

# **Отчёт по лабораторной работе 5**

**Архитектура компьютера**

Рыбалко Николай НБИбд-02-23

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Основы работы в Midnight Commander . . . . .	6
2.2	Подключение внешнего файла in_out.asm . . . . .	10
2.3	Задание для самостоятельной работы . . . . .	15
<b>3</b>	<b>Выводы</b>	<b>20</b>

## Список иллюстраций

2.1	Создание каталога . . . . .	6
2.2	Создание файла lab05-1.asm . . . . .	7
2.3	Код программы lab05-1.asm . . . . .	8
2.4	Просмотр файла lab05-1.asm . . . . .	9
2.5	Сборка и проверка программы lab05-1.asm . . . . .	10
2.6	Копирование файла . . . . .	11
2.7	Код программы lab05-2.asm . . . . .	12
2.8	Сборка и проверка программы lab05-2.asm . . . . .	13
2.9	Код программы lab05-2.asm . . . . .	14
2.10	Сборка и проверка программы lab05-2.asm . . . . .	15
2.11	Код программы lab05-3.asm . . . . .	16
2.12	Сборка и проверка программы lab05-3.asm . . . . .	17
2.13	Код программы lab05-4.asm . . . . .	18
2.14	Сборка и проверка программы lab05-4.asm . . . . .	18

## Список таблиц

# 1 Цель работы

Целью работы является приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.

## 2 Выполнение лабораторной работы

### 2.1 Основы работы в Midnight Commander

Открыл Midnight Commander. Перешел в каталог ~/work/arch-pc. Создал каталог lab05 Указано на рис. [2.1]

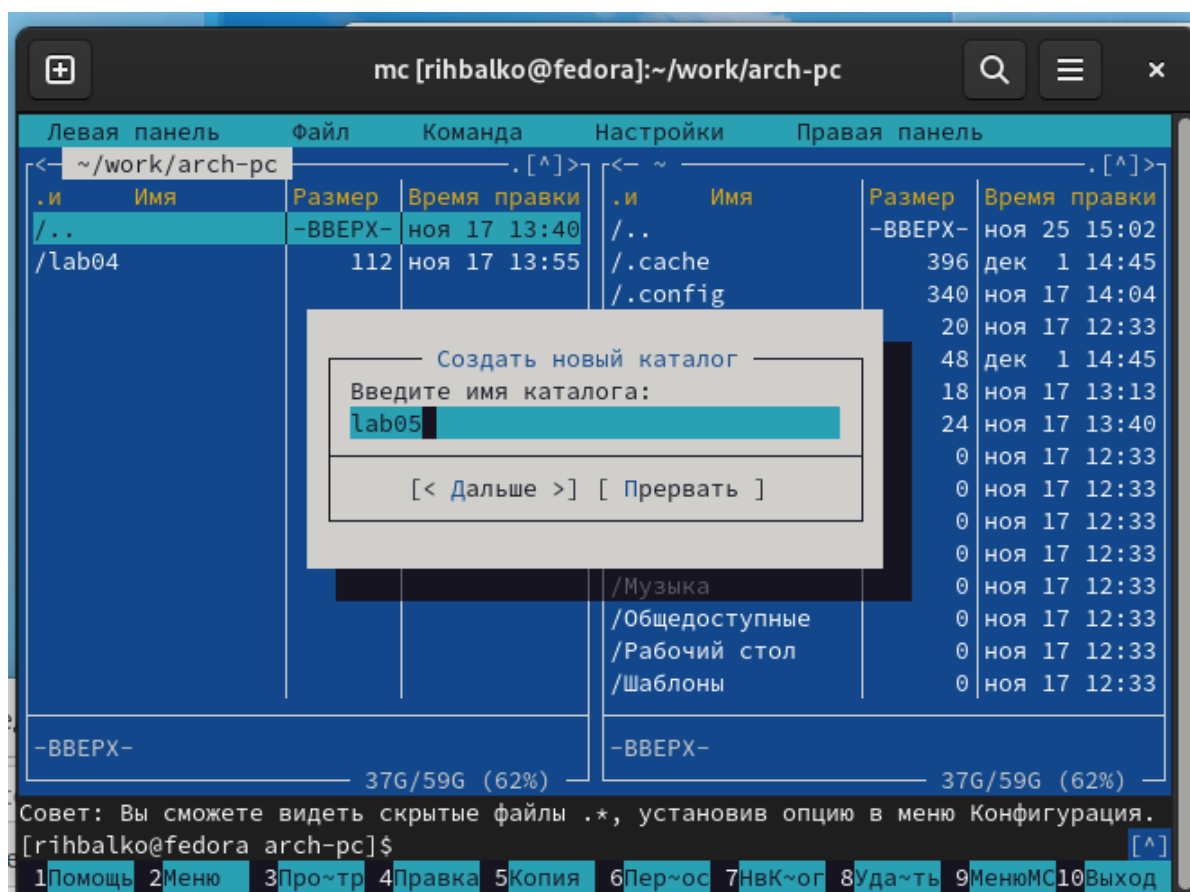


Рис. 2.1: Создание каталога

Создал файл lab05-1.asm Указано на рис. [2.2]

