ для работы с файлами.
2. Задание для самостоятельной работы.

### 3 Теоретическое введение

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.

Для изменения прав доступа служит команда `chmod`, которая понимает как символьное, так и числовое указание прав.

Обработка файлов в операционной системе Linux осуществляется за счет использования определенных системных вызовов. Для корректной работы и доступа к файлу при его открытии или создании, файлу присваивается уникальный номер (16-битное целое число) – дескриптор файла.

Для создания и открытия файла служит системный вызов `sys_creat`, который использует следующие аргументы: права доступа к файлу в регистре `ECX`, имя файла в `EBX` и номер системного вызова `sys_creat` (8) в `EAX`.

Для открытия существующего файла служит системный вызов `sys_open`, который использует следующие аргументы: права доступа к файлу в регистре `EDX`, режим доступа к файлу в регистр `ECX`, имя файла в `EBX` и номер системного вызова `sys_open` (5) в `EAX`.

Для записи в файл служит системный вызов `sys_write`, который использует следующие аргументы: количество байтов для записи в регистре `EDX`, строку содержимого для записи `ECX`, файловый дескриптор в `EBX` и номер системного вызова `sys_write` (4) в `EAX`. Системный вызов возвращает фактическое количество

записанных байтов в регистр EAX. В случае ошибки, код ошибки также будет находиться в регистре EAX. Прежде чем записывать в файл, его необходимо создать или открыть, что позволит получить дескриптор файла.

Для чтения данных из файла служит системный вызов `sys_read`, который использует следующие аргументы: количество байтов для чтения в регистре EDI, адрес в памяти для записи прочитанных данных в ECX, файловый дескриптор в EBX и номер системного вызова `sys_read` (3) в EAX. Как и для записи, прежде чем читать из файла, его необходимо открыть, что позволит получить дескриптор файла.

Для правильного закрытия файла служит системный вызов `sys_close`, который использует один аргумент – дескриптор файла в регистре EBX. После вызова ядра происходит удаление дескриптора файла, а в случае ошибки, системный вызов возвращает код ошибки в регистр EAX.

Для изменения содержимого файла служит системный вызов `sys_lseek`, который использует следующие аргументы: исходная позиция для смещения EDI, значение смещения в байтах в ECX, файловый дескриптор в EBX и номер системного вызова `sys_lseek` (19) в EAX. Значение смещения можно задавать в байтах.

Удаление файла осуществляется системным вызовом `sys_unlink`, который использует один аргумент – имя файла в регистре EBX.



## 4 Выполнение лабораторной работы

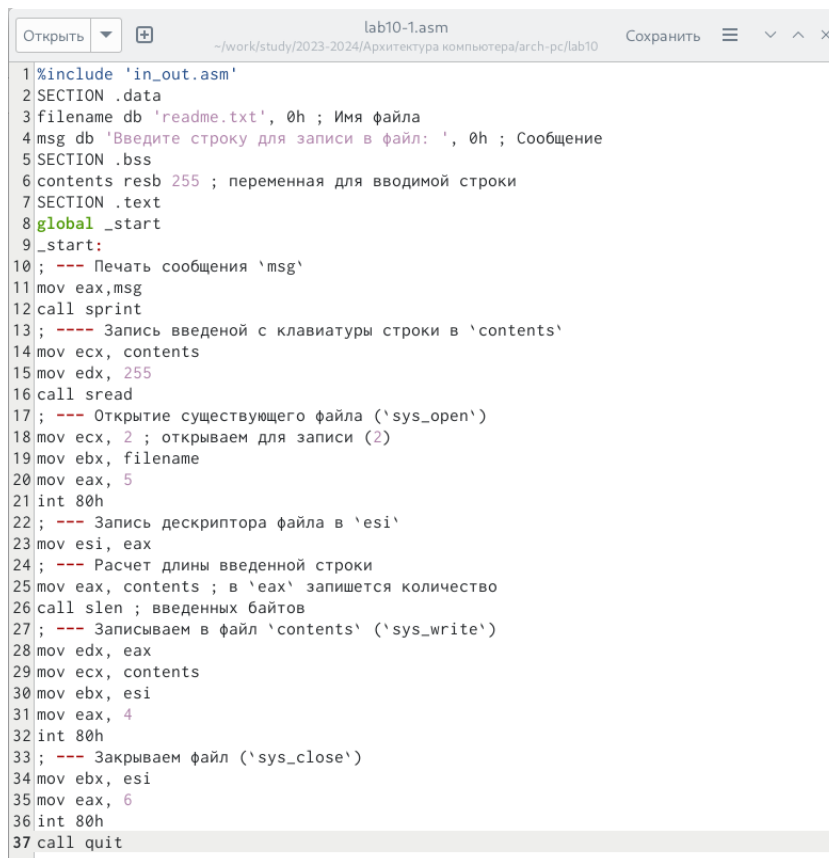
### 4.1 Написание программ для работы с файлами

