

Отчёт по лабораторной работе 5

Архитектура компьютеров и операционные системы

Алина Молокова

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Знакомство с Midnight Commander	6
3.2	Подключение внешнего файла in_out.asm	12
3.3	Задание для самостоятельной работы	17
4	Выводы	21

Список иллюстраций

3.1	Запускаю Midnight Commander	7
3.2	Нажимаю F7, чтобы создать каталог lab05	8
3.3	создаю файл lab05-1.asm	9
3.4	Редактирую файл lab05-1.asm	10
3.5	Просмотр файла lab05-1.asm	11
3.6	Компиляция программы lab05-1.asm	12
3.7	Запуск программы lab05-1.asm	12
3.8	Перемещаю файл in_out.asm	13
3.9	Копирую файл lab05-1.asm	14
3.10	Редактирую файл lab05-2.asm	15
3.11	Запуск программы lab05-2.asm	16
3.12	Редактирую файл lab05-2.asm	16
3.13	Запуск программы lab05-2.asm	17
3.14	Редактирую файл lab05-3.asm	18
3.15	Запуск программы lab05-3.asm	19
3.16	Редактирую файл lab05-4.asm	20
3.17	Запуск программы lab05-4.asm	20

1 Цель работы

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

2 Задание

1. Установить и изучить Midnight Commander
2. Написать программы без файла in_out.asm
3. Написать программы с файлом in_out.asm

3 Выполнение лабораторной работы

3.1 Знакомство с Midnight Commander

Запускаю Midnight Commander (рис. [3.1]), с помощью кнопок со стрелками и Enter перехожу в каталог `~/work/arch-рс`. Далее нажимаю F7, чтобы создать каталог `lab05` (рис. [3.2]).

