

# **Отчёт по лабораторной работе 5**

**Архитектура ЭВМ**

Александр Архипов

# 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	Код lab05-1.asm . . . . .	7
2.3	Просмотр файла lab05-1.asm . . . . .	8
2.4	Запуск программы lab05-1.asm . . . . .	9
2.5	Копирование файла . . . . .	9
2.6	Код lab05-2.asm . . . . .	10
2.7	Запуск программы lab05-2.asm . . . . .	11
2.8	Код lab05-2.asm . . . . .	12
2.9	Запуск программы lab05-2.asm . . . . .	12
2.10	Код lab05-3.asm . . . . .	13
2.11	Запуск программы lab05-3.asm . . . . .	14
2.12	Код lab05-4.asm . . . . .	15
2.13	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. Я открыл Midnight Commander и перешел в каталог ~/work/arch-рс. Создал новый каталог под названием lab05.

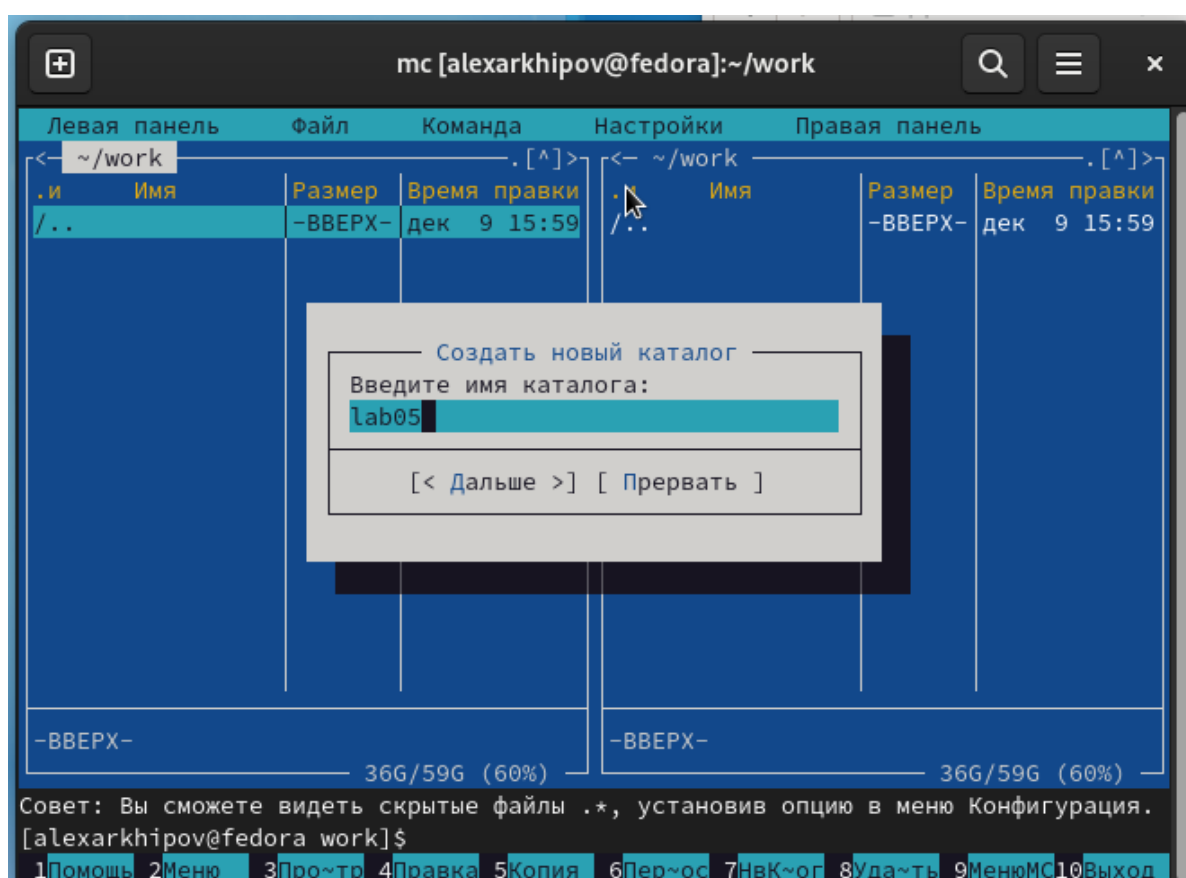
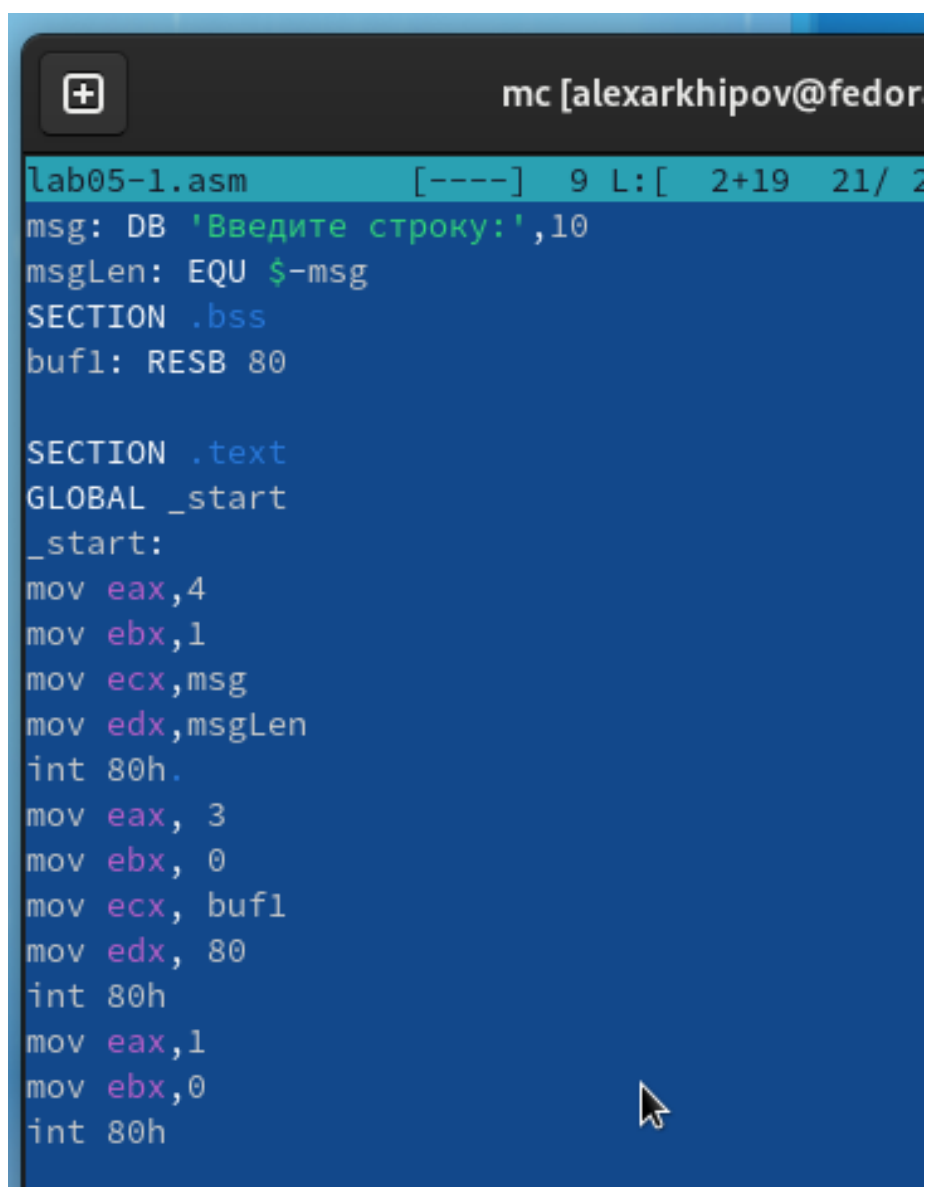


Рис. 2.1: Создание каталога

2. Внутри каталога lab05 я создал файл lab05-1.asm. Затем я открыл этот файл для редактирования и начал писать код.