Создаю каталог для программ лабораторной работы № 10, перехожу в него и создаю файлы lab10-1.asm, readme-1.txt и readme-2.txt. (рис. 4.1)

```
ndlarina@dk6n58 ~$ mkdir ~/work/study/2023-2024/Архитектура\ компьютера/arch-pc/lab10
ndlarina@dk6n58 ~$ cd ~/work/study/2023-2024/Архитектура\ компьютера/arch-pc/lab10
ndlarina@dk6n58 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10$ touch lab10-1.asm readme-1.txt readme-2.txt
```

Рис. 4.1: Создание файлов для лабораторной работы

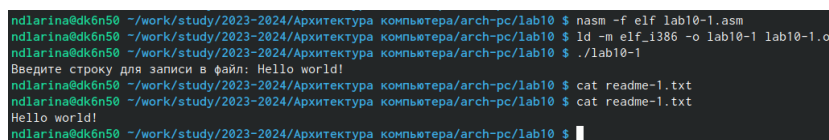
Ввожу в файл lab10-1.asm текст программы, записывающей в файл сообщения, из листинга 10.1. (рис. 4.2)



```
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. 4.2: Ввод текста программы из листинга 10.1

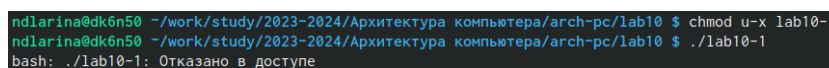
Создаю исполняемый файл и проверяю его работу. (рис. 4.3)



```
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ nasm -f elf lab10-1.asm
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ ld -m elf_i386 -o lab10-1 lab10-1.o
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ ./lab10-1
Введите строку для записи в файл: Hello world!
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ cat readme-1.txt
Hello world!
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $
```

Рис. 4.3: Запуск исполняемого файла

Далее с помощью команды `chmod` у-х изменяю права доступа к исполняемому файлу `lab10-1`, запретив его выполнение и пытаюсь выполнить файл. (рис. 4.4)



```
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ chmod u-x lab10-1
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ ./lab10-1
bash: ./lab10-1: Отказано в доступе
```

Рис. 4.4: Запрет на выполнение файла

Файл не выполняется, т.к в команде я указала “u” - владелец (себя), “-” - отменить набор прав, “x” - право на исполнение.

С помощью команды `chmod u+x` изменяю права доступа к файлу `lab10-1.asm` с исходным текстом программы, добавив права на исполнение, и пытаюсь выполнить его. (рис. 4.5)

```
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ chmod u+x lab10-1.asm
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ ./lab10-1.asm
./lab10-1.asm: строка 1: синтаксическая ошибка рядом с неожиданным маркером «;»
./lab10-1.asm: строка 1: `;-----'
```

Рис. 4.5: Добавление прав на исполнение

Текстовый файл начинает исполнение, но не исполняется, т.к не содержит в себе команд для терминала.

В соответствии со своим вариантом (6) в таблице 10.4 предоставляю права доступа к файлу `readme1.txt` представленные в символьном виде, а для файла `readme-2.txt` – в двоичном виде:

`-w- r-x -w- 011 001 111`

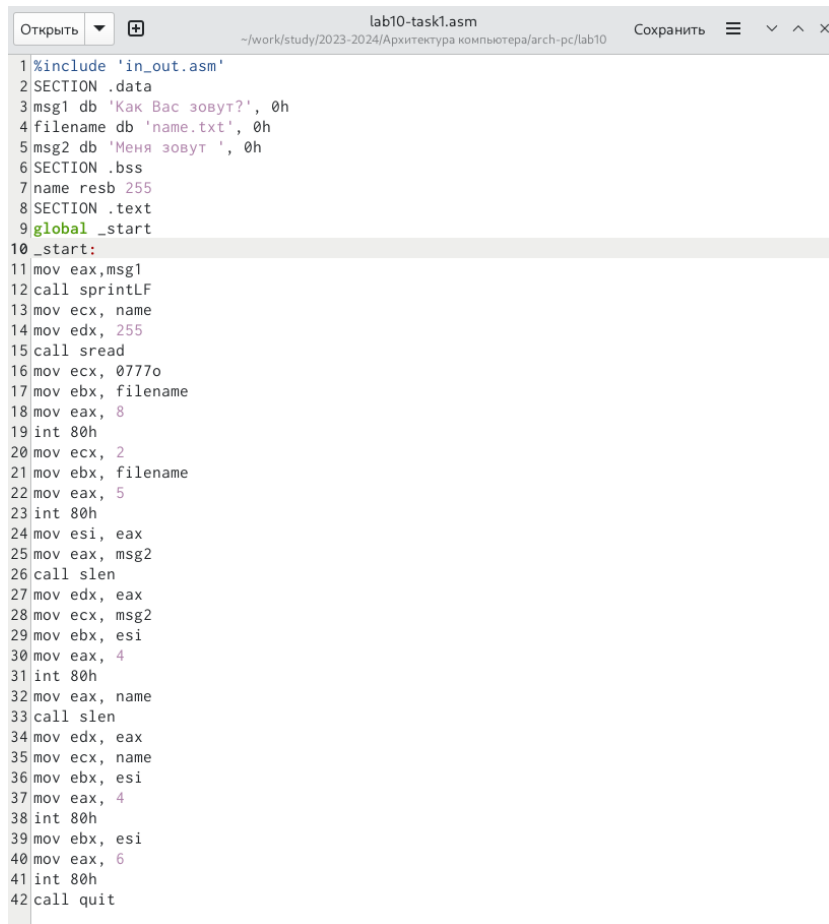
И проверяю правильность выполнения с помощью команды `ls -l`. (рис. 4.6)