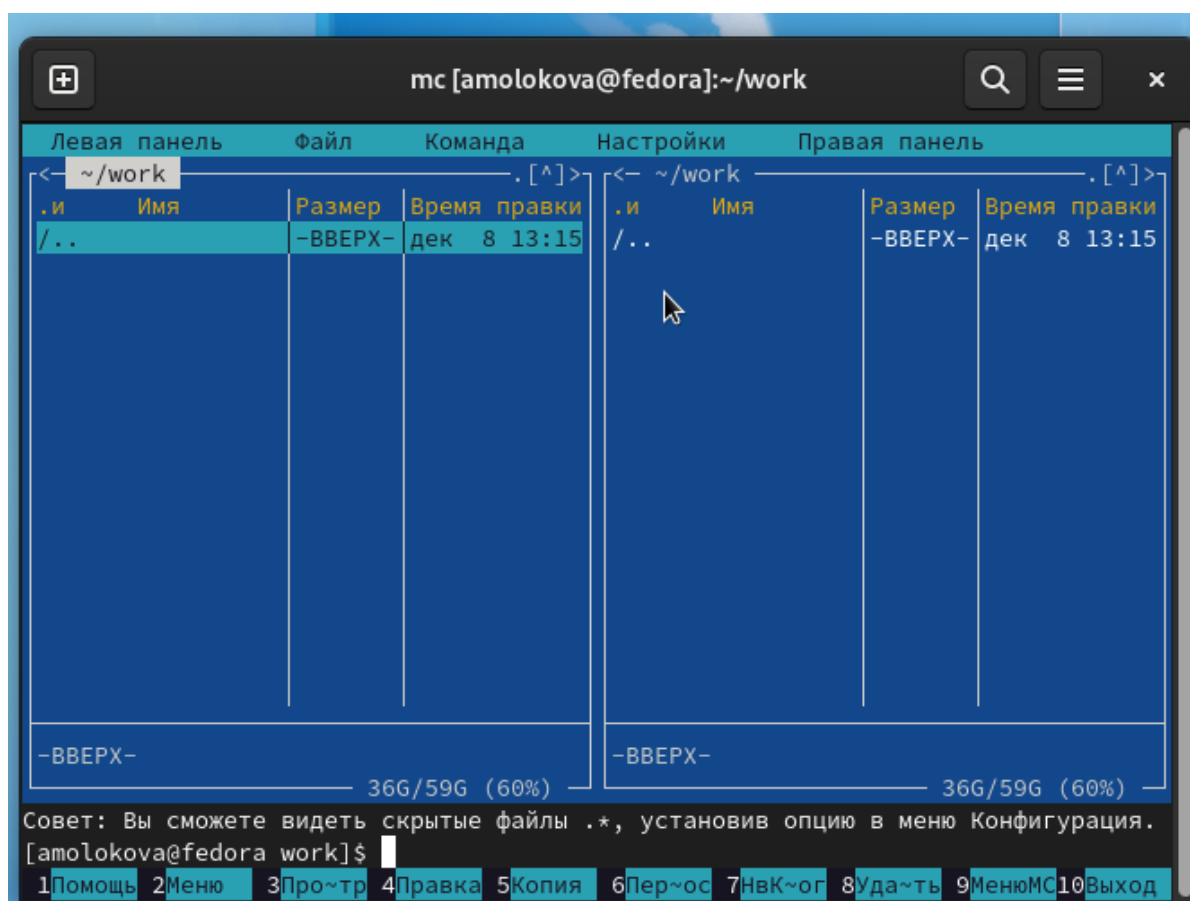


Рис. 3.1: Запускаю Midnight Commander

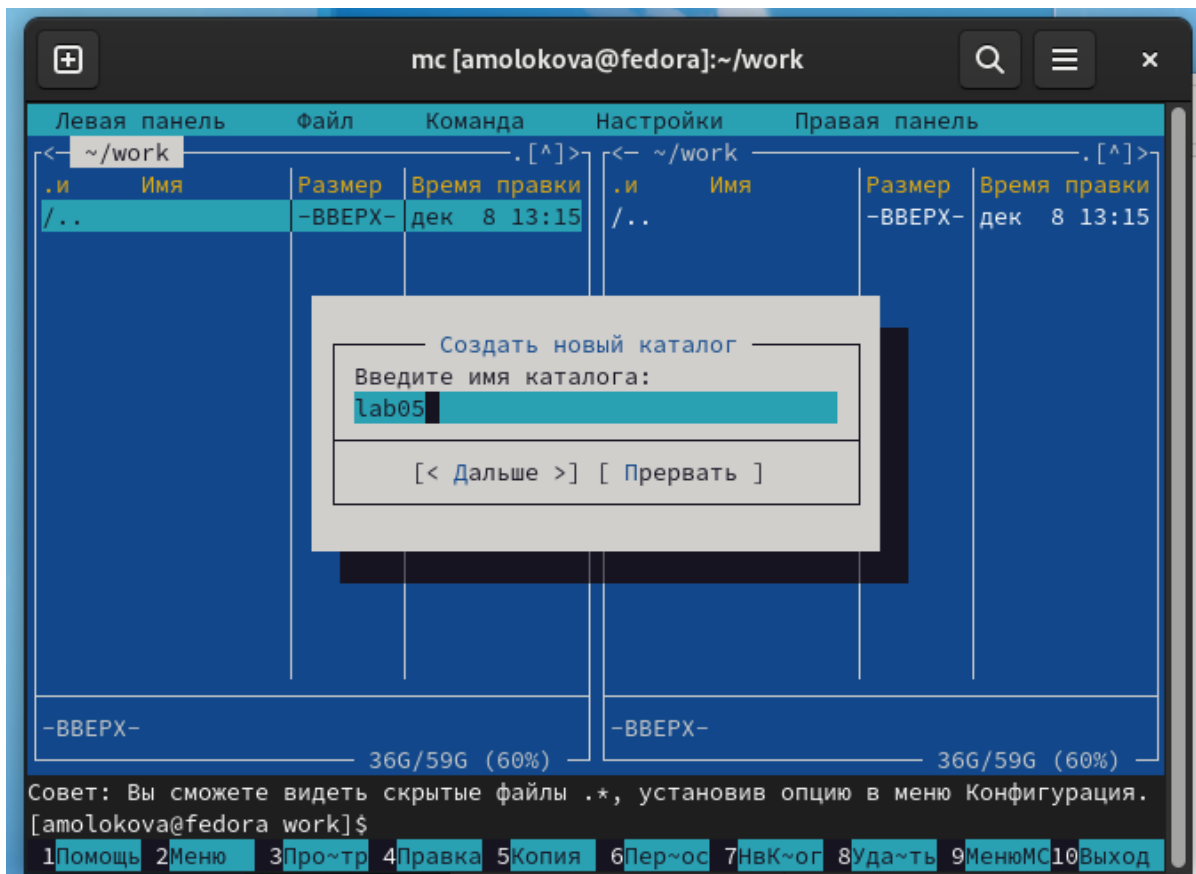


Рис. 3.2: Нажимаю F7, чтобы создать каталог lab05

При помощи команды touch создаю файл lab05-1.asm (рис. [3.3])