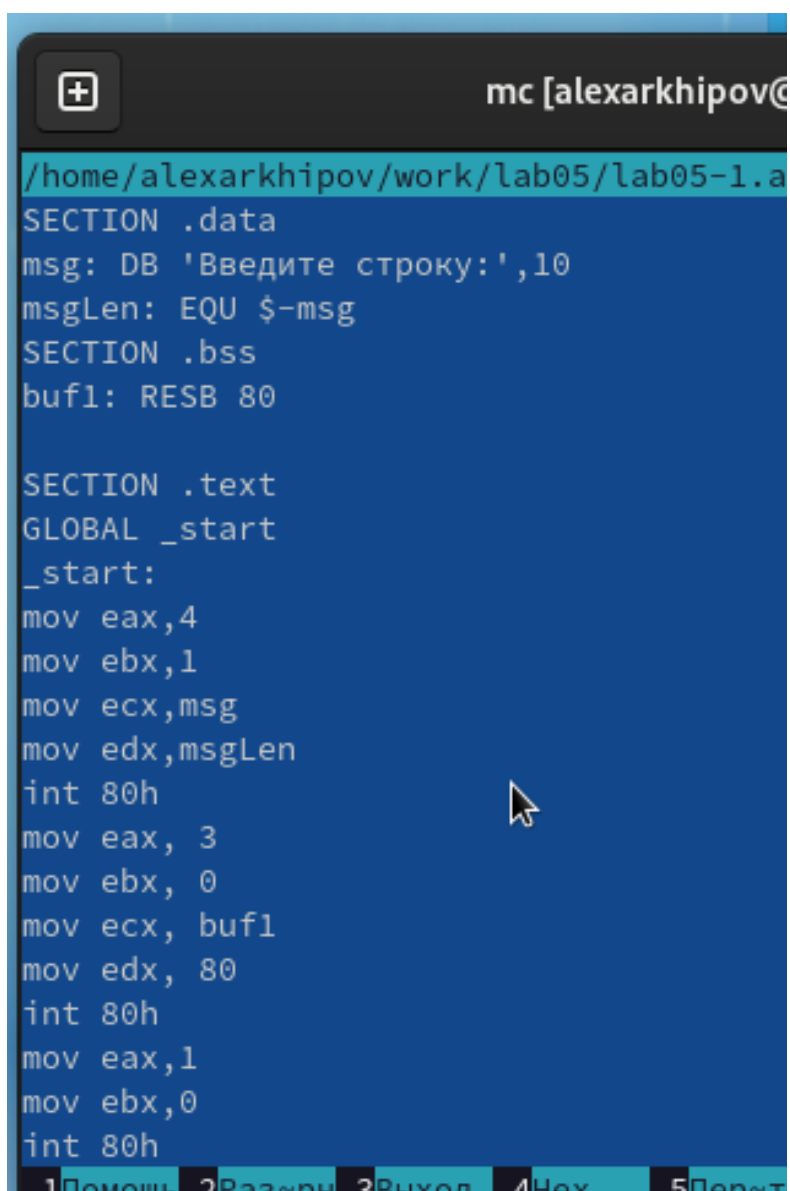


```
lab05-1.asm [----] 9 L: [ 2+19 21/ 2
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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. 2.2: Код lab05-1.asm

3. Я также открыл файл lab05-1.asm для просмотра и проверил, что код был написан правильно.



```
mc [alexarkhipov@
/home/alexarkhipov/work/lab05/lab05-1.a
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

1Помощь 2Разбор 3Выход 4Чех 5Перет
```

Рис. 2.3: Просмотр файла lab05-1.asm

4. После того, как я получил исполняемый файл из кода, я проверил его работу, чтобы убедиться, что все функционирует должным образом.



