

# **Отчёта по лабораторной работе 5**

**Основы работы с Midnight Commander (mc). Структура программы на языке ассемблера NASM**

Тукаев Тимур Ильшатович НММбд-03-23

# Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	20

## Список иллюстраций

2.1	Создание каталога . . . . .	6
2.2	Создание файла lab05-1.asm . . . . .	7
2.3	Программа в файле lab05-1.asm . . . . .	8
2.4	Просмотр файла lab05-1.asm . . . . .	10
2.5	Запуск программы lab05-1.asm . . . . .	11
2.6	Копирование файла . . . . .	12
2.7	Программа в файле lab05-2.asm . . . . .	13
2.8	Запуск программы lab05-2.asm . . . . .	14
2.9	Программа в файле lab05-2.asm . . . . .	14
2.10	Запуск программы lab05-2.asm . . . . .	15
2.11	Программа в файле lab05-3.asm . . . . .	16
2.12	Запуск программы lab05-3.asm . . . . .	18
2.13	Программа в файле lab05-4.asm . . . . .	18
2.14	Запуск программы lab05-4.asm . . . . .	19

## 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

1. Открыл Midnight Commander
2. Перешел в каталог ~/work/arch-pc
3. Создал каталог lab05

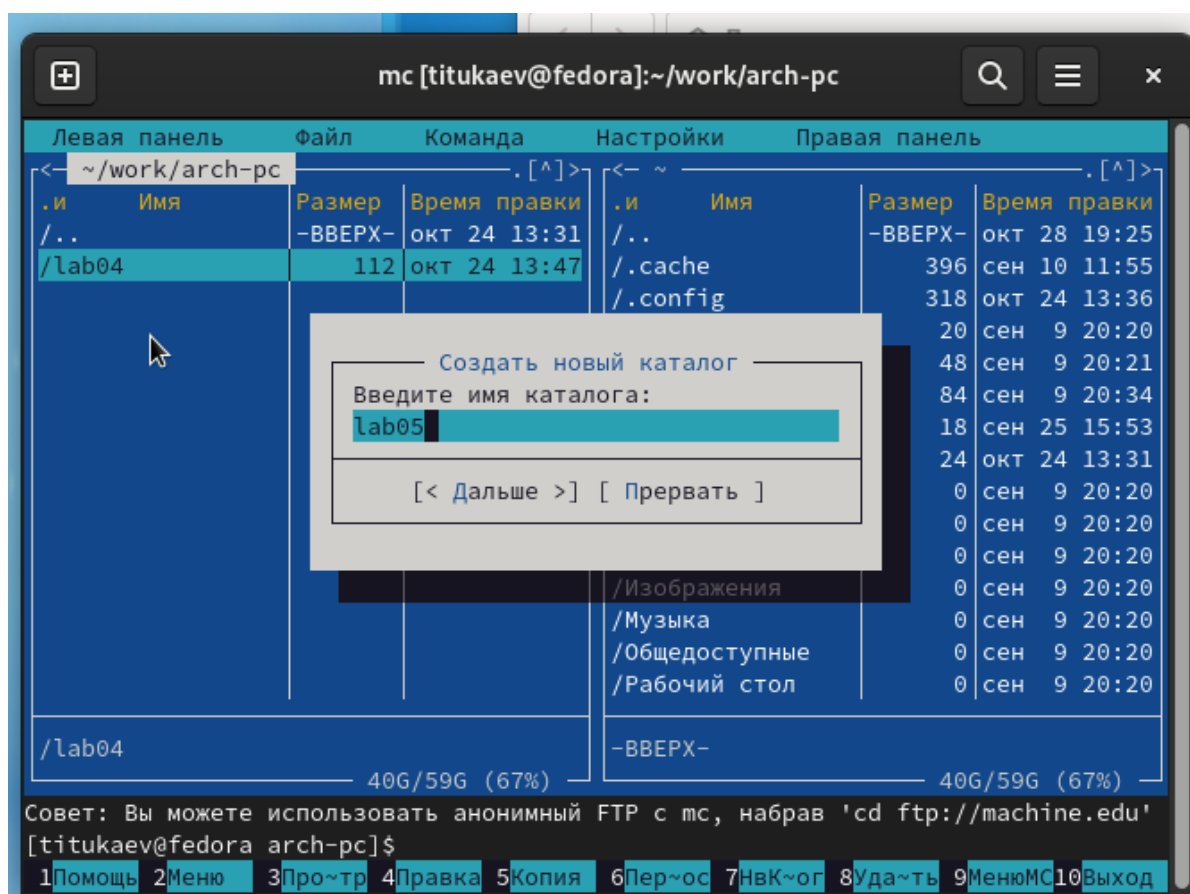


Рис. 2.1: Создание каталога

4. Создал файл lab05-1.asm

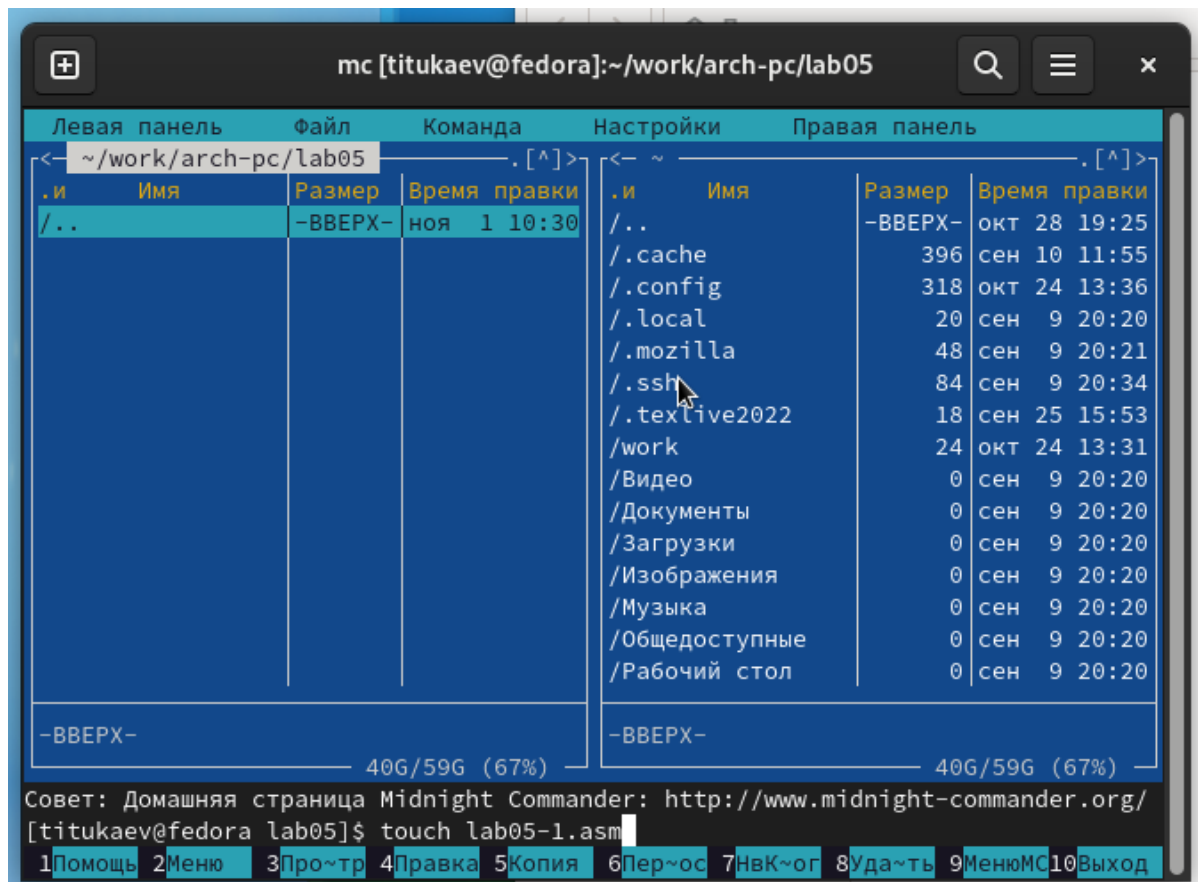
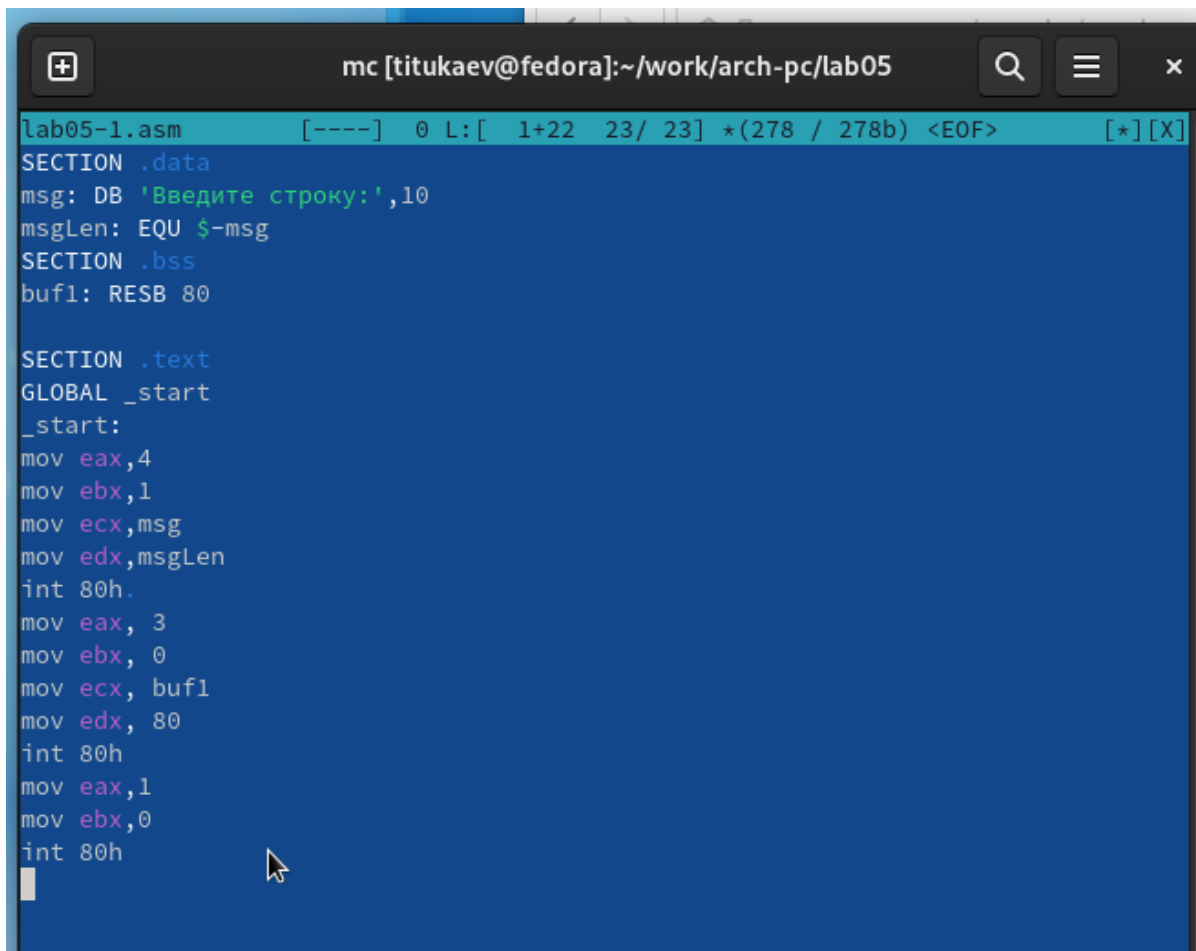


