

# **Отчёт по лабораторной работе 5**

**Архитектура компьютера**

Кликушин Георгий Александрович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выводы</b>	<b>16</b>

## 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 . . . . .	9
2.5	Запуск программы lab05-1.asm . . . . .	10
2.6	Копирование файла . . . . .	10
2.7	Программа в файле lab05-2.asm . . . . .	11
2.8	Запуск программы lab05-2.asm . . . . .	11
2.9	Программа в файле lab05-2.asm . . . . .	12
2.10	Запуск программы lab05-2.asm . . . . .	12
2.11	Программа в файле lab05-3.asm . . . . .	13
2.12	Запуск программы lab05-3.asm . . . . .	14
2.13	Программа в файле lab05-4.asm . . . . .	14
2.14	Запуск программы lab05-4.asm . . . . .	15

## **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

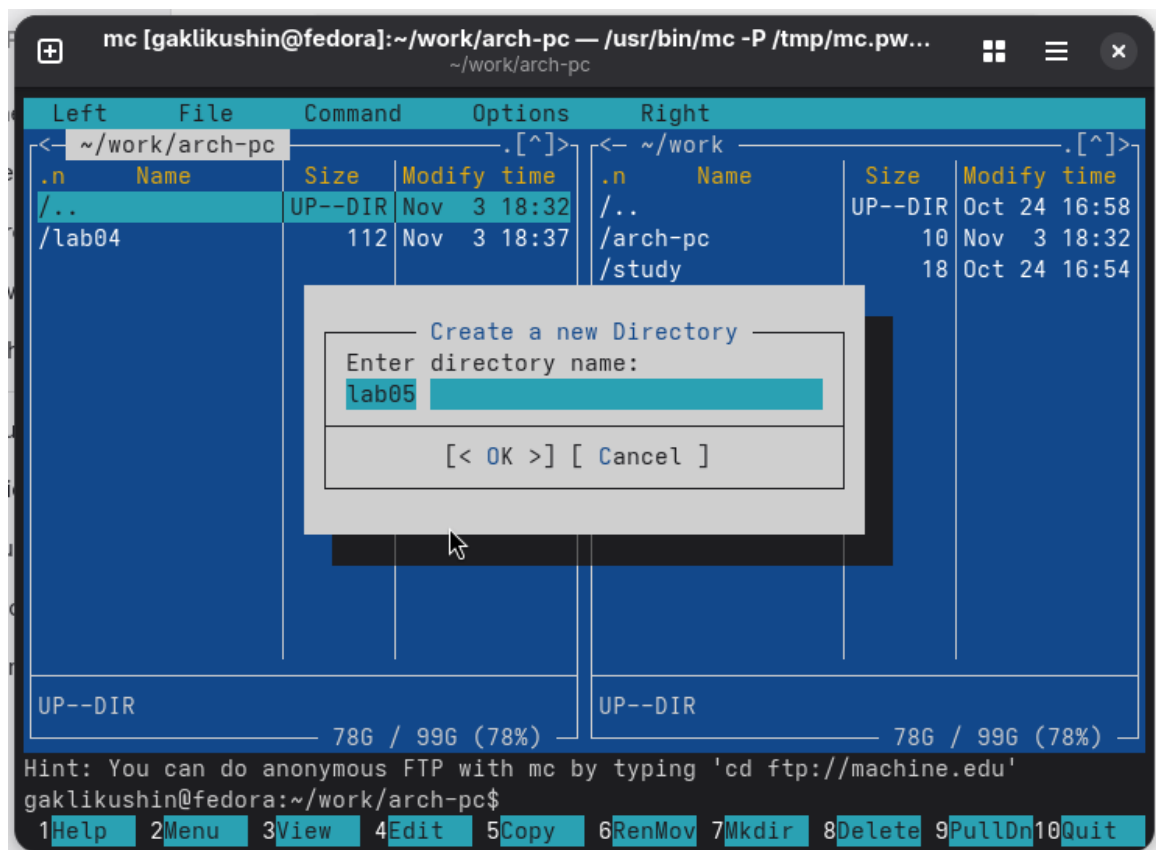


Рисунок 2.1: Создание каталога

4. Создал файл lab05-1.asm

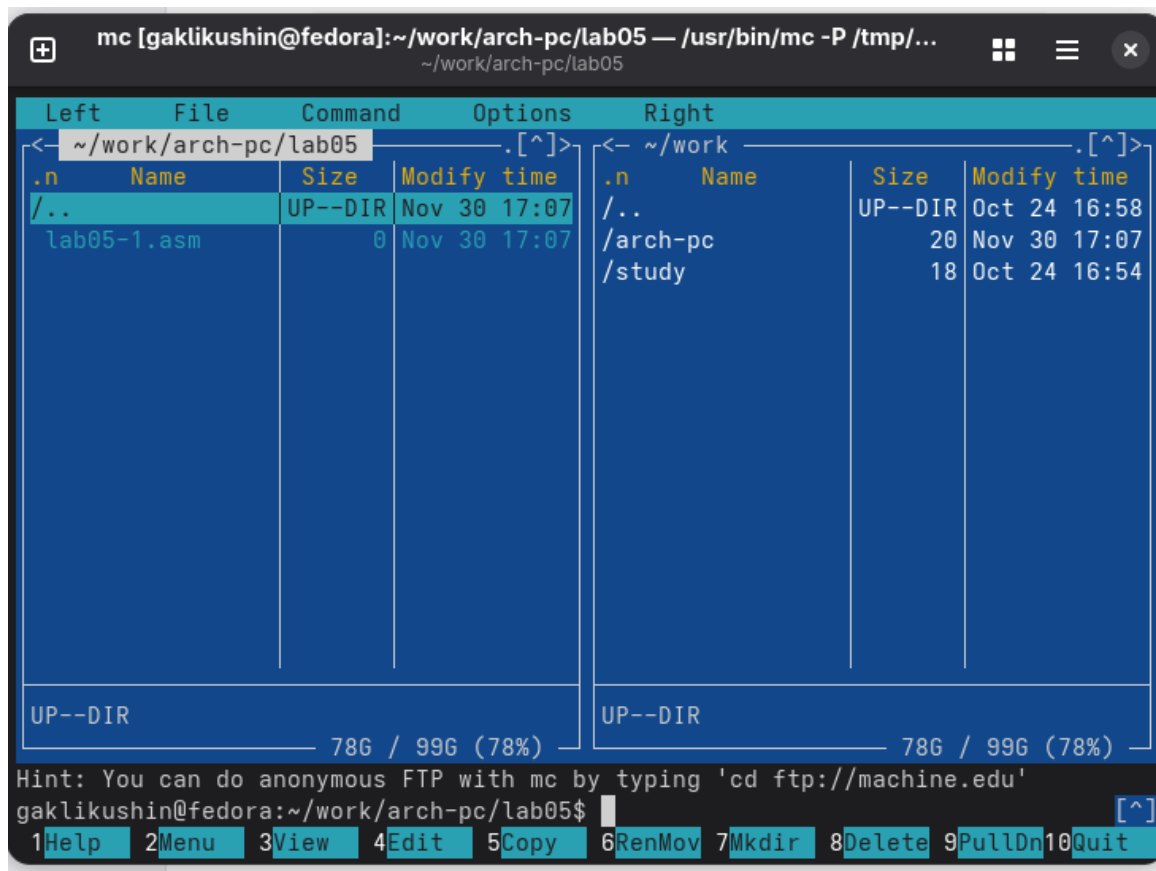
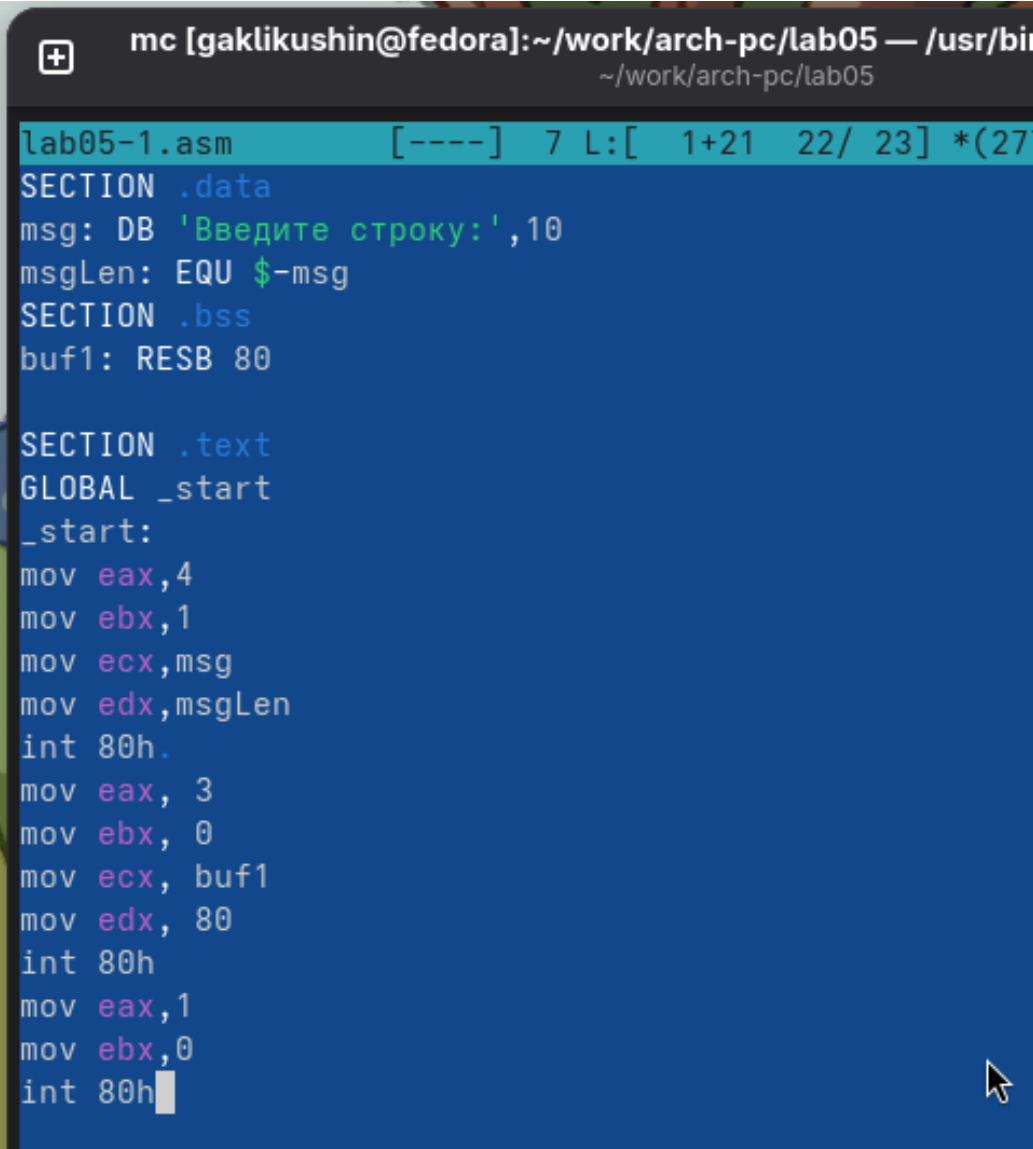


