

Лабораторная работа 10

Работа с файлами средствами Nasm

Головина Мария Игоревна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	8
4	Выполнение лабораторной работы	11
5	Выводы	19
	Список литературы	20

Список иллюстраций

4.1	Создание каталога и файлов для выполнения лабораторной работы	11
4.2	Листинг 1	12
4.3	Результаты работы программы из листинга 1 методического указания	13
4.4	Содержание файла readme.txt	13
4.5	Изменение прав доступа к файлу	13
4.6	Изменение прав доступа к файлу lab10-1.asm	14
4.7	Права доступа для файла readme-1.txt	15
4.8	Права доступа для файла readme-2.txt	15
4.9	Листинг самостоятельного задания №1	16
4.10	Продолжение листинга самостоятельного задания №1	17
4.11	Результаты работы программы по самостоятельному заданию №1	17
4.12	Содержимое файла name.txt	18

Список таблиц

1 Цель работы

Приобретение навыков написания программ для работы с файлами.

2 Задание

1. Создать каталог для программ лабораторной работы № 10, перейти в него и создать файлы lab10-1.asm, readme.txt, readme-1.txt, readme-2.txt, readme.txt.
2. Ввести в файл lab10-1.asm текст программы из листинга 1 методического указания. Создать исполняемый файл и проверить его работу.
3. Изменить права доступа к исполняемому файлу lab10-1, запретив его выполнение.
4. С помощью команды `chmod` изменить права доступа к файлу lab10-1.asm с исходным текстом программы, добавив права на исполнение.
5. В соответствии с вариантом в таблице методического указания (мой вариант №11) 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`.

Задание для самостоятельной работы

Напишите программу, работающую по следующему алгоритму:

Вывод приглашения “Как Вас зовут?”

ввести с клавиатуры свои фамилию и имя

создать файл с именем name.txt

записать в файл сообщение “Меня зовут”

дописать в файл строку введенную с клавиатуры

закрыть файл.

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

3 Теоретическое введение

Права доступа к файлам

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, буква означает наличие права (установлен в единицу второй бит триады `г` — чтение, первый бит `w` — запись, нулевой бит `х` — исполнение), а дефис означает отсутствие права (нулевое значение соответствующего бита). Также права доступа могут быть представлены как восьмеричное число. Так, права доступа `gw-` (чтение и запись, без исполнения) понимаются как три двоичные цифры `110` или как восьмеричная цифра `6`.

Свойства (атрибуты) файлов и каталогов можно вывести на терминал с помощью команды `ls` с ключом `-l`. Тип файла определяется первой позицией, это может быть: каталог — `d`, обычный файл — дефис (`-`) или символьная ссылка на другой

файл — 1. 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 символьное, так и числовое указание прав.

Работа с файлами средствами Nasm.

В операционной системе Linux существуют различные методы управления файлами, например, такие как создание и открытие файла, только для чтения или для чтения и записи, добавления в существующий файл, закрытия и удаления файла, предоставление прав доступа. 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.

Общий алгоритм работы с системными вызовами в Nasm можно представить в следующем виде:

Поместить номер системного вызова в регистр EAX;

Поместить аргументы системного вызова в регистрах EBX, ECX и EDX;

Вызов прерывания (`int 80h`);

Результат обычно возвращается в регистр EAX.

Открытие и создание 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.

Запись в файл.

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.

Чтение файла.

Для чтения данных из файла служит системный вызов `sys_read`, который использует следующие аргументы: количество байтов для чтения в регистре EDX, 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.

Закрытие 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.

Изменение содержимого файла.