Рис. 2.2: Создание файла lab05-1.asm

5. Открыл файл на редактирование

6. Написал код



```
mc [titukaev@fedora]:~/work/arch-pc/lab05
lab05-1.asm [----] 0 L: [ 1+22 23/ 23] *(278 / 278b) <EOF> [*] [X]
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: Программа в файле lab05-1.asm

Также размещаю код программы в отчете.

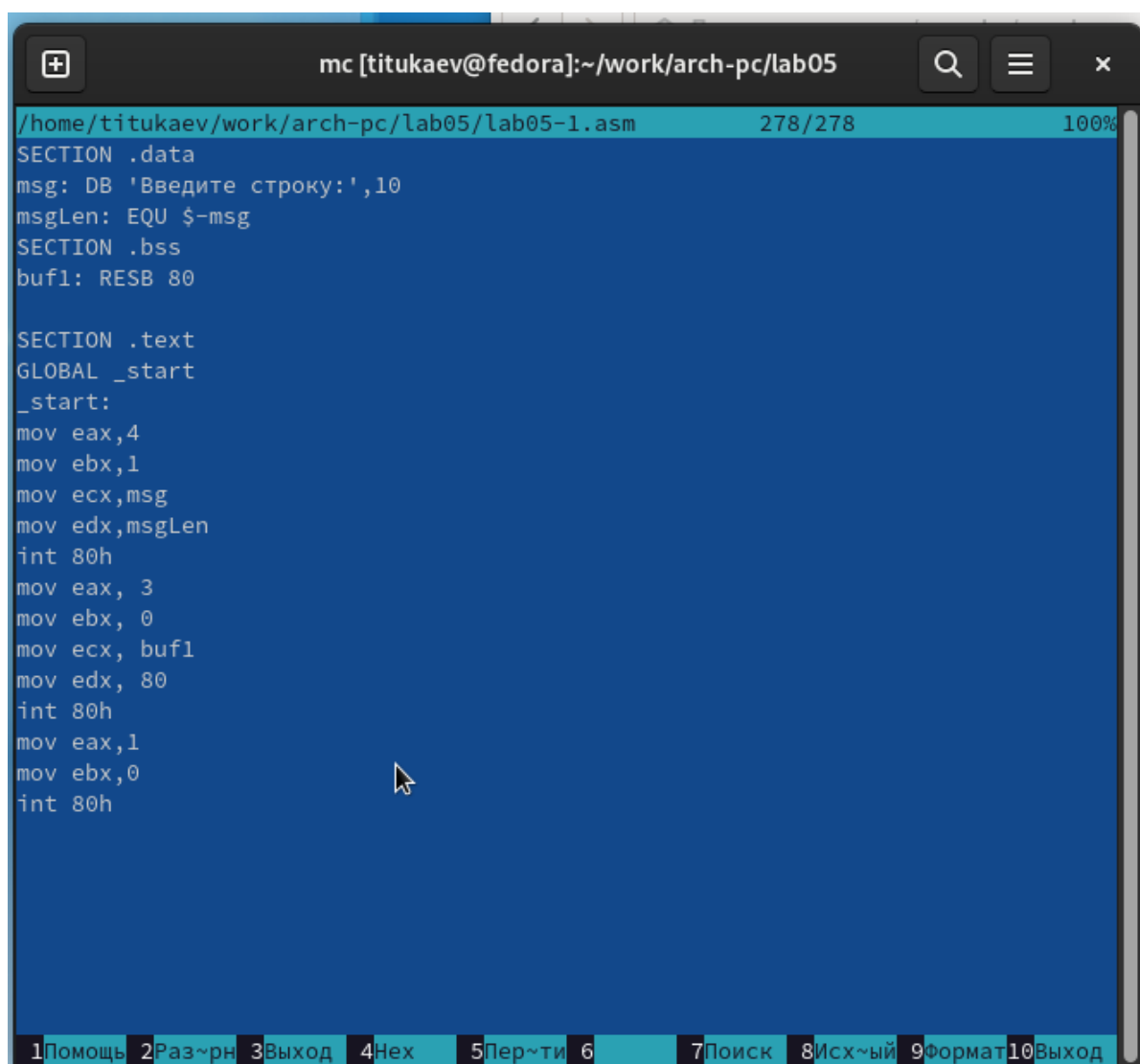
```
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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
```

7. Открыл файл на просмотр и убелился, что он содержит набранный код.



The screenshot shows a file editor window titled "mc [titukaev@fedora]:~/work/arch-pc/lab05". The file being edited is "/home/titukaev/work/arch-pc/lab05/lab05-1.asm", which