```

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

```

Рис. 2.4: Запуск программы lab05-1.asm

5. Я скачал файл in\_out.asm и добавил его в рабочий каталог. Затем я скопировал содержимое файла lab05-1.asm в новый файл под названием lab05-2.asm.

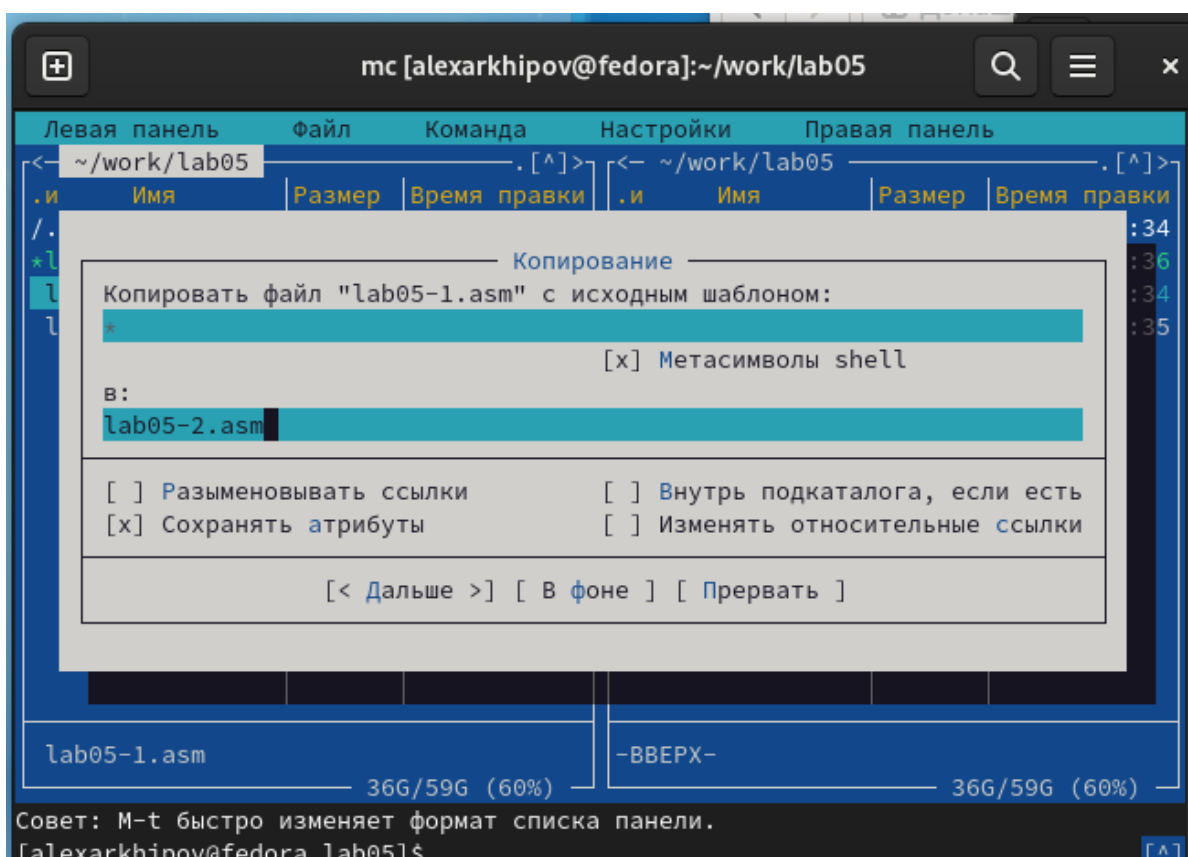
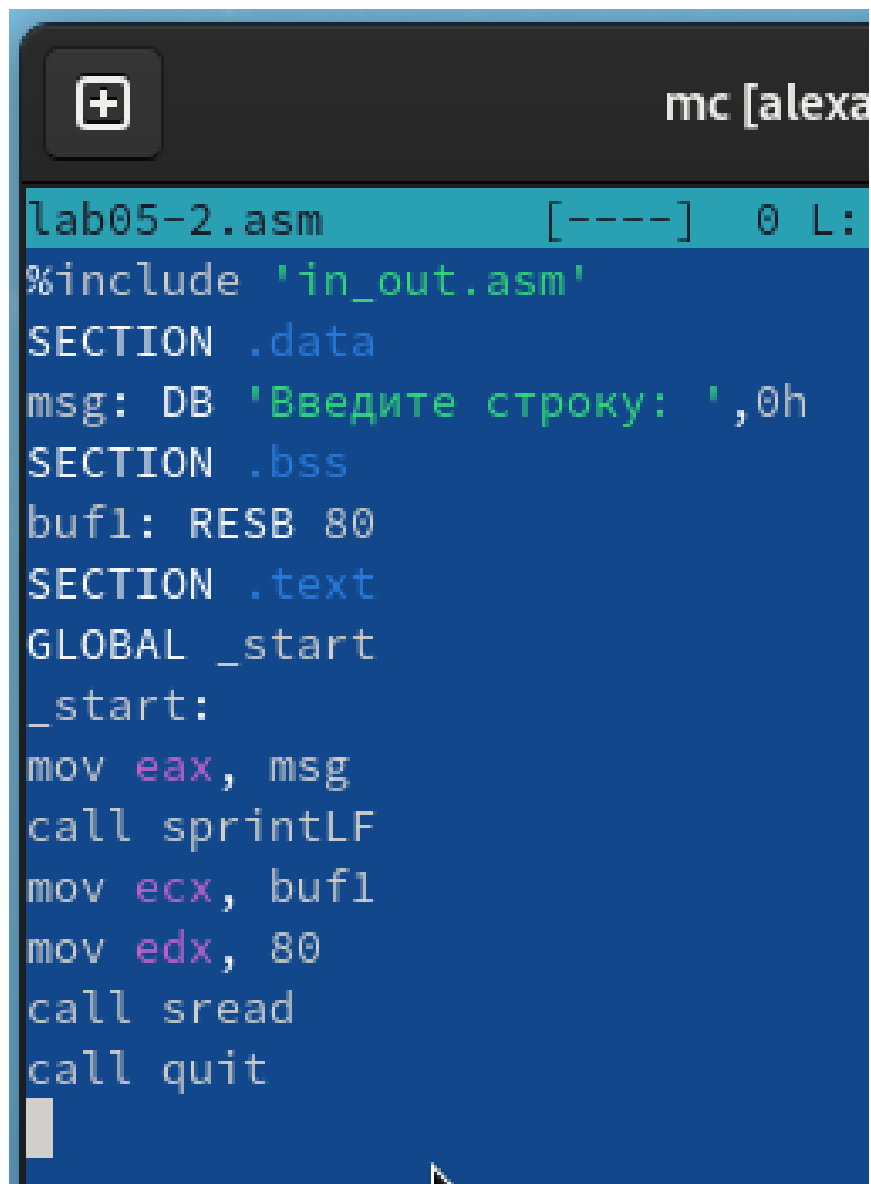


Рис. 2.5: Копирование файла

6. В файле lab05-2.asm я написал код программы. После этого я скомпилировал

программу и проверил ее запуск.