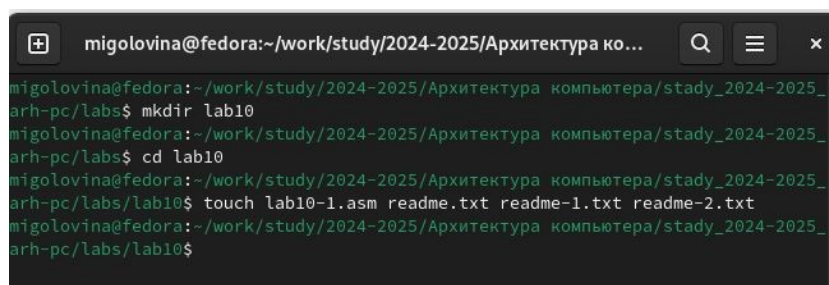
Для изменения содержимого файла служит системный вызов `sys_lseek`, который использует следующие аргументы: исходная позиция для смещения EDX, значение смещения в байтах в ECX, файловый дескриптор в EBX и номер системного вызова `sys_lseek` (19) в EAX.

Удаление файла.

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

4 Выполнение лабораторной работы

1 Создали каталог для программ лабораторной работы № 10, перешли в него и создали файлы lab10-1.asm, readme.txt, readme-1.txt, readme-2.txt, readme.txt (рис. 4.1 Создание каталога и файлов для выполнения лабораторной работы).



```
migolovina@fedora:~/work/study/2024-2025/Архитектура ко...
mih-pc/labs$ mkdir lab10
mih-pc/labs$ cd lab10
mih-pc/labs/lab10$ touch lab10-1.asm readme.txt readme-1.txt readme-2.txt
mih-pc/labs/lab10$
```

Рис. 4.1: Создание каталога и файлов для выполнения лабораторной работы

2 Ввели в файл lab10-1.asm текст программы из листинга 1 методического указания (Рис. 4.2 Листинг 1). Создали исполняемый файл и проверили его работу (рис. 4.3 Результаты работы программы из листинга 1 методического указания).

Открыть ▾  • lab10-1.asm
~/work/study/2024-20... 25_arh-pc/labs/lab10

```
%include 'in_out.asm'

SECTION .data
filename db 'readme.txt', 0h
msg db 'Введите строку для записи в файл: ', 0h

SECTION .bss

contents resb 255
SECTION .text
global _start
_start:

mov eax,msg
call sprint

mov ecx, contents
mov edx, 255
call sread

mov ecx, 2
mov ebx, filename
mov eax, 5
int 80h

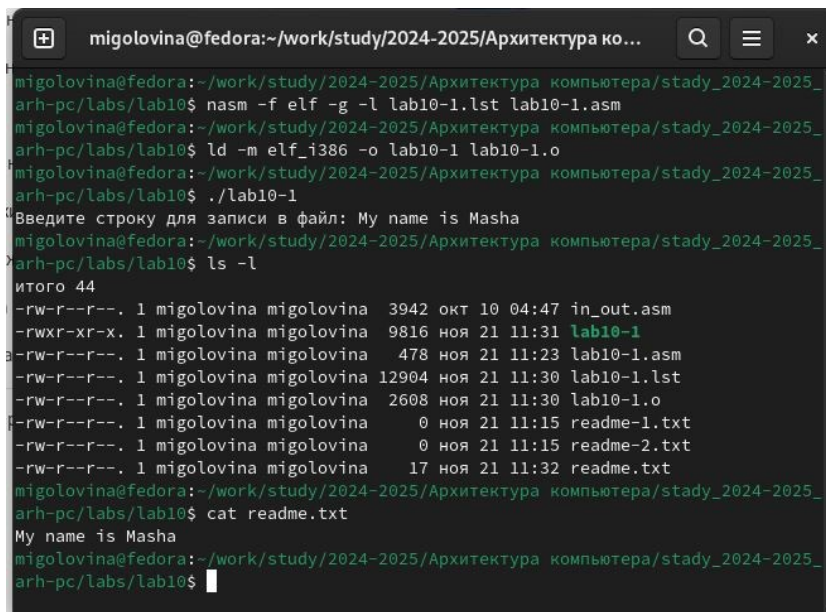
mov esi, eax

mov eax, contents
call slen

mov edx, eax
mov ecx, contents
mov ebx, esi
mov eax, 4
int 80h

mov ebx, esi
mov eax, 6
int 80h
call quit
```