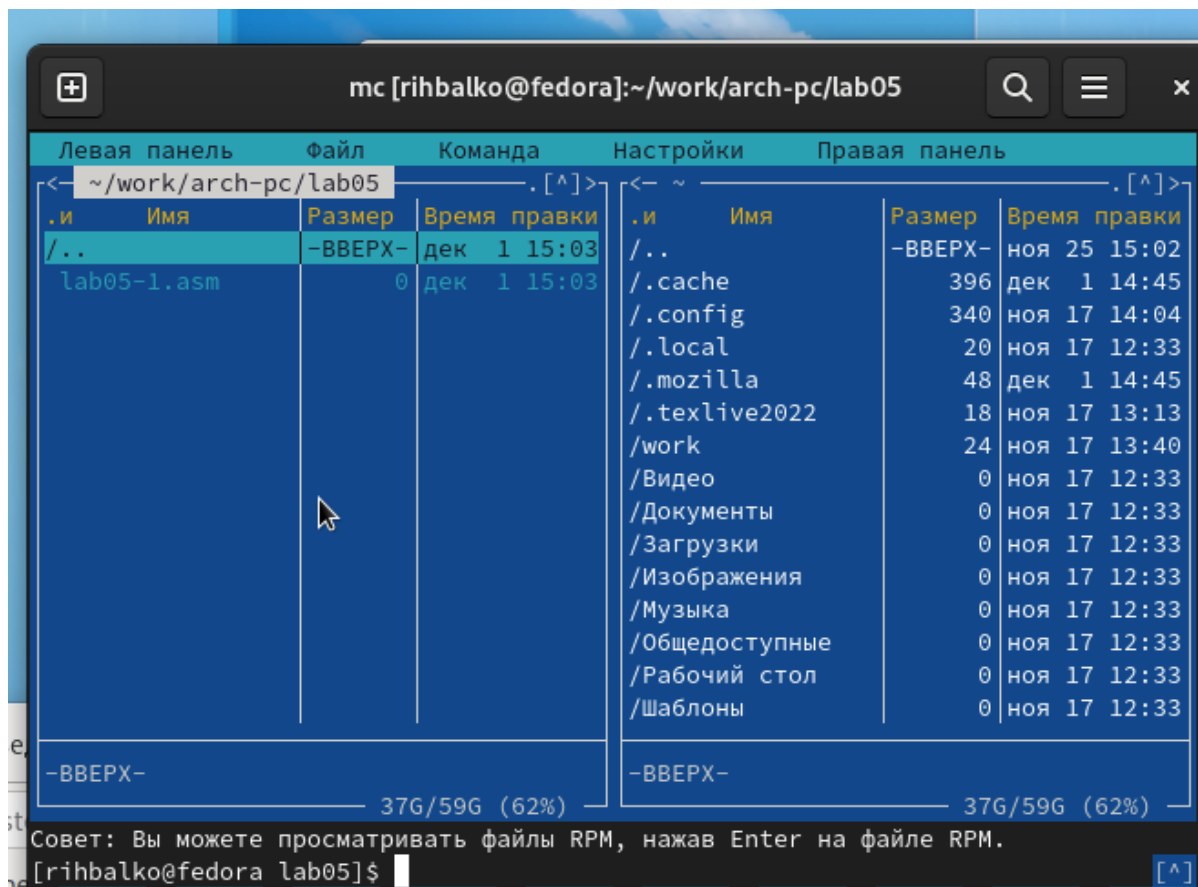
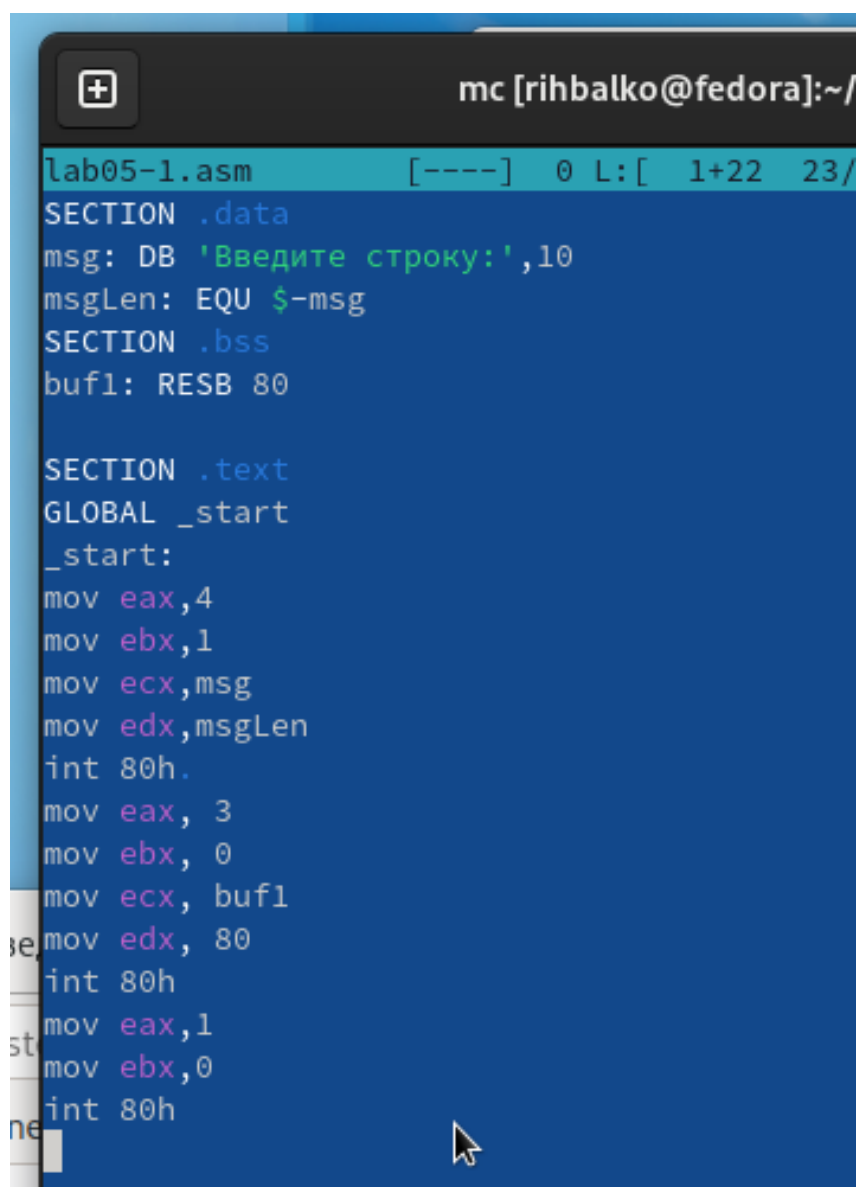