Рисунок 2.2: Создание файла lab05-1.asm

5. Открыл файл на редактирование

6. Написал код




```
mc [gaklikushin@fedora]:~/work/arch-pc/lab05 — /usr/bin
~/work/arch-pc/lab05
lab05-1.asm [----] 7 L:[ 1+21 22/ 23] *(27
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
```

Рисунок 2.3: Программа в файле lab05-1.asm

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





```
mc [gaklikushin@fedora]:~/work/arch-pc/lab05 — /usr
~/work/arch-pc/lab05
/home/gaklikushin/work/arch-pc/lab05/lab05-1.asm
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
```

Рисунок 2.4: Просмотр файла lab05-1.asm

8. Получил исполняемый файл программы и проверил ее работу.

```

gaklikushin@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-1.asm
gaklikushin@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-1.o -o lab05-1
gaklikushin@fedora:~/work/arch-pc/lab05$ ./lab05-1
Введите строку:
privet
gaklikushin@fedora:~/work/arch-pc/lab05$

```

Рисунок 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.

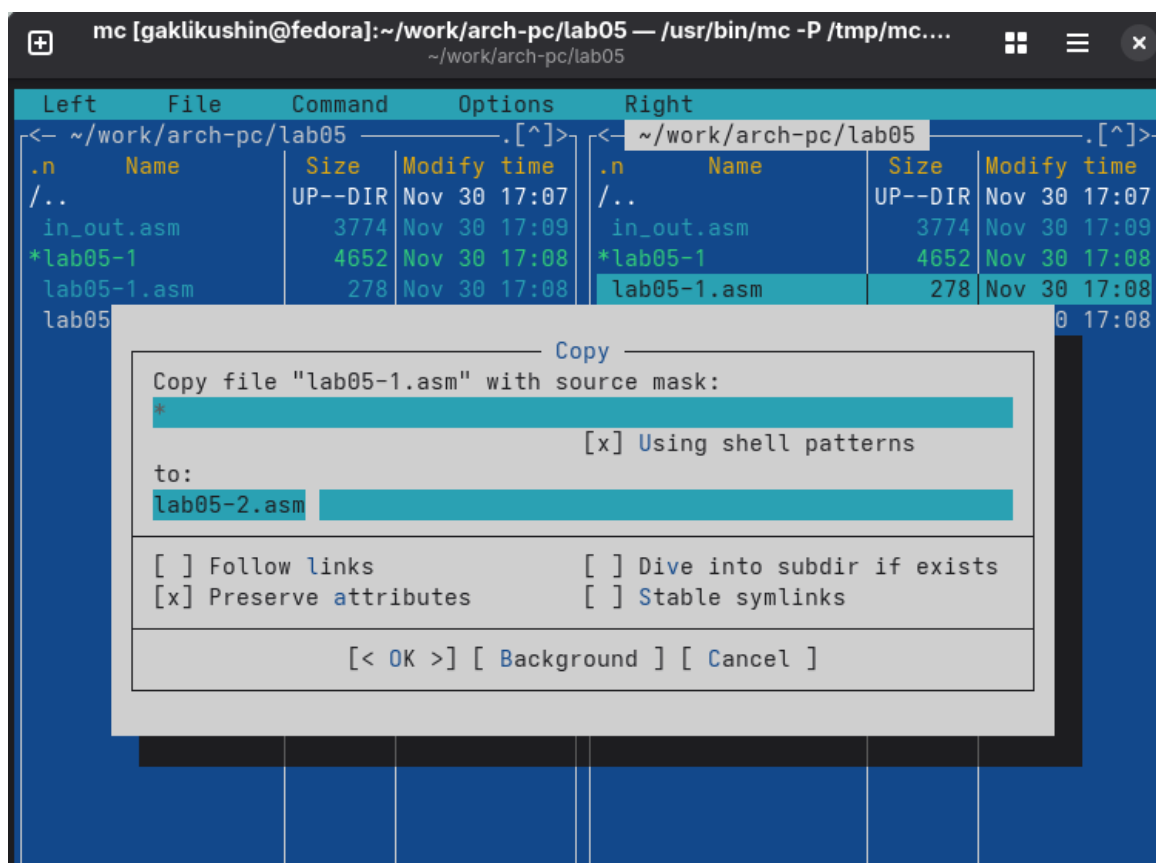
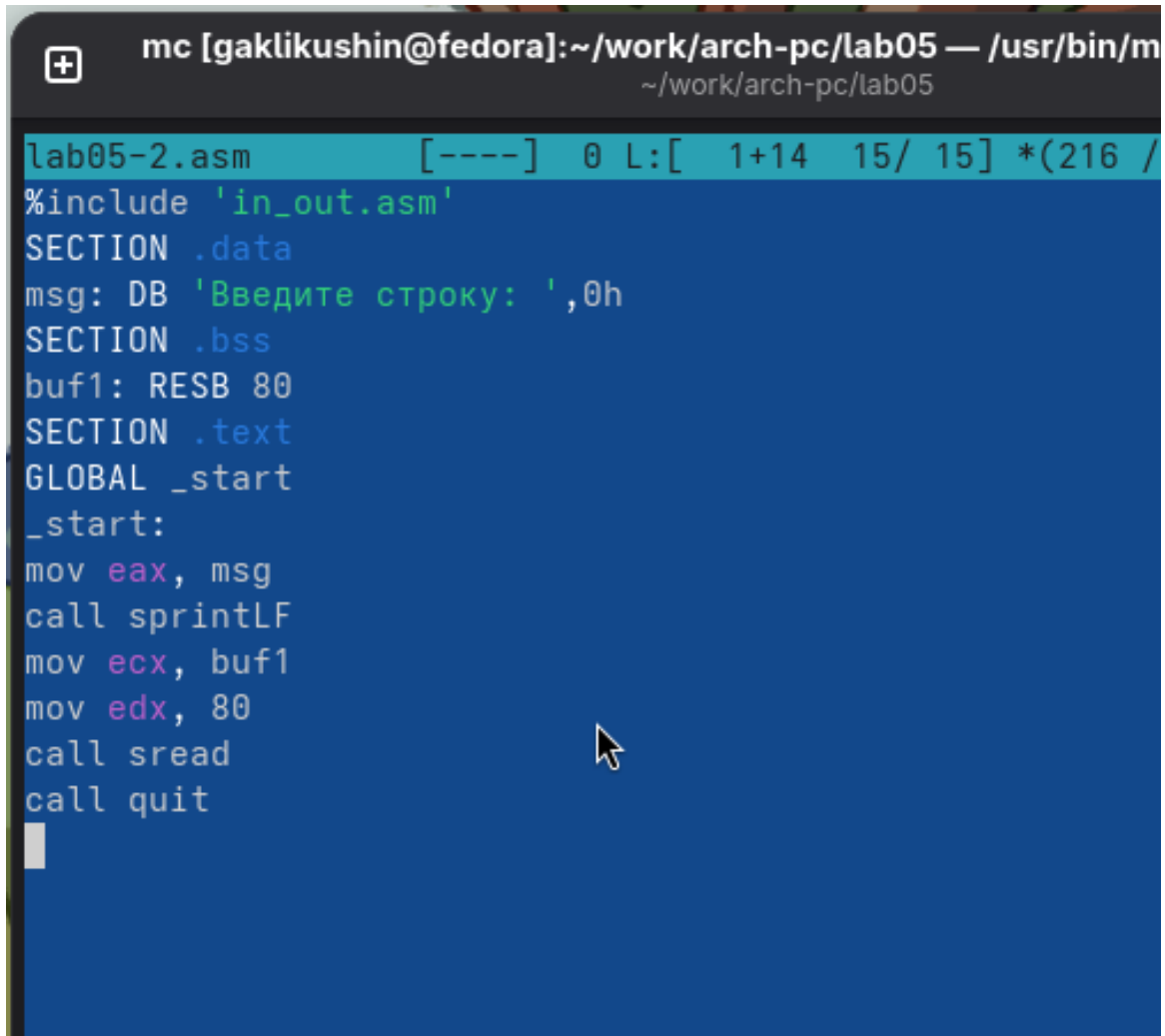