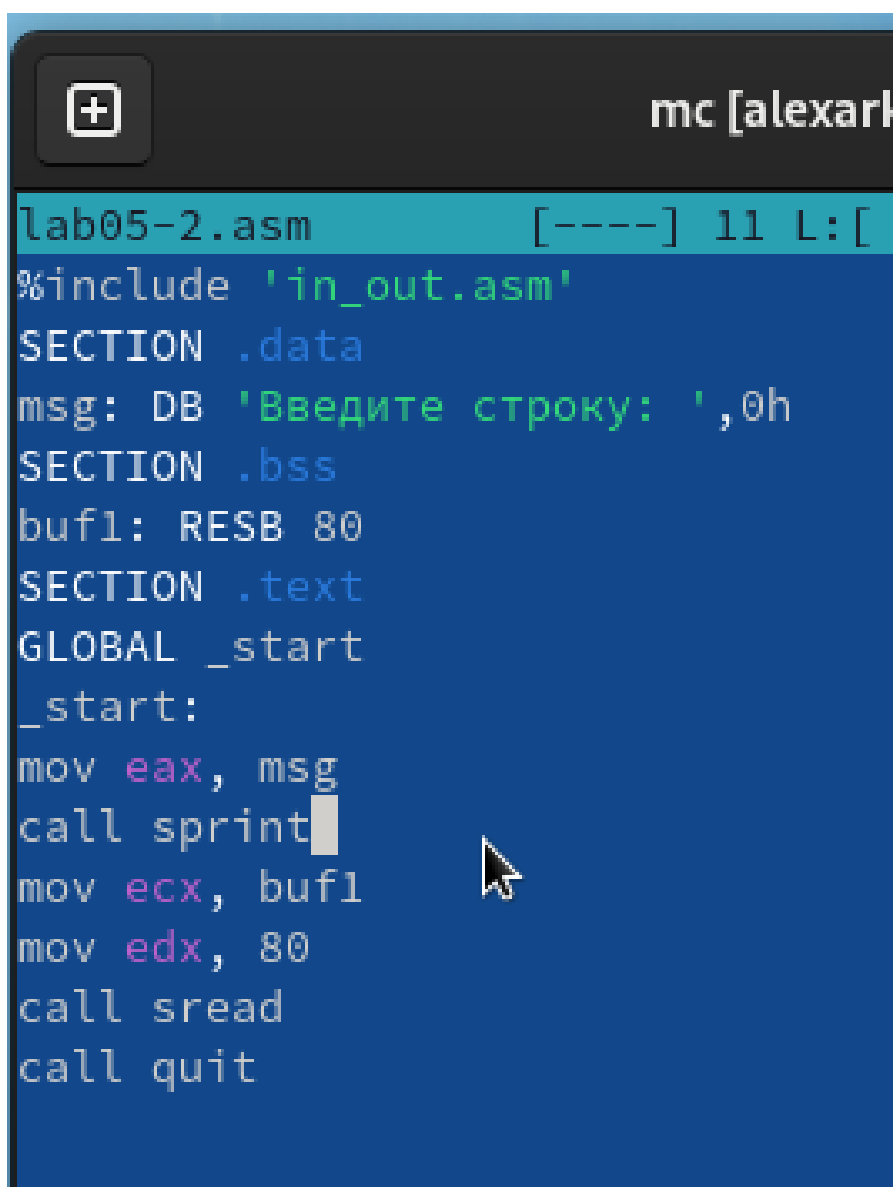
```
lab05-2.asm [-----] 0 L:
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку: ',0h
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintLF
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рис. 2.6: Код lab05-2.asm