Рис. 2.2: Создание файла lab05-1.asm

Открыл файл на редактирование. Написал код. Указано на рис. [2.3]



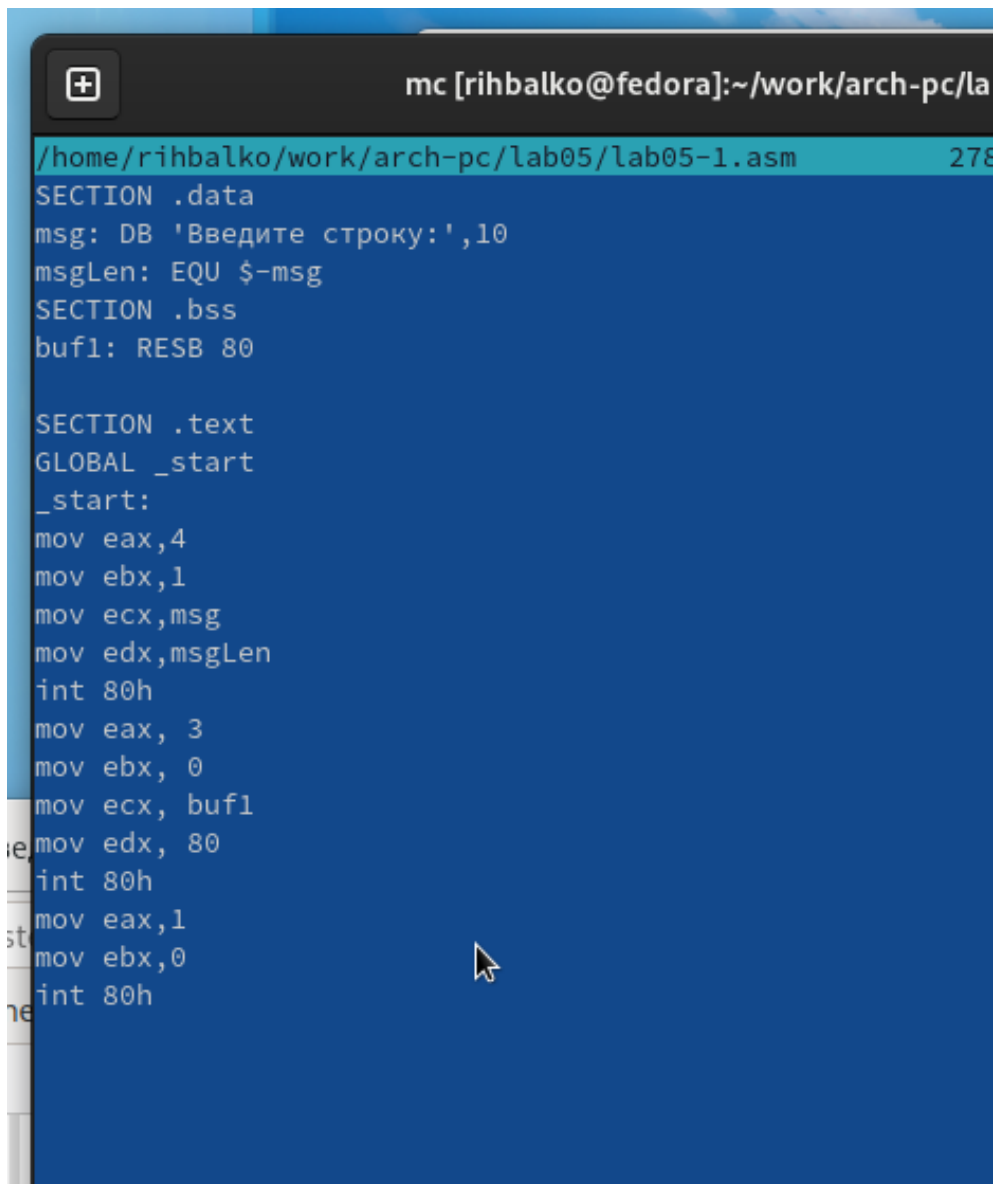
```
mc [rihbalko@fedora]:~/
lab05-1.asm [----] 0 L:[ 1+22 23/
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h.
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рис. 2.3: Код программы lab05-1.asm

Открыл файл для просмотра и убедился, что он содержит написанный код.  
Указано на рис. [2.4]