```
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ chmod 640 readme-1.txt # -w- r-x -w-
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ chmod 640 readme-1.txt # 011 001 111
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ ls -l
итого 18
-rw-r--r-- 1 ndlarina studsci 3942 дек  4 13:50 in_out.asm
-rw-r--r-x 1 ndlarina studsci 9164 дек 13 13:50 lab10-1
-rwxr--r-- 1 ndlarina studsci 1289 дек 13 13:53 lab10-1.asm
-rw-r--r-- 1 ndlarina studsci 1472 дек 13 13:50 lab10-1.o
-rw-r----- 1 ndlarina studsci 13 дек 13 13:51 readme-1.txt
-rw-r--r-- 1 ndlarina studsci 0 дек 13 13:43 readme-2.txt
```

Рис. 4.6: Предоставление прав доступа в символьном и двоичном виде

## 4.2 Задание для самостоятельной работы

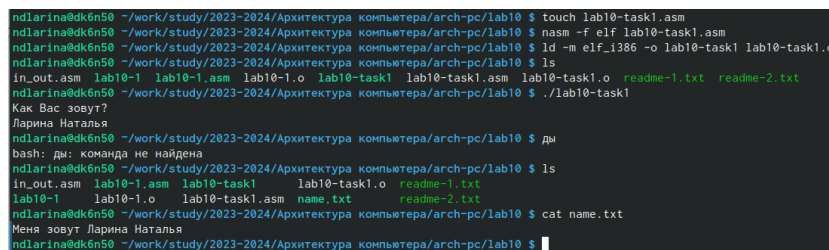
Пишу код программы, выводящей приглашения “Как Вас зовут?”, считывающей с клавиатуры фамилию и имя и создающую файл, в который записывается сообщение “Меня зовут”ФИ””. (рис. 4.7)



```
1 %include 'in_out.asm'
2 SECTION .data
3 msg1 db 'Как Вас зовут?', 0h
4 filename db 'name.txt', 0h
5 msg2 db 'Меня зовут ', 0h
6 SECTION .bss
7 name resb 255
8 SECTION .text
9 global _start
10 _start:
11 mov eax, msg1
12 call sprintf
13 mov ecx, name
14 mov edx, 255
15 call sread
16 mov ecx, 0777o
17 mov ebx, filename
18 mov eax, 8
19 int 80h
20 mov ecx, 2
21 mov ebx, filename
22 mov eax, 5
23 int 80h
24 mov esi, eax
25 mov eax, msg2
26 call slen
27 mov edx, eax
28 mov ecx, msg2
29 mov ebx, esi
30 mov eax, 4
31 int 80h
32 mov eax, name
33 call slen
34 mov edx, eax
35 mov ecx, name
36 mov ebx, esi
37 mov eax, 4
38 int 80h
39 mov ebx, esi
40 mov eax, 6
41 int 80h
42 call quit
```

Рис. 4.7: Написание текста программы

Создаю исполняемый файл и проверяю его работу. Проверяю наличие файла и его содержимое с помощью команд `ls` и `cat`. (рис. 4.8)



```
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ touch lab10-task1.asm
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ nasm -f elf lab10-task1.asm
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ ld -m elf_i386 -o lab10-task1 lab10-task1.o
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ ls
in_out.asm lab10-1 lab10-1.asm lab10-1.o lab10-task1 lab10-task1.asm lab10-task1.o readme-1.txt readme-2.txt
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ ./lab10-task1
Как Вас зовут?
Ларина Наталья
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ dy
bash: dy: команда не найдена
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ ls
in_out.asm lab10-1.asm lab10-task1 lab10-task1.o readme-1.txt
lab10-1 lab10-1.o lab10-task1.asm name.txt readme-2.txt
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $ cat name.txt
Меня зовут Ларина Наталья
ndlarina@dk6n50 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab10 $
```

Рис. 4.8: Запуск исполняемого файла и проверка его работы

## **5 Выводы**

В ходе работы над данной лабораторной работе мне удалось приобрести навыки написания программ для работы с файлами.

## **6 Список литературы**

1. Лабораторная работа №10