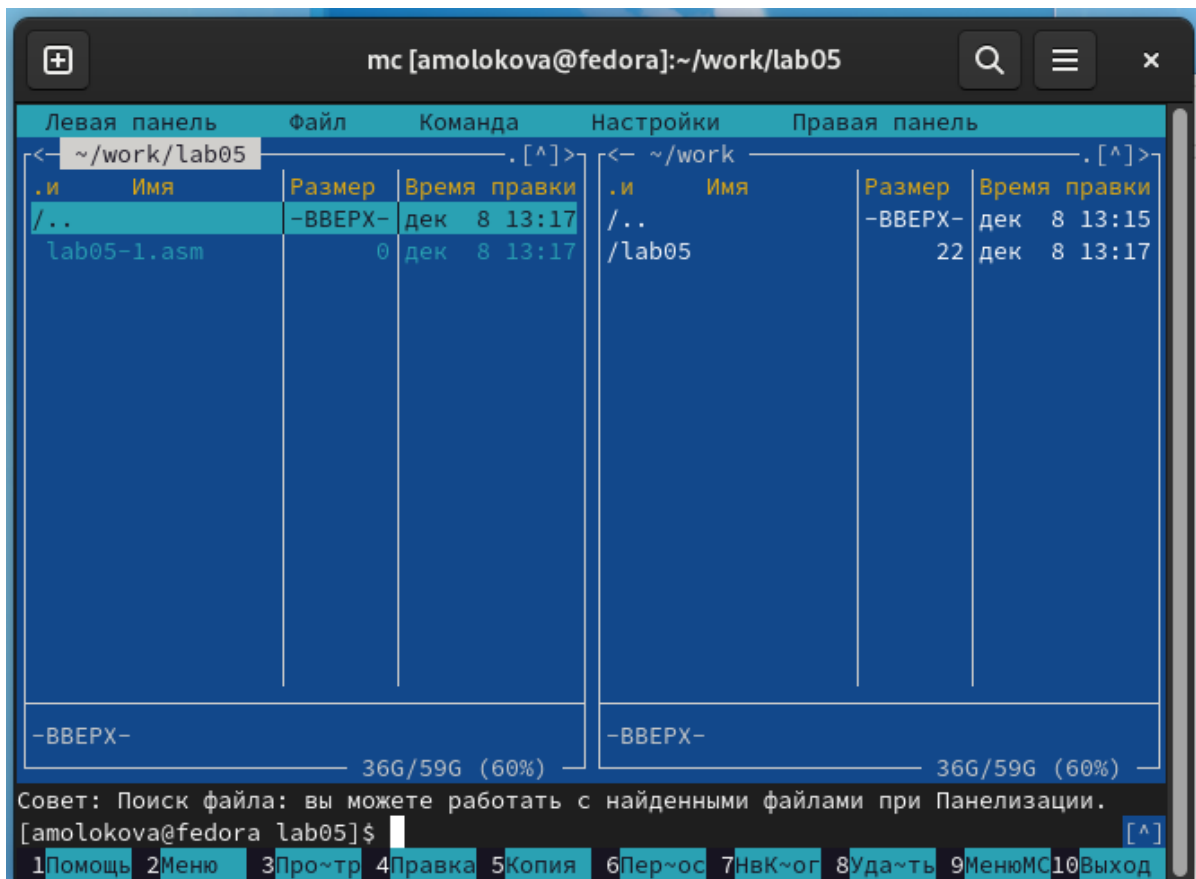
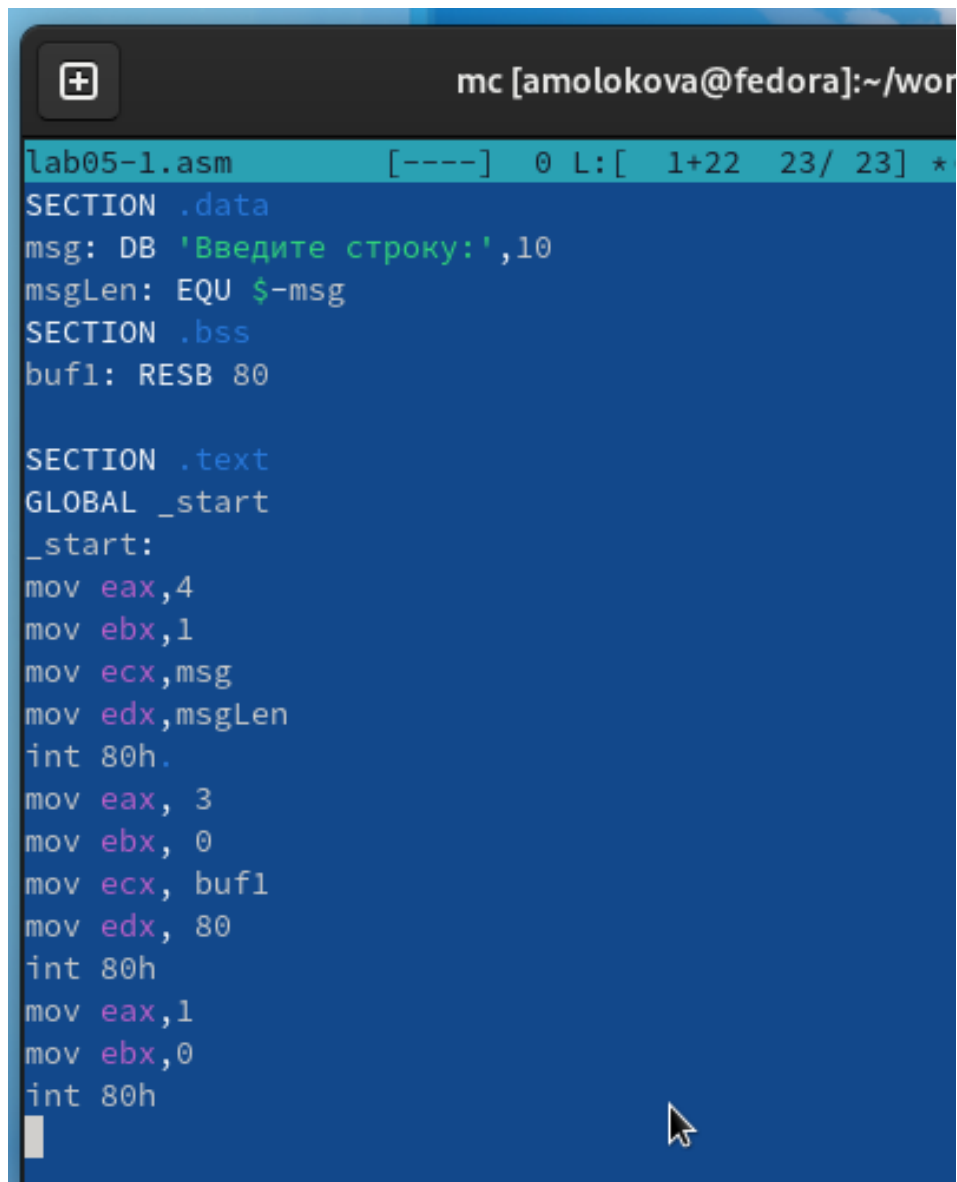


Рис. 3.3: создаю файл lab05-1.asm

Открываю файл на редактирование клавишей F4, выбираю редактор mcedit, пишу код программы из задания. (рис. [3.4])