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

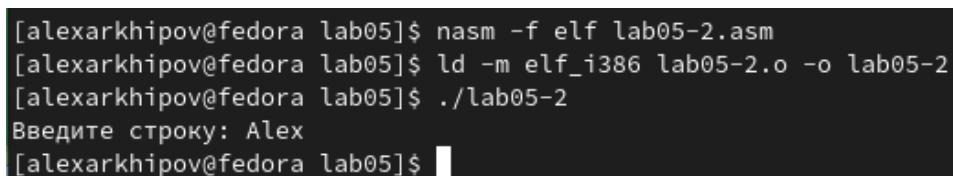
Рис. 2.7: Запуск программы lab05-2.asm

7. Внутри файла lab05-2.asm я заменил подпрограмму sprintLF на sprint. Затем я пересобрал исполняемый файл. Теперь при выводе строки нет перехода на следующую строку.



```
lab05-2.asm [----] 11 L: [
#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
```

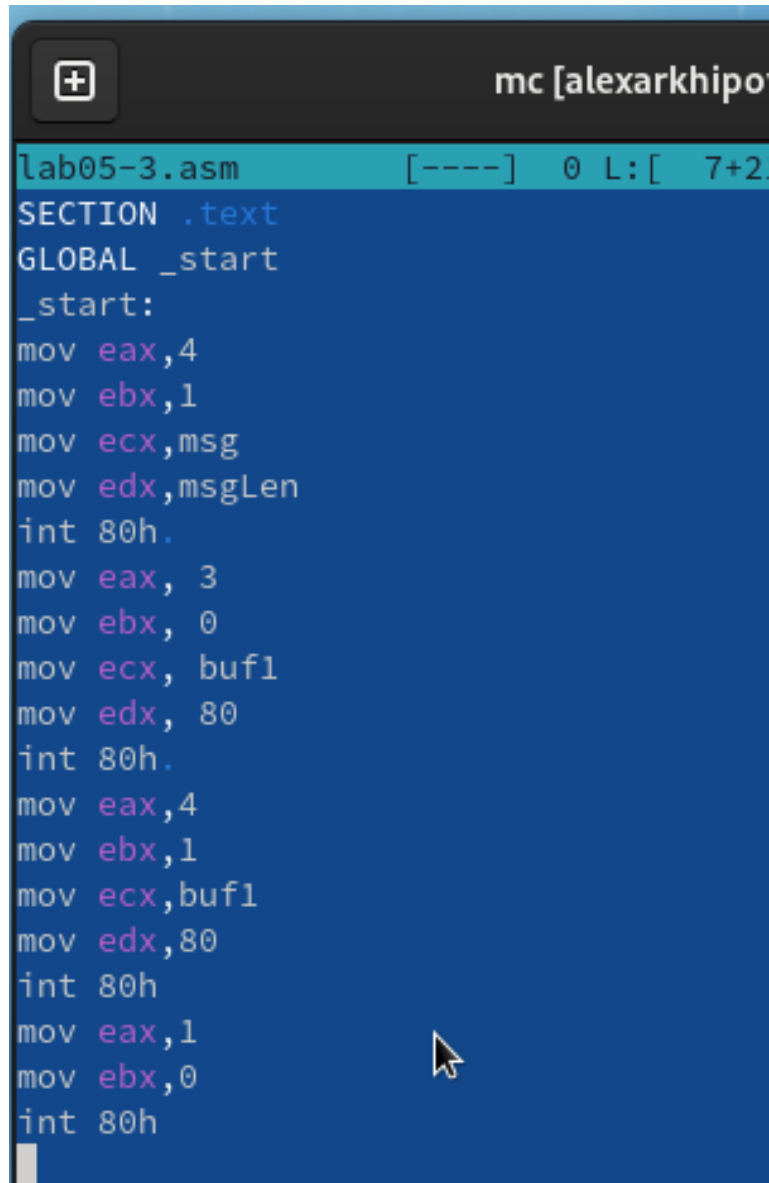
Рис. 2.8: Код lab05-2.asm



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

Рис. 2.9: Запуск программы lab05-2.asm

8. Я скопировал программу lab05-1.asm и изменил код так, чтобы сначала выводилось приглашение “Введите строку:”, затем пользователь вводил строку с клавиатуры, и введенная строка выводилась на экран.