is 278/278 bytes in size and is viewed at 100% zoom. The code is written in assembly and is as follows:

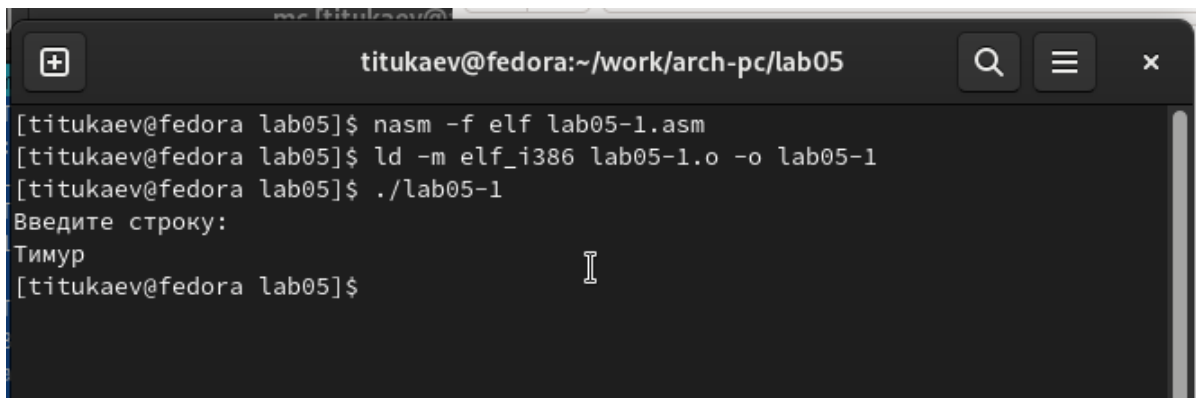
```
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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
```

At the bottom of the window, there is a menu bar with the following items: 1Помощь, 2Раз~рн, 3Выход, 4Hex, 5Пер~ти, 6, 7Поиск, 8Исх~ый, 9Формат, 10Выход.

Рис. 2.4: Просмотр файла lab05-1.asm

8. Получил исполняемый файл программы и проверил ее работу.

A terminal window titled 'titukaev@fedora:~/work/arch-pc/lab05'. The terminal shows the following commands and output:

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