```
mc [rihbalko@fedora]:~/work/arch-pc/la
/home/rihbalko/work/arch-pc/lab05/lab05-1.asm 278
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рис. 2.4: Просмотр файла lab05-1.asm

Получил исполняемый файл программы и проверил его работу. Указано на рис.  
[2.5]

```
[rihbalko@fedora lab05]$ nasm -f elf lab05-1.asm
[rihbalko@fedora lab05]$ ld -m elf_i386 lab05-1.o -o lab05-1
[rihbalko@fedora lab05]$ ./lab05-1
Введите строку:
Nikolay
[rihbalko@fedora lab05]$
```

Рис. 2.5: Сборка и проверка программы lab05-1.asm

## 2.2 Подключение внешнего файла in\_out.asm

Скачал файл in\_out.asm. Добавил файл in\_out.asm в рабочий каталог. Скопировал lab05-1.asm в lab05-2.asm. Указано на рис. [2.6]

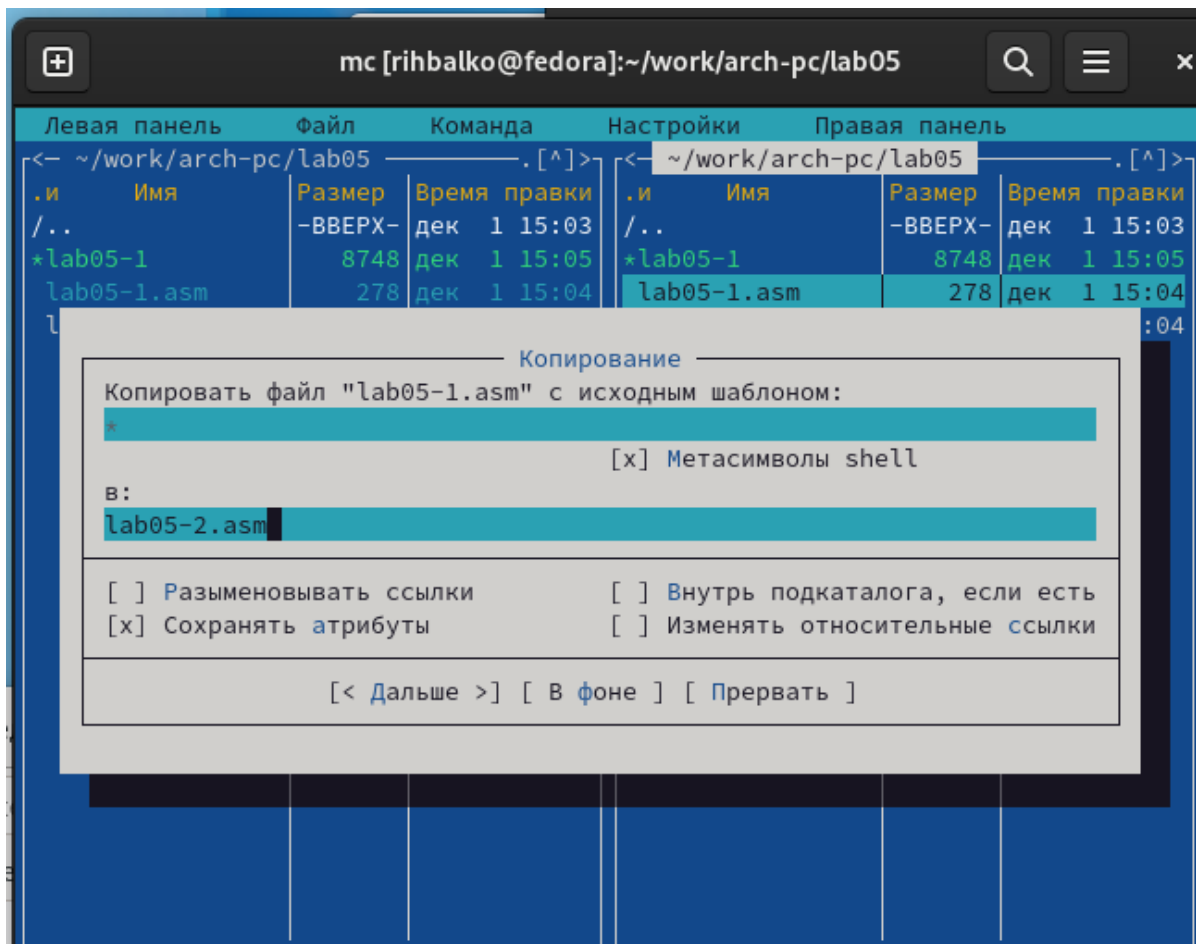
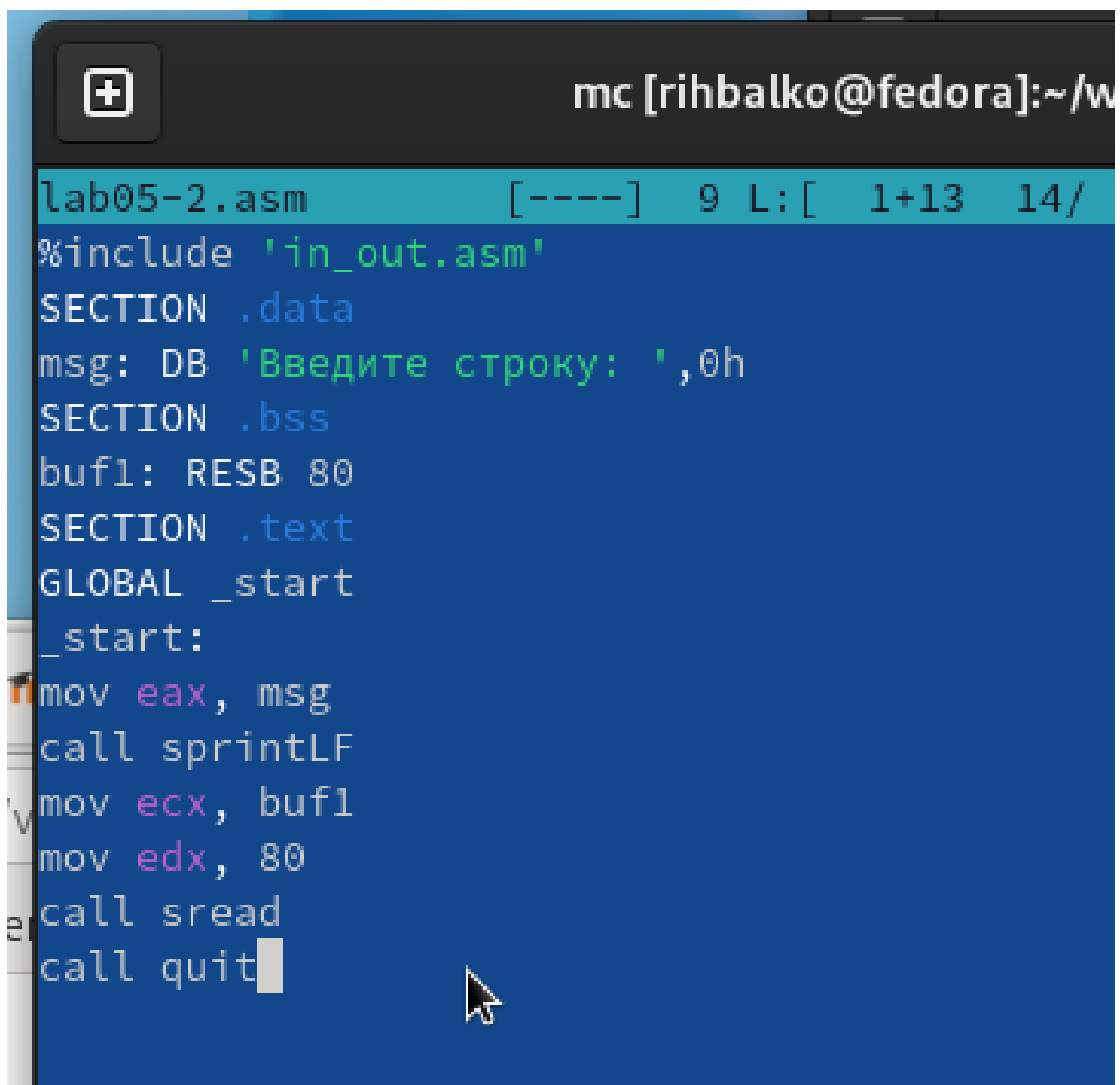


Рис. 2.6: Копирование файла

Написал код программы lab05-2.asm. Указано на рис. [2.7]



```
lab05-2.asm [-----] 9 L: [ 1+13 14/
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку: ',0h
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintLF
mov ecx, buf1
mov edx, 80
call sread
call quit
```