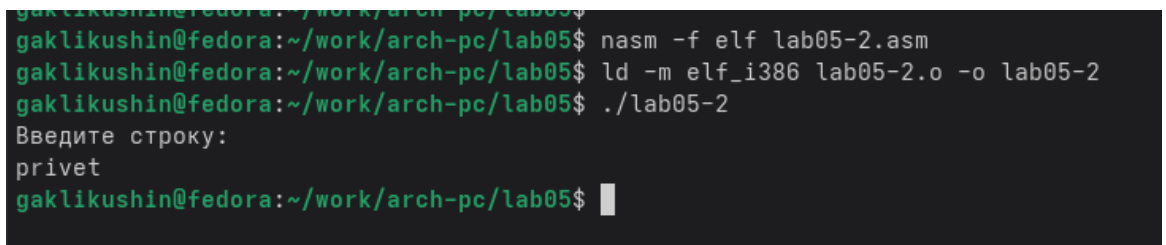
Рисунок 2.6: Копирование файла

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



```
mc [gaklikushin@fedora]:~/work/arch-pc/lab05 — /usr/bin/m
~/work/arch-pc/lab05
lab05-2.asm [----] 0 L:[ 1+14 15/ 15] *(216 /
#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 sprintf
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рисунок 2.7: Программа в файле lab05-2.asm



```
gaklikushin@fedora:~/work/arch-pc/lab05$
gaklikushin@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
gaklikushin@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
gaklikushin@fedora:~/work/arch-pc/lab05$ ./lab05-2
Введите строку:
privet
gaklikushin@fedora:~/work/arch-pc/lab05$
```

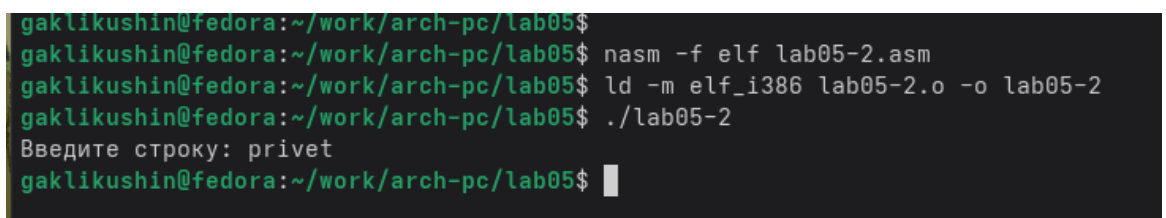
Рисунок 2.8: Запуск программы lab05-2.asm

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 [gaklikushin@fedora]:~/work/arch-pc/lab05 — /usr/bin/mc
~/work/arch-pc/lab05
lab05-2.asm [----] 11 L:[ 1+ 9 10/ 15] *(166 /
#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
```