Рис. 2.5: Запуск программы lab05-1.asm

9. Скачал файл in\_out.asm.
10. Добавил файл in\_out.asm в рабочий каталог.
11. Скопировал lab05-1.asm в lab05-2.asm.

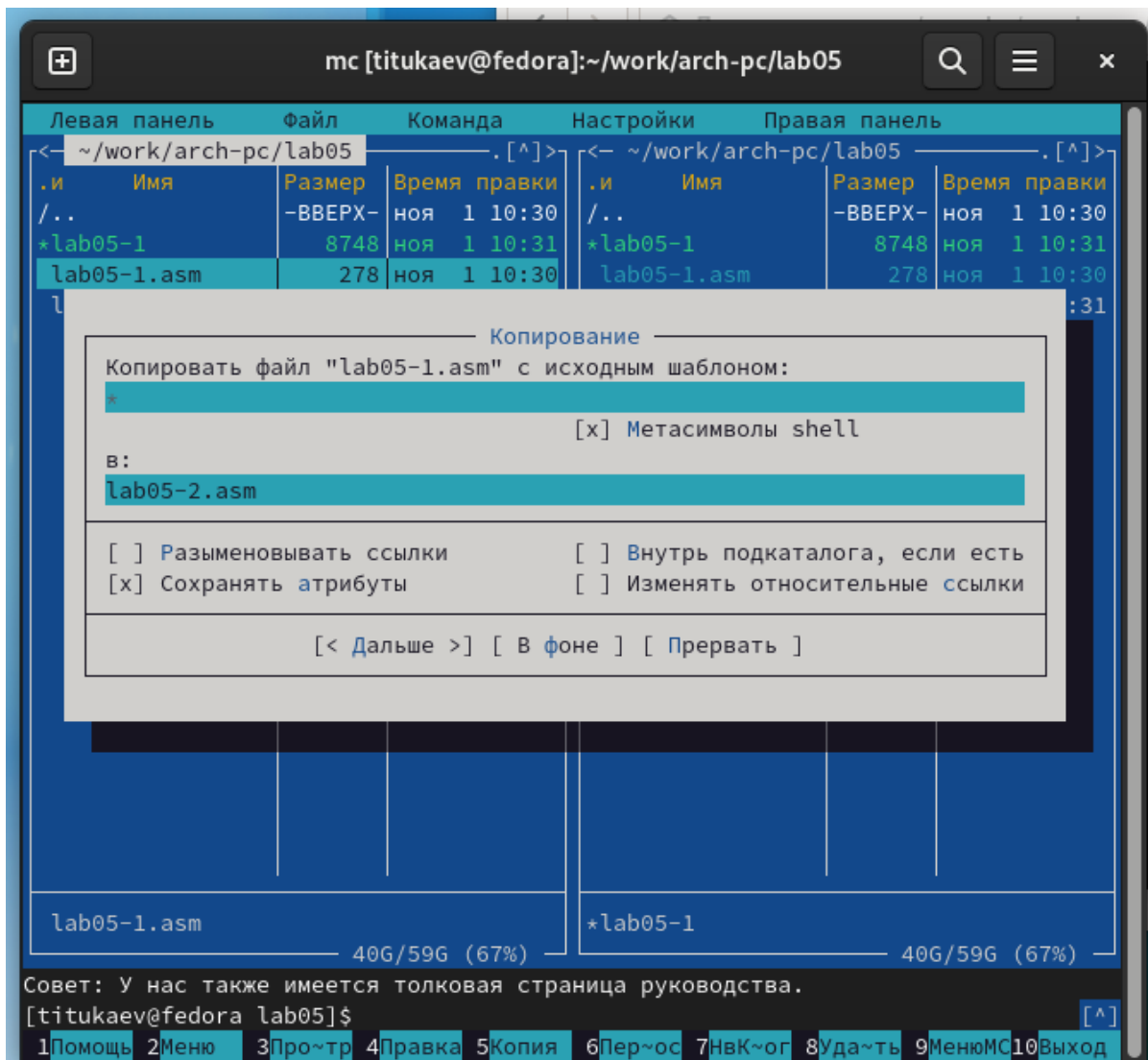
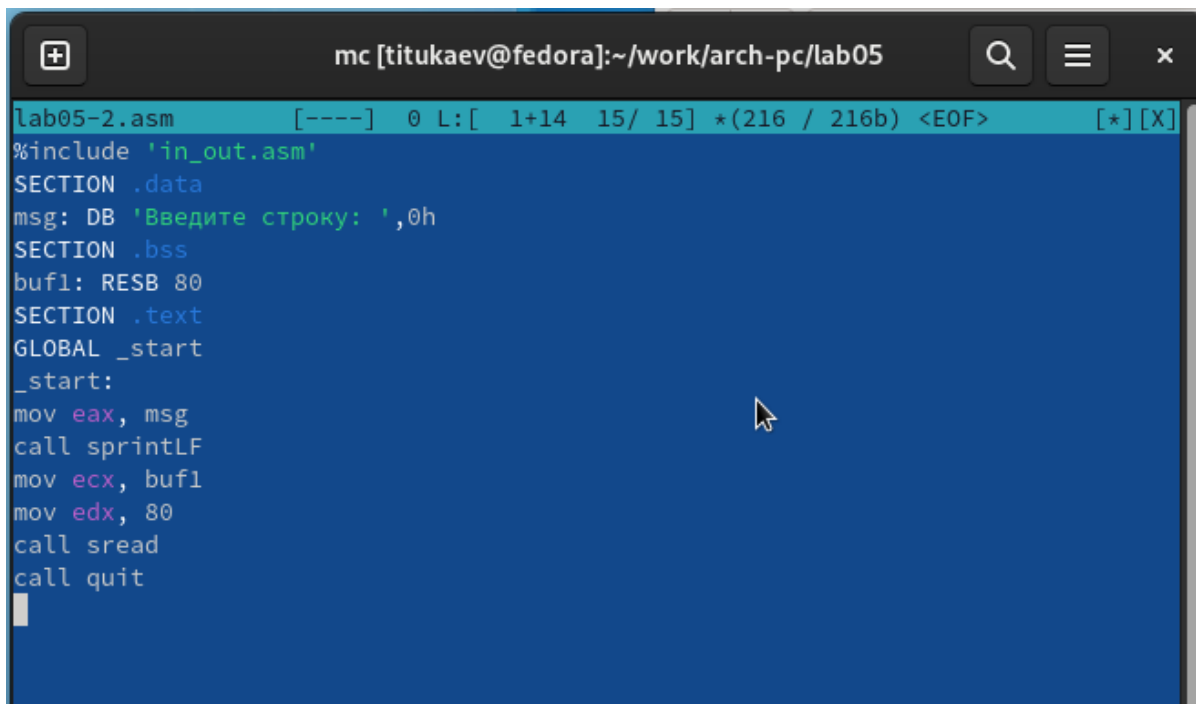


Рис. 2.6: Копирование файла

12. Написал код программы lab05-2.asm. Скомпилировал программу и проверили запуск.



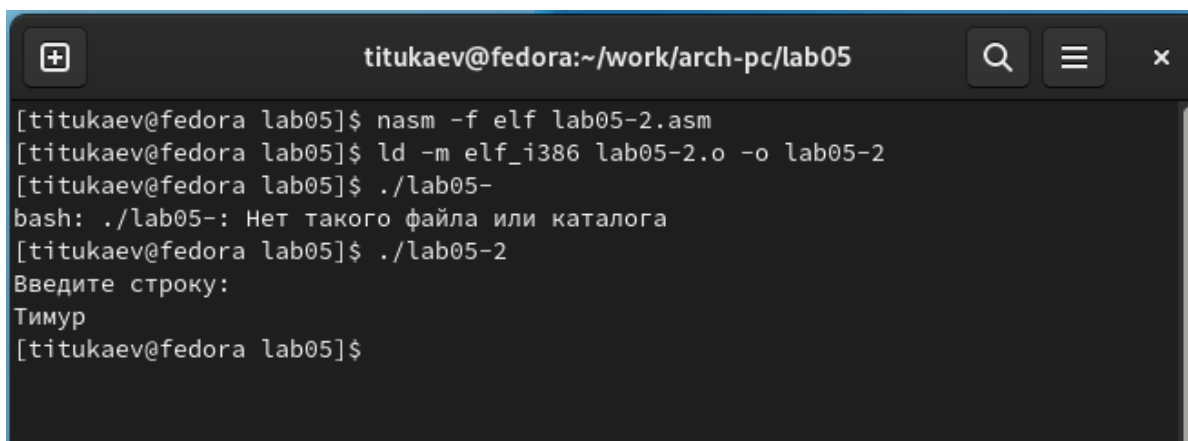
```
mc [titukaev@fedora]:~/work/arch-pc/lab05
lab05-2.asm [----] 0 L:[ 1+14 15/ 15] *(216 / 216b) <EOF> [*][X]
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку: ',0h
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintLF
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рис. 2.7: Программа в файле lab05-2.asm