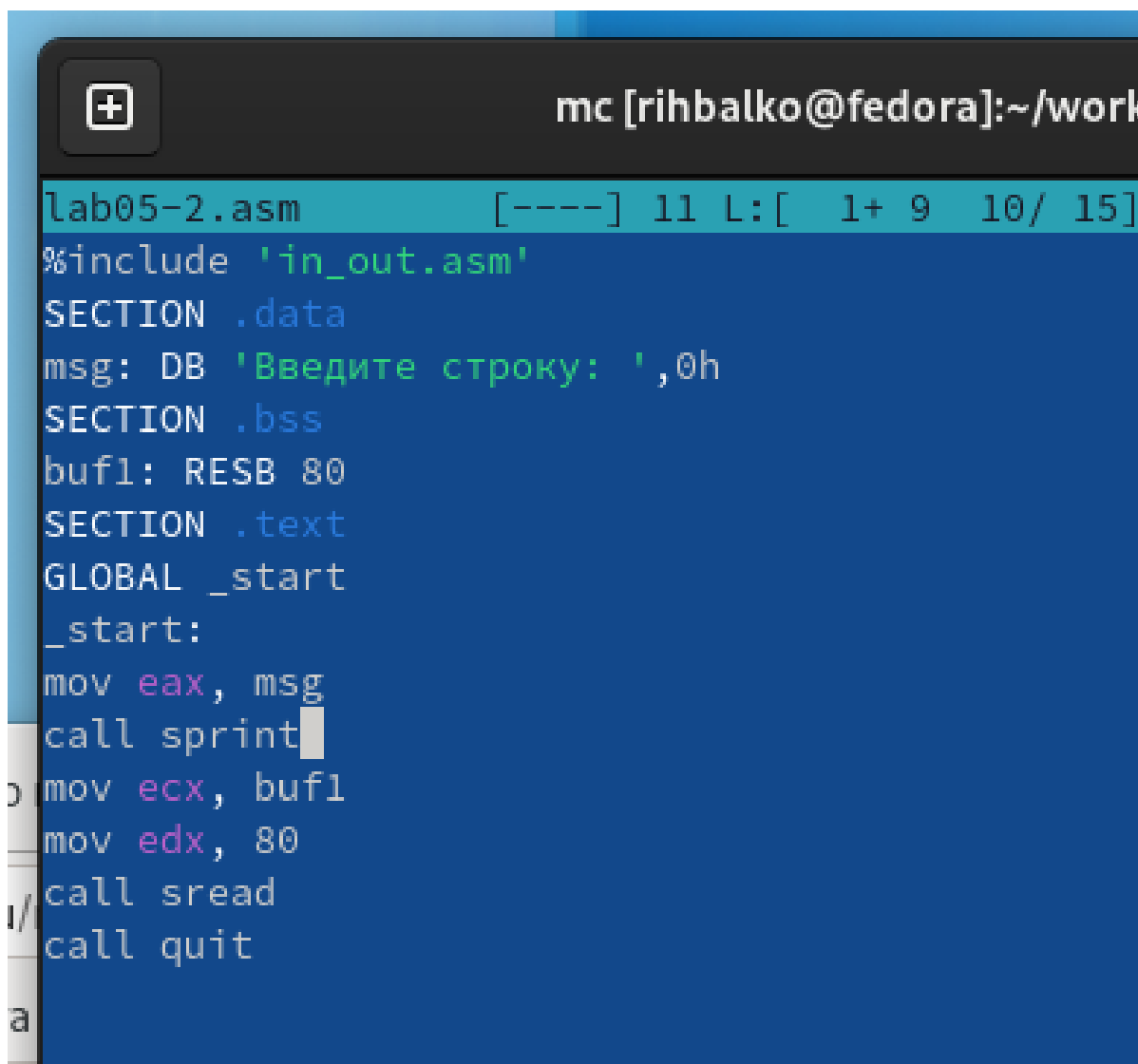
Рис. 2.7: Код программы lab05-2.asm

Скомпилировал программу и проверил запуск. Указано на рис. [2.8]

```
[rihbalko@fedora lab05]$  
[rihbalko@fedora lab05]$ nasm -f elf lab05-2.asm  
[rihbalko@fedora lab05]$ ld -m elf_i386 lab05-2.o -o lab05-2  
[rihbalko@fedora lab05]$ ./lab05-2  
Введите строку:  
Nikolay  
[rihbalko@fedora lab05]$
```