Рис. 4.2: Листинг 1



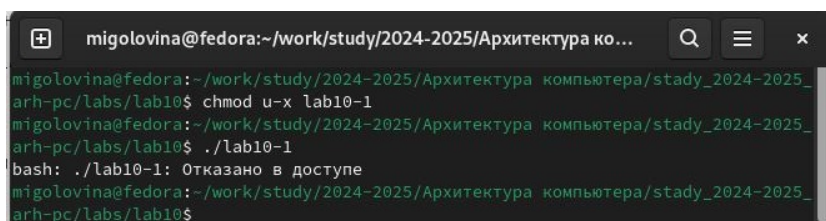
```
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ nasm -f elf -g -l lab10-1.lst lab10-1.asm
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ ld -m elf_i386 -o lab10-1 lab10-1.o
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ ./lab10-1
Введите строку для записи в файл: My name is Masha
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ ls -l
итого 44
-rw-r--r--. 1 migolovina migolovina 3942 окт 10 04:47 in_out.asm
-rwxr-xr-x. 1 migolovina migolovina 9816 ноя 21 11:31 lab10-1
-rw-r--r--. 1 migolovina migolovina 478 ноя 21 11:23 lab10-1.asm
-rw-r--r--. 1 migolovina migolovina 12904 ноя 21 11:30 lab10-1.lst
-rw-r--r--. 1 migolovina migolovina 2608 ноя 21 11:30 lab10-1.o
-rw-r--r--. 1 migolovina migolovina 0 ноя 21 11:15 readme-1.txt
-rw-r--r--. 1 migolovina migolovina 0 ноя 21 11:15 readme-2.txt
-rw-r--r--. 1 migolovina migolovina 17 ноя 21 11:32 readme.txt
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ cat readme.txt
My name is Masha
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$
```

Рис. 4.3: Результаты работы программы из листинга 1 методического указания



Рис. 4.4: Содержание файла readme.txt

3 С помощью команды `chmod` изменили права доступа к исполняемому файлу `lab10-1`, запретив его выполнение. Попытались выполнить (рис. 4.5 Изменение прав доступа к файлу).

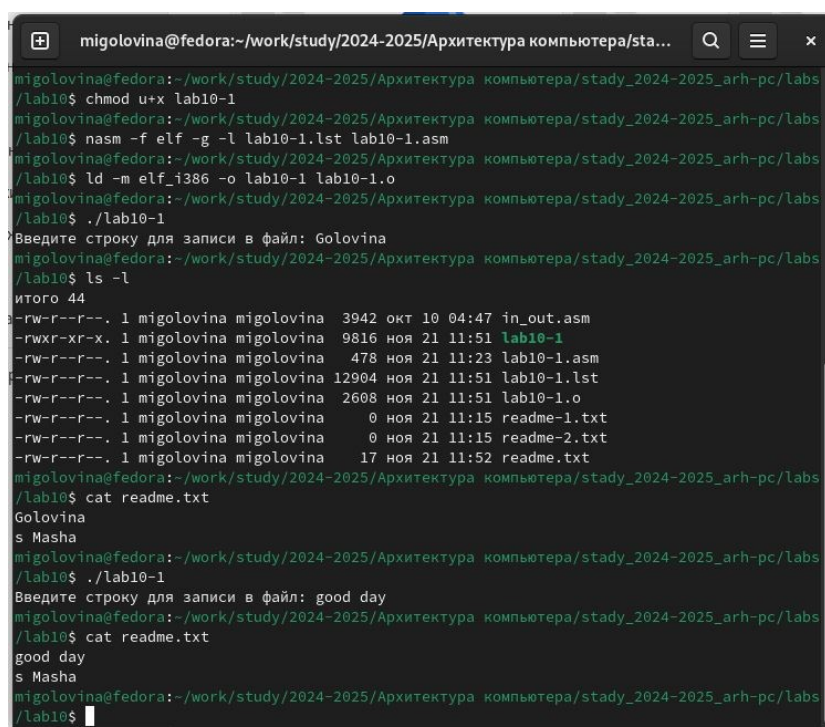


```
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ chmod u-x lab10-1
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ ./lab10-1
bash: ./lab10-1: Отказано в доступе
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$
```