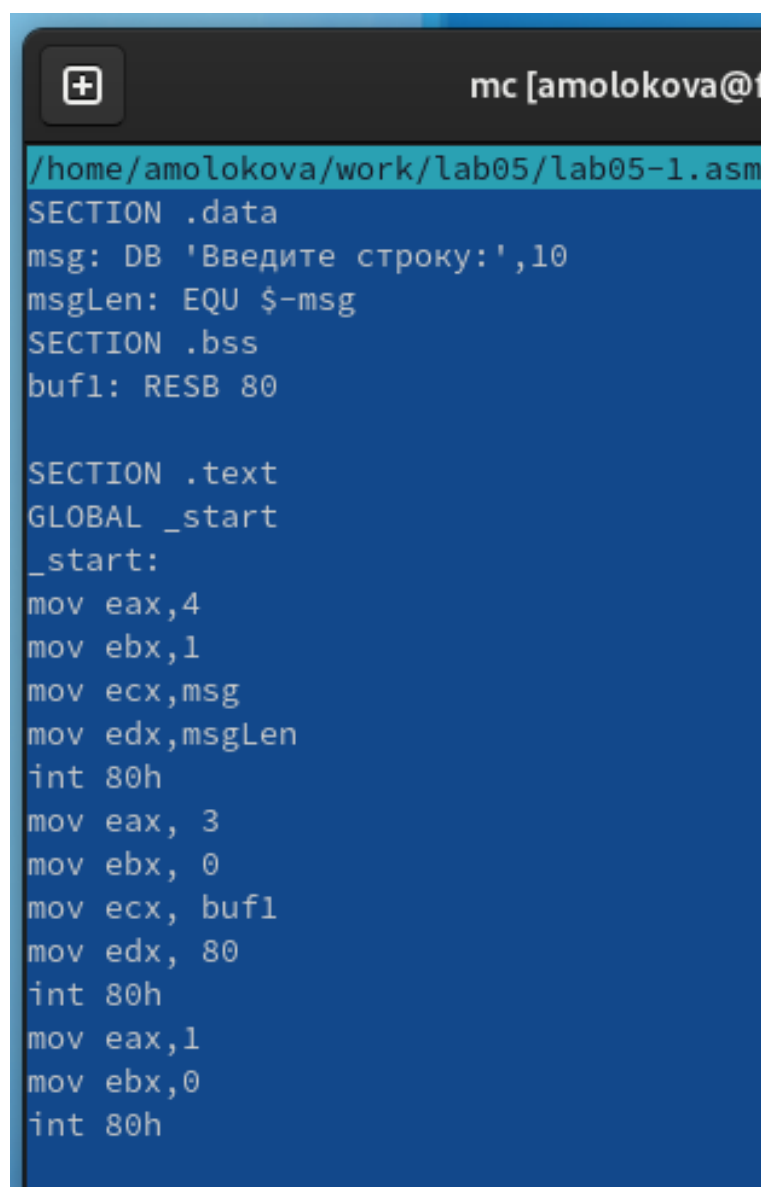


```
mc [amolokova@fedora]:~/wor
lab05-1.asm [----] 0 L: [ 1+22 23/ 23] *
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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. 3.4: Редактирую файл lab05-1.asm

Открываю файл на просмотр клавишей F3 и проверяю, что он содержит набранный код. (рис. [3.5])



```
mc [amolokova@...]  
/home/amolokova/work/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
```

Рис. 3.5: Просмотр файла lab05-1.asm

Транслирую файл программы в объектный файл, выполняю компоновку объектного файла, получился исполняемый файл программы. (рис. [3.6]) (рис. [3.7])

```
[amolokova@fedora lab05]$  
[amolokova@fedora lab05]$ nasm -f elf lab05-1.asm  
[amolokova@fedora lab05]$ ld -m elf_i386 lab05-1.o -o lab05-1  
[amolokova@fedora lab05]$
```

Рис. 3.6: Компиляция программы lab05-1.asm