Рис. 2.8: Сборка и проверка программы lab05-2.asm

В файле lab5-2.asm я заменил подпрограмму sprintLF на sprint Указано на рис. [2.9].



```
lab05-2.asm [-----] 11 L:[ 1+ 9 10/ 15]
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку: ',0h
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprint
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рис. 2.9: Код программы lab05-2.asm

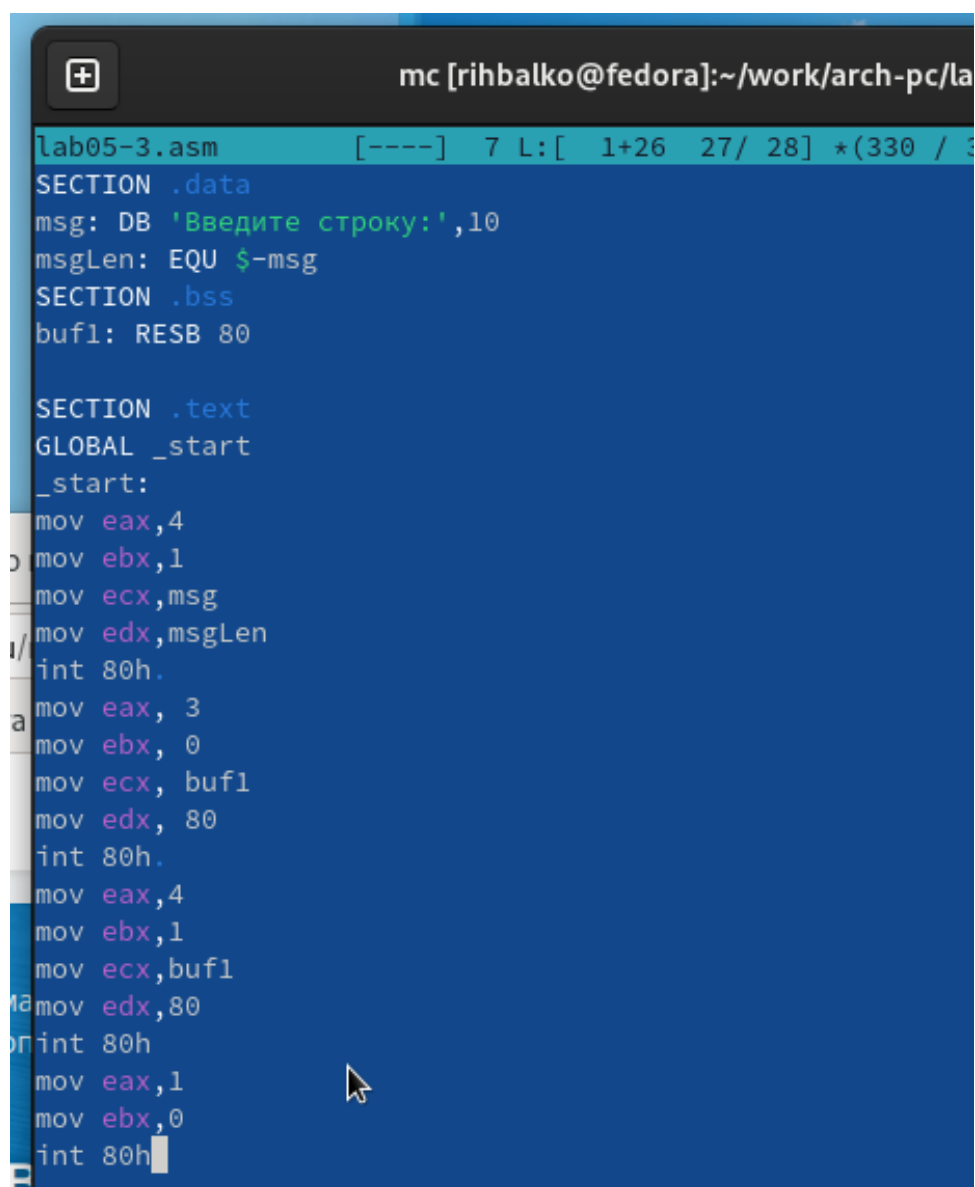
Затем я снова собрал исполняемый файл. Указано на рис. [2.10]. Теперь после вывода строки она не завершается символом перехода на новую строку.

```
[rihbalko@fedora lab05]$  
[rihbalko@fedora lab05]$ nasm -f elf lab05-2.asm  
[rihbalko@fedora lab05]$ ld -m elf_i386 lab05-2.o -o lab05-2  
[rihbalko@fedora lab05]$ ./lab05-2  