Также размещаю код программы в отчете.

```
%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
```

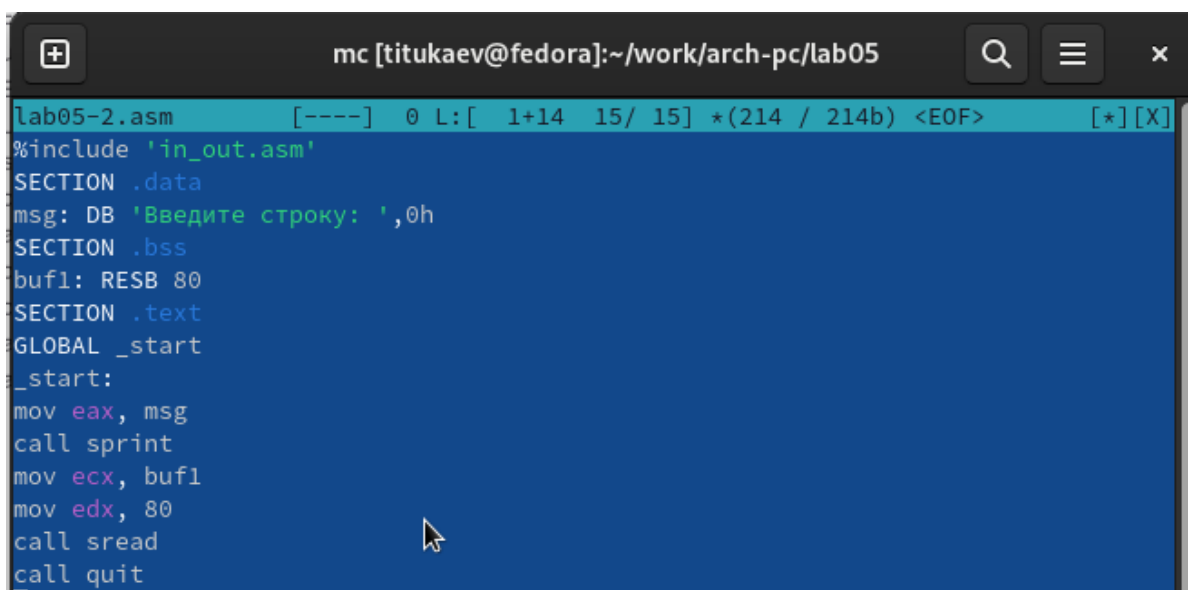
call quit



```
titukaev@fedora:~/work/arch-pc/lab05
[titukaev@fedora lab05]$ nasm -f elf lab05-2.asm
[titukaev@fedora lab05]$ ld -m elf_i386 lab05-2.o -o lab05-2
[titukaev@fedora lab05]$ ./lab05-
bash: ./lab05-: Нет такого файла или каталога
[titukaev@fedora lab05]$ ./lab05-2
Введите строку:
Тимур
[titukaev@fedora lab05]$
```