Рисунок 2.9: Программа в файле lab05-2.asm

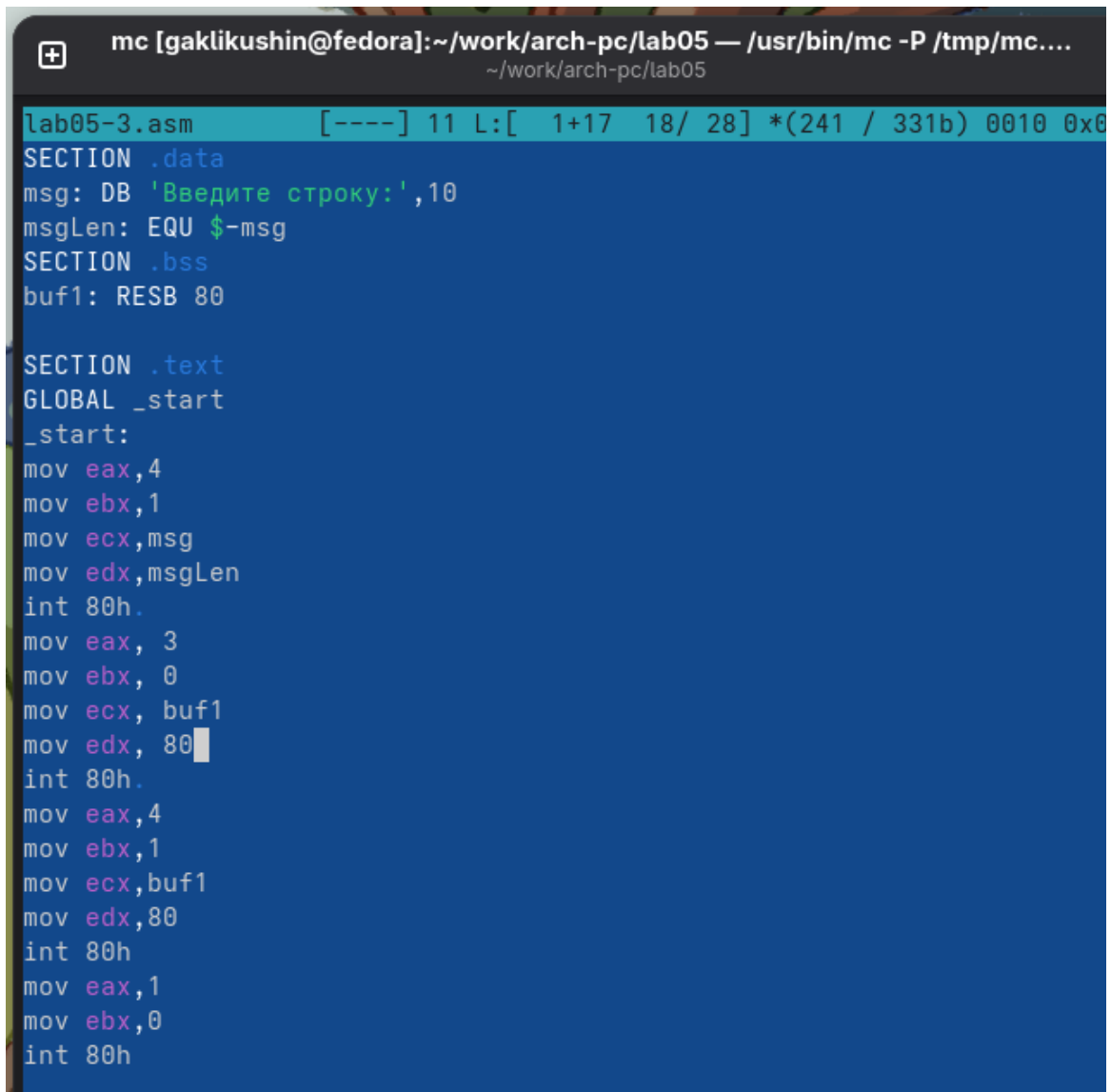


```
gaklikushin@fedora:~/work/arch-pc/lab05$
gaklikushin@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
gaklikushin@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
gaklikushin@fedora:~/work/arch-pc/lab05$ ./lab05-2
Введите строку: privet
gaklikushin@fedora:~/work/arch-pc/lab05$
```

Рисунок 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.



```
mc [gaklikushin@fedora]:~/work/arch-pc/lab05 — /usr/bin/mc -P /tmp/mc....
~/work/arch-pc/lab05
lab05-3.asm [----] 11 L:[ 1+17 18/ 28] *(241 / 331b) 0010 0x0
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
```