Введите строку: Nikolay  
[rihbalko@fedora lab05]$
```

Рис. 2.10: Сборка и проверка программы lab05-2.asm

## 2.3 Задание для самостоятельной работы

Скопировал программу lab05-1.asm и изменил код, чтобы программа выводила приглашение типа “Введите строку:”, затем считывала строку с клавиатуры и выводила введенную строку на экран. Указано на рис. [-2.11][2.12]



```
mc [rihbalko@fedora]:~/work/arch-pc/la
lab05-3.asm [----] 7 L: [ 1+26 27/ 28] *(330 / 3
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h.
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h.
mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,80
int 80h
mov eax,1
mov ebx,0
int 80h
```

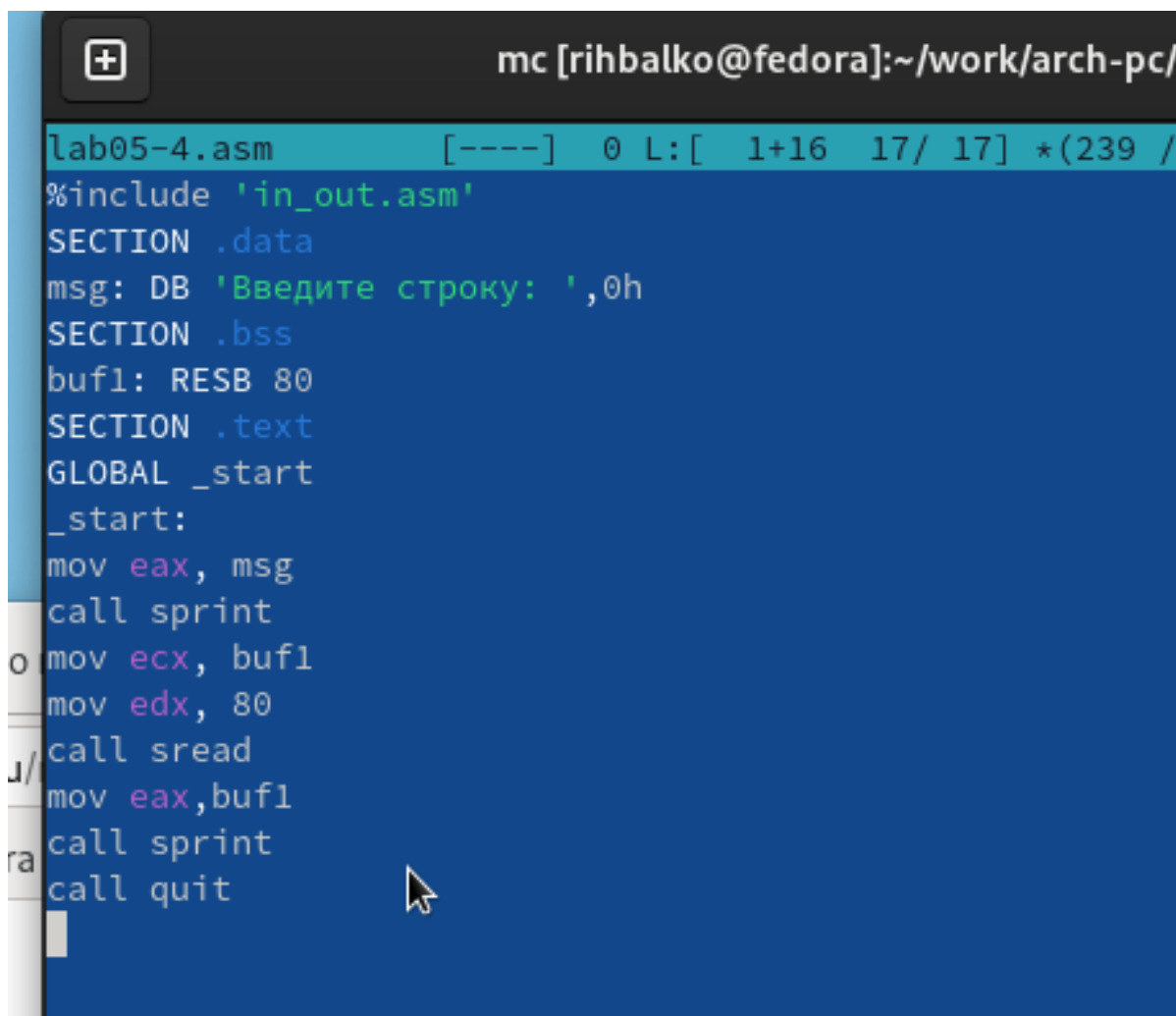
Рис. 2.11: Код программы lab05-3.asm