Рис. 2.8: Запуск программы lab05-2.asm

13. В файле lab5-2.asm заменил подпрограмму sprintLF на sprint. Заново собрал исполняемый файл. Теперь после вывода строки она не завершается символом перехода на новую строку.



```
mc [titukaev@fedora]:~/work/arch-pc/lab05
lab05-2.asm  [----]  0 L: [ 1+14 15/ 15] *(214 / 214b) <EOF>  [*] [X]
%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: Программа в файле lab05-2.asm

Также размещаю код программы в отчете.

```

%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

```

```

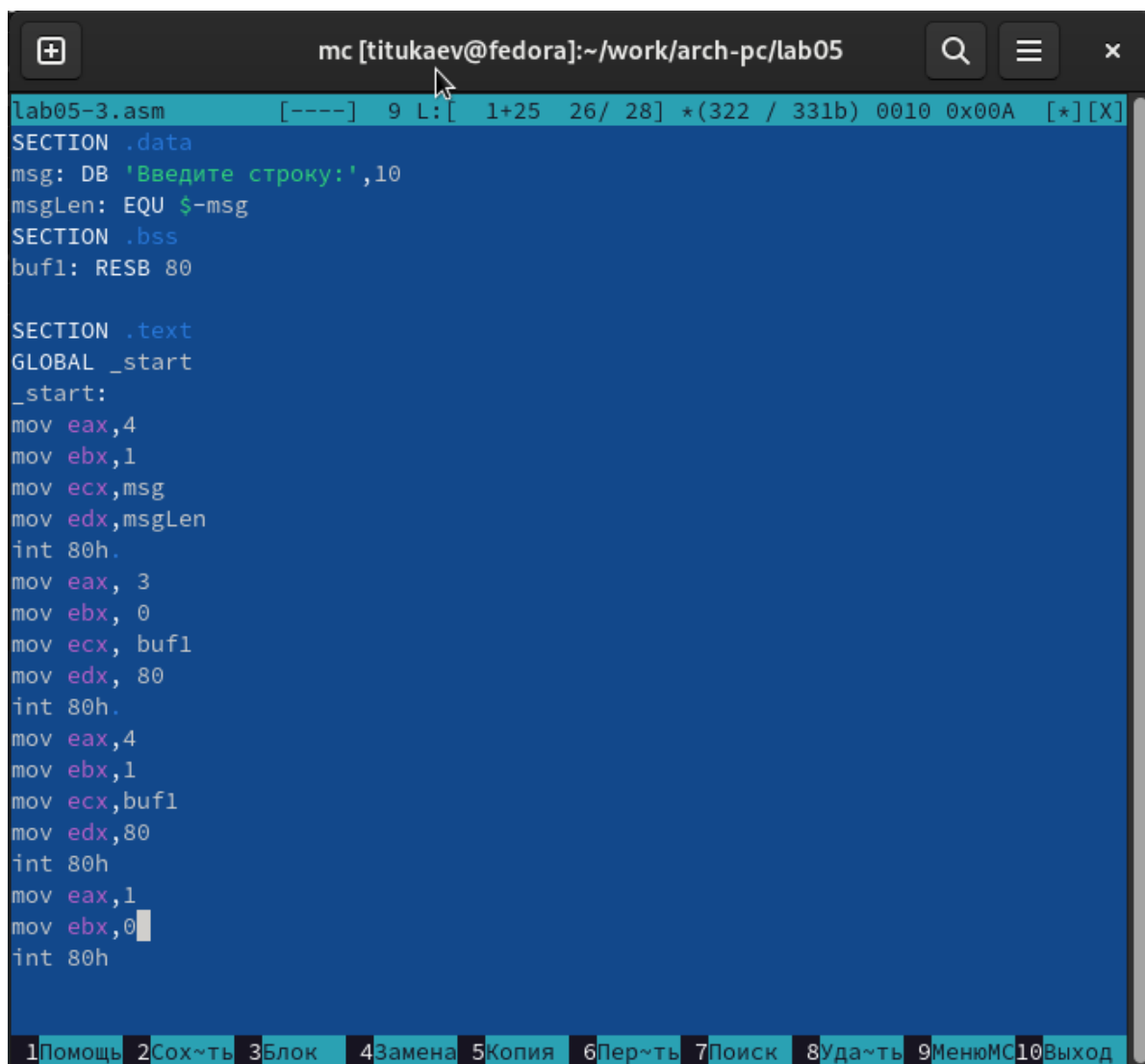
titukaev@fedora:~/work/arch-pc/lab05
[titukaev@fedora lab05]$ nasm -f elf lab05-2.asm
[titukaev@fedora lab05]$ ld -m elf_i386 lab05-2.o -o lab05-2
[titukaev@fedora lab05]$ ./lab05-2
bash: ./lab05-2: Нет такого файла или каталога
[titukaev@fedora lab05]$ ./lab05-2
Введите строку:
Тимур
[titukaev@fedora lab05]$
[titukaev@fedora lab05]$ nasm -f elf lab05-2.asm
[titukaev@fedora lab05]$ ld -m elf_i386 lab05-2.o -o lab05-2
[titukaev@fedora lab05]$ ./lab05-2
Введите строку: Тимур
[titukaev@fedora lab05]$

```

Рис. 2.10: Запуск программы lab05-2.asm

14. Скопировал программу lab05-1.asm и изменил код, чтобы вывести при-

глашение типа “Введите строку:”, ввести строку с клавиатуры, вывести введенную строку на экран.