Рисунок 2.11: Программа в файле lab05-3.asm

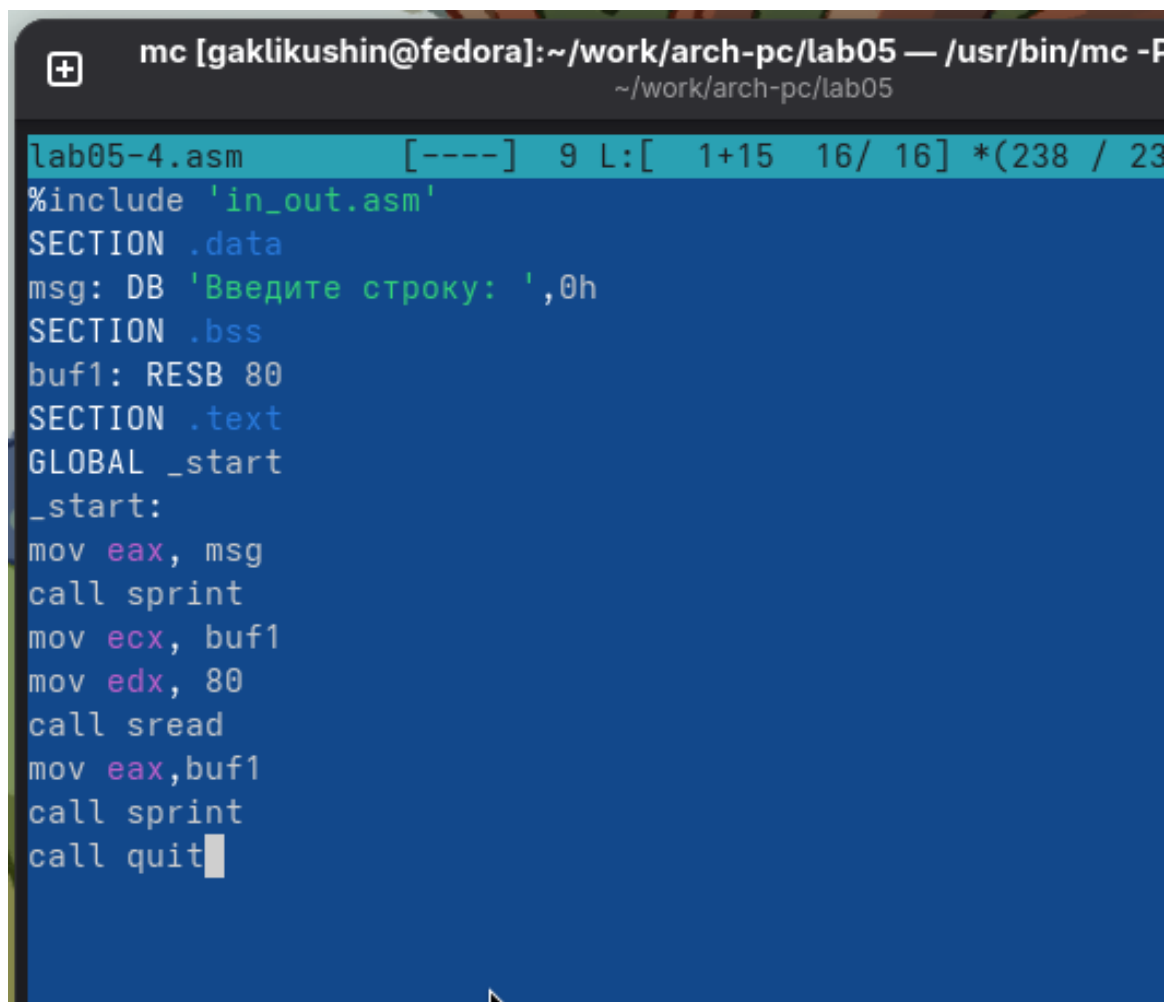
```

gaklikushin@fedora:~/work/arch-pc/lab05$
gaklikushin@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-3.asm
gaklikushin@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3
gaklikushin@fedora:~/work/arch-pc/lab05$ ./lab05-3
Введите строку:
privet
privet
gaklikushin@fedora:~/work/arch-pc/lab05$

```

Рисунок 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.



```

mc [gaklikushin@fedora]:~/work/arch-pc/lab05 — /usr/bin/mc -P
~/work/arch-pc/lab05

lab05-4.asm [----] 9 L:[ 1+15 16/ 16] *(238 / 23
%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

```

Рисунок 2.13: Программа в файле lab05-4.asm

```
gaklikushin@fedora:~/work/arch-pc/lab05$  
gaklikushin@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-4.asm  
gaklikushin@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-4.o -o lab05-4  
gaklikushin@fedora:~/work/arch-pc/lab05$ ./lab05-4  
Введите строку: privet  
privet  
gaklikushin@fedora:~/work/arch-pc/lab05$
```

Рисунок 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`.