Рис. 4.5: Изменение прав доступа к файлу

При попытке выполнить файл выдало сообщение, что отказано в доступе, что и следовало ожидать, так как я запретила запускать программу для владельца, то есть для самой себя.

4 С помощью команды `chmod` изменили права доступа к файлу `lab10-1.asm` с исходным текстом программы, добавив права на исполнение. Выполнили его (рис. 4.6 Изменение прав доступа к файлу `lab10-1.asm`).



```
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ chmod u+x lab10-1
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ nasm -f elf -g -l lab10-1.lst lab10-1.asm
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ ld -m elf_i386 -o lab10-1 lab10-1.o
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ ./lab10-1
Введите строку для записи в файл: Golovina
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ ls -l
итого 44
-rw-r--r--. 1 migolovina migolovina 3942 окт 10 04:47 in_out.asm
-rwxr-xr-x. 1 migolovina migolovina 9816 ноя 21 11:51 lab10-1
-rw-r--r--. 1 migolovina migolovina 478 ноя 21 11:23 lab10-1.asm
-rw-r--r--. 1 migolovina migolovina 12904 ноя 21 11:51 lab10-1.lst
-rw-r--r--. 1 migolovina migolovina 2608 ноя 21 11:51 lab10-1.o
-rw-r--r--. 1 migolovina migolovina 0 ноя 21 11:15 readme-1.txt
-rw-r--r--. 1 migolovina migolovina 0 ноя 21 11:15 readme-2.txt
-rw-r--r--. 1 migolovina migolovina 17 ноя 21 11:52 readme.txt
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ cat readme.txt
Golovina
s Masha
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ ./lab10-1
Введите строку для записи в файл: good day
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ cat readme.txt
good day
s Masha
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$
```

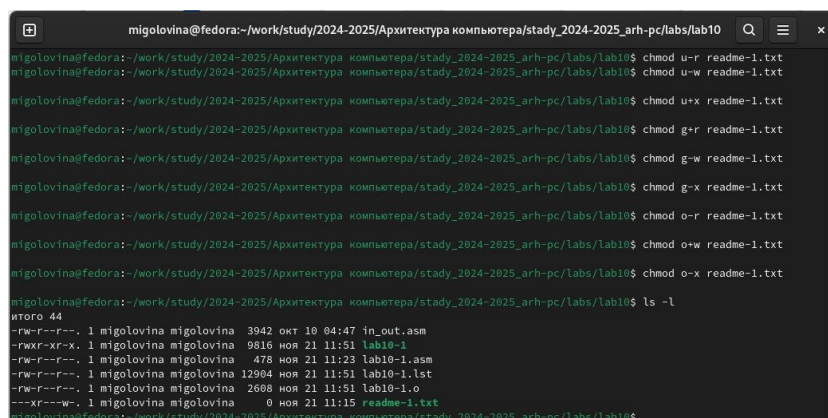
Рис. 4.6: Изменение прав доступа к файлу `lab10-1.asm`

С помощью команды я дала разрешение на исполнение файлу с исходным текстом и перекомпилировала программу, она заработала, так как файл был со всеми разрешениями. После повторной компиляции это фактически «новая программа», которая обладает другими разрешениями, поэтому они и запустилась (до этого я запретила выполняться уже готовой программе).