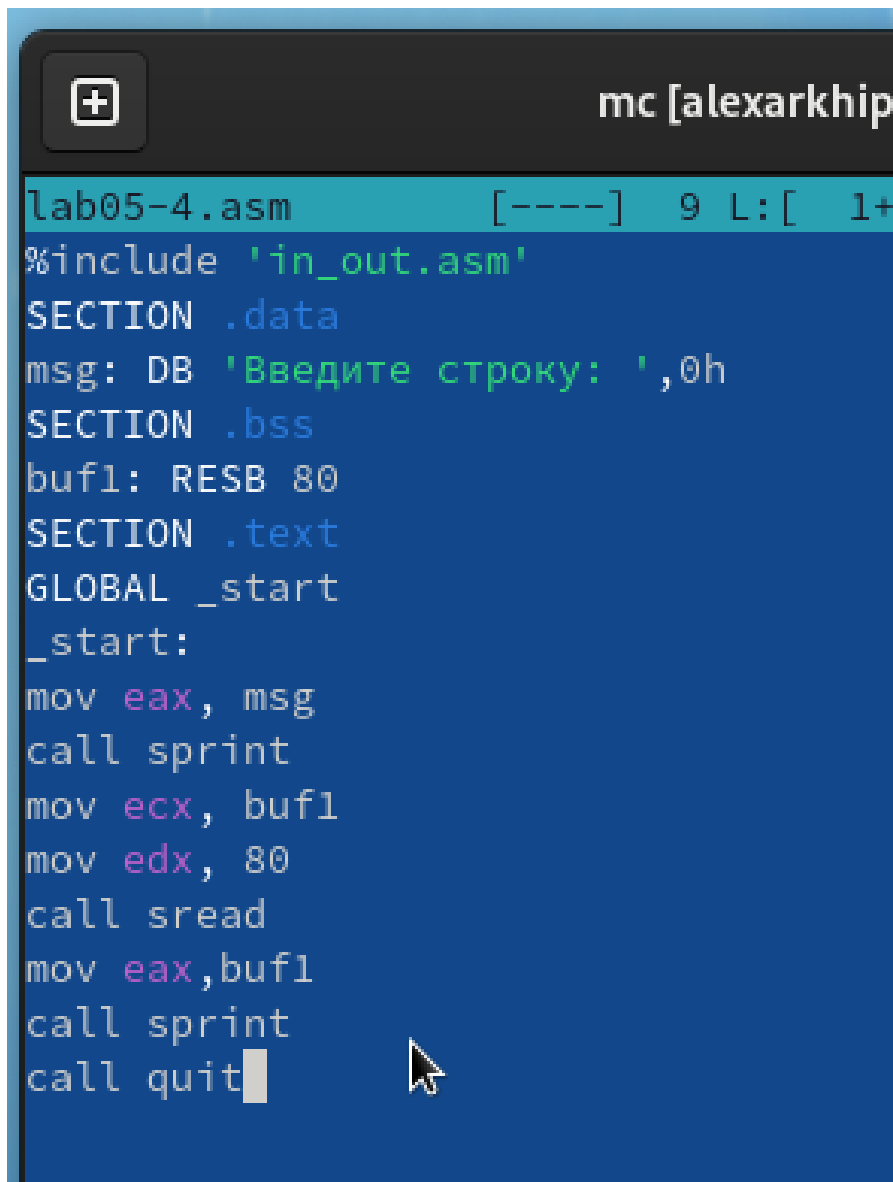
```
mc [alexarkhipo]
lab05-3.asm [----] 0 L: [ 7+2
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h.
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h.
mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,80
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рис. 2.10: Код lab05-3.asm

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

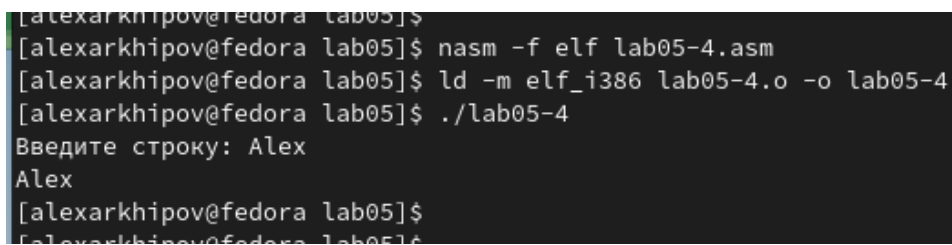
Рис. 2.11: Запуск программы lab05-3.asm

9. Затем я скопировал программу lab05-2.asm и выполнил аналогичные действия, описанные выше, но теперь использовал возможности из файла in\_out.asm.



```
lab05-4.asm [----] 9 L: [ 1+
#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. 2.12: Код lab05-4.asm



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

Рис. 2.13: Запуск программы lab05-4.asm

## 3 Выводы

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