```
[rihbalko@fedora lab05]$  
[rihbalko@fedora lab05]$  
[rihbalko@fedora lab05]$ nasm -f elf lab05-3.asm  
[rihbalko@fedora lab05]$ ld -m elf_i386 lab05-3.o -o lab05-3  
[rihbalko@fedora lab05]$ ./lab05-3  
Введите строку:  
Nikolay  
Nikolay  
[rihbalko@fedora lab05]$
```

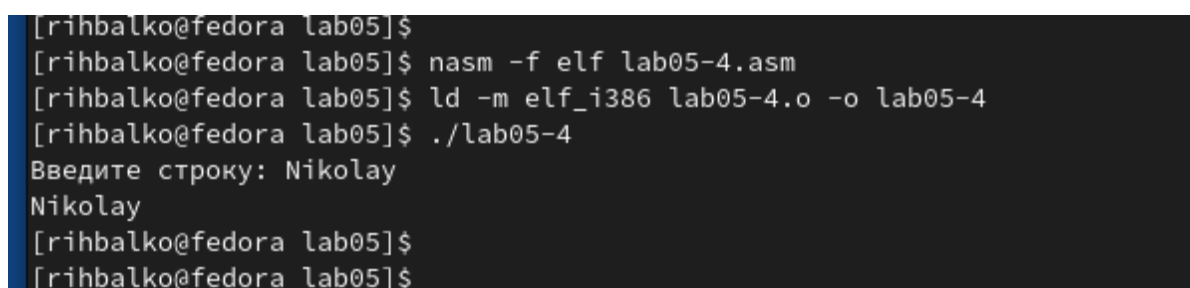
Рис. 2.12: Сборка и проверка программы lab05-3.asm

Также я скопировал программу lab05-2.asm и внес соответствующие изменения в код, чтобы программа выводила приглашение типа “Введите строку:”, затем считывала строку с клавиатуры и выводила введенную строку на экран. Указано на рис. [-2.13][2.14]



```
lab05-4.asm [----] 0 L: [ 1+16 17/ 17] *(239 /
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку: ',0h
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprint
mov ecx, buf1
mov edx, 80
call sread
mov eax, buf1
call sprint
call quit
```

Рис. 2.13: Код программы lab05-4.asm



```
[rihbalko@fedora lab05]$
[rihbalko@fedora lab05]$ nasm -f elf lab05-4.asm
[rihbalko@fedora lab05]$ ld -m elf_i386 lab05-4.o -o lab05-4
[rihbalko@fedora lab05]$ ./lab05-4
Введите строку: Nikolay
Nikolay
[rihbalko@fedora lab05]$
[rihbalko@fedora lab05]$
```

Рис. 2.14: Сборка и проверка программы lab05-4.asm

Отличие этих двух реализаций заключается в том, что файл in\_out.asm содер-

жит уже готовые подпрограммы для обеспечения ввода/вывода. Таким образом, нам остается только разместить данные в нужных регистрах и вызвать желаемую подпрограмму с помощью инструкции call.

## 3 Выводы

Научились писать базовые ассемблерные программы. Освоили ассемблерные инструкции `mov` и `int`.