5 В соответствии с вариантом в таблице методического указания (мой вариант №11) предоставили права доступа к файлу `readme-1.txt` представленные в символьном виде (рис. 4.7 Права доступа для файла `readme-1.txt`), а для файла

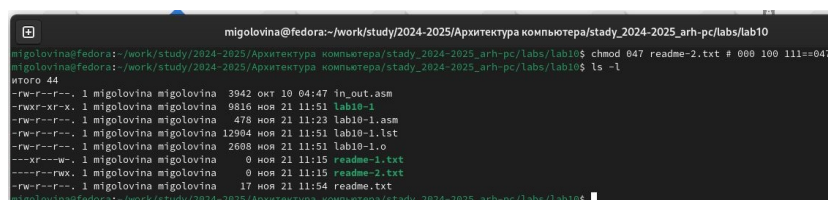
readme-2.txt – в двочном виде (рис. 4.8 Права доступа для файла readme-2.txt).

Проверили правильность выполнения с помощью команды ls -l.



```
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ chmod u-r readme-1.txt
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ chmod u-w readme-1.txt
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ chmod u+x readme-1.txt
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ chmod g+r readme-1.txt
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ chmod g-w readme-1.txt
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ chmod g-x readme-1.txt
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ chmod o-r readme-1.txt
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ chmod o-w readme-1.txt
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ chmod o-x readme-1.txt
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ ls -l
итого 44
-rw-r--r-- 1 migolovina migolovina 3942 окт 10 04:47 in_out.asm
-rwxr-xr-x 1 migolovina migolovina 9816 ноя 21 11:51 lab10-1
-rw-r--r-- 1 migolovina migolovina 478 ноя 21 11:23 lab10-1.asm
-rw-r--r-- 1 migolovina migolovina 12904 ноя 21 11:51 lab10-1.lst
-rw-r--r-- 1 migolovina migolovina 2608 ноя 21 11:51 lab10-1.o
---x---w- 1 migolovina migolovina 0 ноя 21 11:15 readme-1.txt
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$
```

Рис. 4.7: Права доступа для файла readme-1.txt



```
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ chmod 047 readme-2.txt # 000 100 111==047
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$ ls -l
итого 44
-rw-r--r-- 1 migolovina migolovina 3942 окт 10 04:47 in_out.asm
-rwxr-xr-x 1 migolovina migolovina 9816 ноя 21 11:51 lab10-1
-rw-r--r-- 1 migolovina migolovina 478 ноя 21 11:23 lab10-1.asm
-rw-r--r-- 1 migolovina migolovina 12904 ноя 21 11:51 lab10-1.lst
-rw-r--r-- 1 migolovina migolovina 2608 ноя 21 11:51 lab10-1.o
---x---w- 1 migolovina migolovina 0 ноя 21 11:15 readme-1.txt
---x---w- 1 migolovina migolovina 0 ноя 21 11:15 readme-2.txt
-rw-r--r-- 1 migolovina migolovina 17 ноя 21 11:54 readme.txt
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_arh-pc/labs/lab10$
```

Рис. 4.8: Права доступа для файла readme-2.txt

Задание для самостоятельной работы

1. Напишите программу работающую по следующему алгоритму:

Вывод приглашения “Как Вас зовут?”