```
lab05-3.asm  [-----]  9  L: [ 1+25 26/ 28] *(322 / 331b) 0010 0x00A  [*] [X]
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

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9МенюMC10Выход
```

Рис. 2.11: Программа в файле lab05-3.asm

Также размещаю код программы в отчете.

```
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
```

```

[titukaev@fedora lab05]$ cp lab05-1.asm lab05-3.asm
[titukaev@fedora lab05]$
[titukaev@fedora lab05]$ nasm -f elf lab05-3.asm
[titukaev@fedora lab05]$ ld -m elf_i386 lab05-3.o -o lab05-3
[titukaev@fedora lab05]$ ./lab05-3
Введите строку:
Тимур
Тимур
[titukaev@fedora lab05]$

```

Рис. 2.12: Запуск программы lab05-3.asm

15. Скопировал программу lab05-2.asm и изменил код, чтобы вывести приглашение типа “Введите строку:”, ввести строку с клавиатуры, вывести введённую строку на экран.

```

lab05-4.asm  [----] 11 L: [ 1+14 15/ 17] *(228 / 239b) 0010 0x00A [*][X]
%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: Программа в файле lab05-4.asm

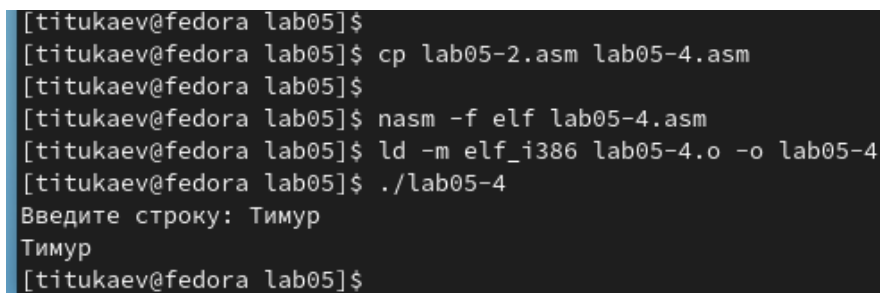
Также размещаю код программы в отчете.

```
%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
```



```
[titukaev@fedora lab05]$
[titukaev@fedora lab05]$ cp lab05-2.asm lab05-4.asm
[titukaev@fedora lab05]$
[titukaev@fedora lab05]$ nasm -f elf lab05-4.asm
[titukaev@fedora lab05]$ ld -m elf_i386 lab05-4.o -o lab05-4
[titukaev@fedora lab05]$ ./lab05-4
Введите строку: Тимур
Тимур
[titukaev@fedora lab05]$
```

Рис. 2.14: Запуск программы lab05-4.asm

Отличие этих двух реализаций в том, что файл in\_out.asm содержит уже готовые подпрограммы для обеспечения ввода/вывода. Таким образом, нам остается только разместить данные в нужных регистрах и вызвать желаемую подпрограмму с помощью call.

## 3 Выводы

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