ввести с клавиатуры свои фамилию и имя

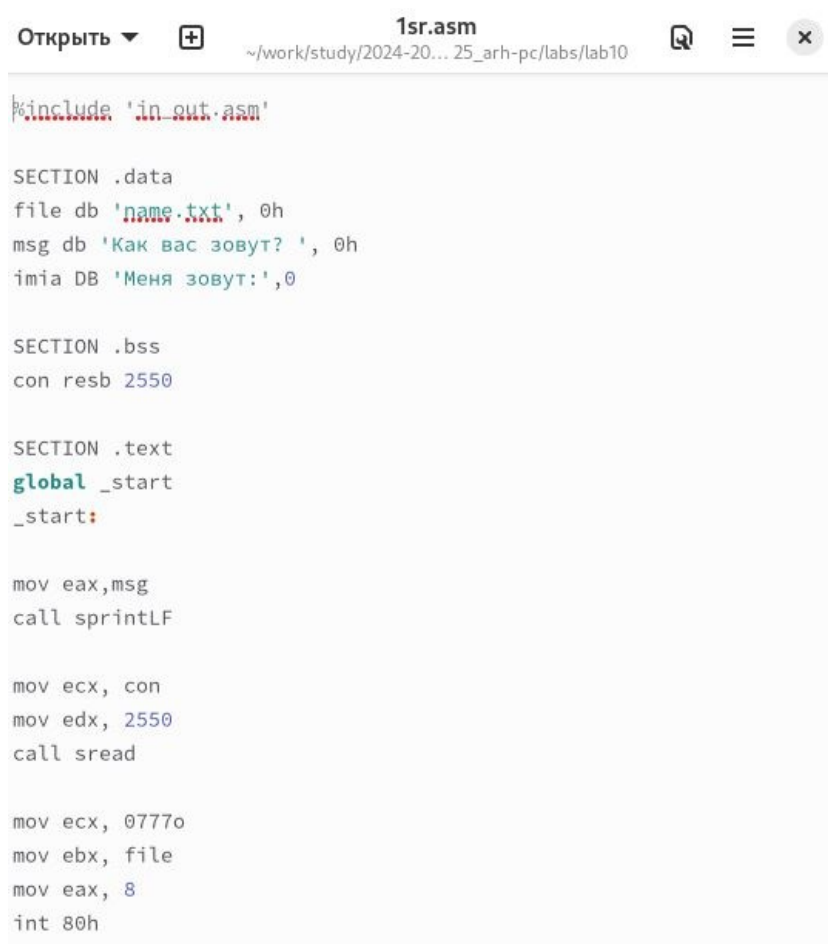
создать файл с именем name.txt

записать в файл сообщение “Меня зовут”

дописать в файл строку введенную с клавиатуры

закреть файл.

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



```
Открыть ▾ 1sr.asm ~/work/study/2024-20... 25_arh-pc/labs/lab10

%include 'in_out.asm'

SECTION .data
file db 'name.txt', 0h
msg db 'Как вас зовут? ', 0h
imia DB 'Меня зовут:',0

SECTION .bss
con resb 2550

SECTION .text
global _start
_start:

mov eax,msg
call sprintLF

mov ecx, con
mov edx, 2550
call sread

mov ecx, 0777o
mov ebx, file
mov eax, 8
int 80h
```

Рис. 4.9: Листинг самостоятельного задания №1


```

mov esi, eax

mov eax, imia
call slen

mov edx, eax
mov ecx, imia
mov ebx, esi
mov eax, 4
int 80h

mov eax, con
call slen

mov edx, eax
mov ecx, con
mov ebx, esi
mov eax, 4
int 80h

mov ebx, esi
mov eax, 6
int 80h
call quit

```

Рис. 4.10: Продолжение листинга самостоятельного задания №1

```

migolovina@fedora:~/work/study/2024-2025/Архитектура ко...
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_
arh-pc/labs/lab10$ nasm -f elf lsr.asm
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_
arh-pc/labs/lab10$ ld -m elf_i386 -o lsr lsr.o
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_
arh-pc/labs/lab10$ ./lsr
Как вас зовут?
Masha
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_
arh-pc/labs/lab10$ ls
lsr      lsr.o      lab10-1    lab10-1.lst  name.txt    readme-2.txt
lsr.asm  in_out.asm lab10-1.asm lab10-1.o    readme-1.txt readme.txt
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_
arh-pc/labs/lab10$ cat name.txt
Меня зовут:Masha
migolovina@fedora:~/work/study/2024-2025/Архитектура компьютера/stady_2024-2025_
arh-pc/labs/lab10$

```

Рис. 4.11: Результаты работы программы по самостоятельному заданию №1



Рис. 4.12: Содержимое файла name.txt

5 Выводы

Приобрели навыки написания программ для работы с файлами.

Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.

15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).