```
[amolokova@fedora lab05]$  
[amolokova@fedora lab05]$ ./lab05-1  
Введите строку:  
Alina  
[amolokova@fedora lab05]$
```

Рис. 3.7: Запуск программы lab05-1.asm

3.2 Подключение внешнего файла in_out.asm

Скачиваю файл in_out.asm и размещаю его в рабочем каталоге. (рис. [3.8]) Для копирования используется клавиша F5. Для перемещения используется клавиша F6.

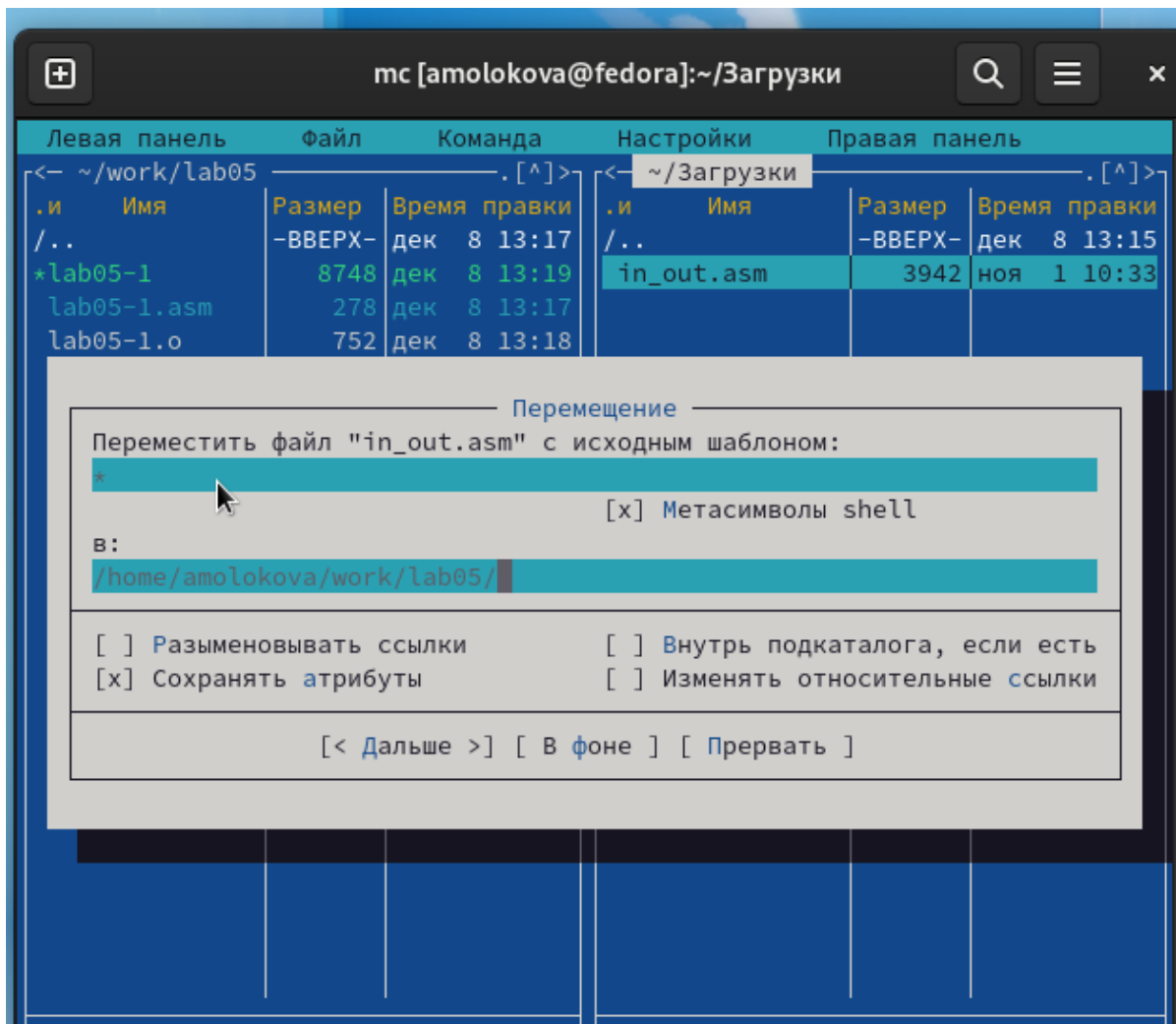


Рис. 3.8: Перемещаю файл in_out.asm

Скопировала lab05-1.asm в lab05-2.asm. (рис. [3.9])

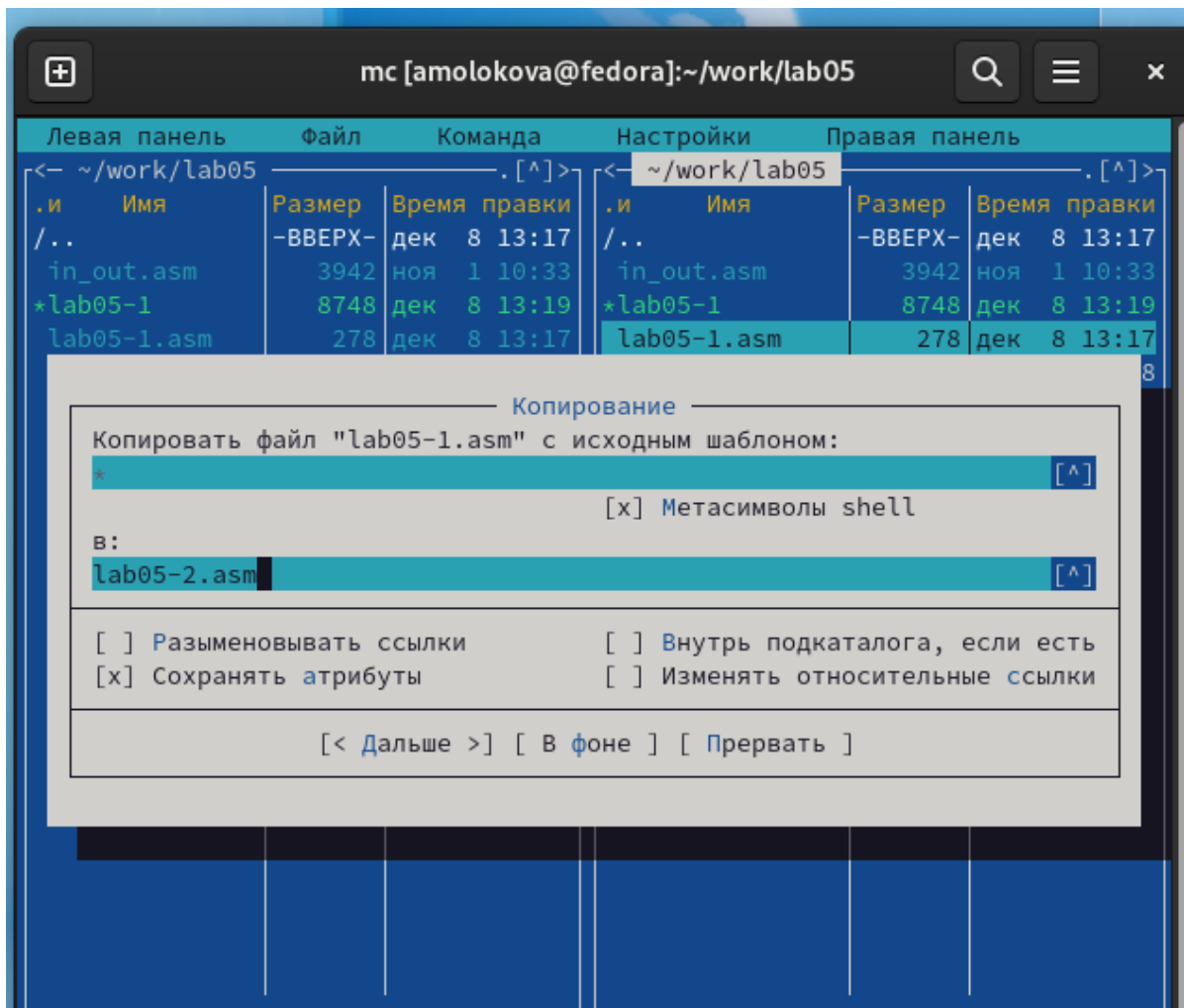
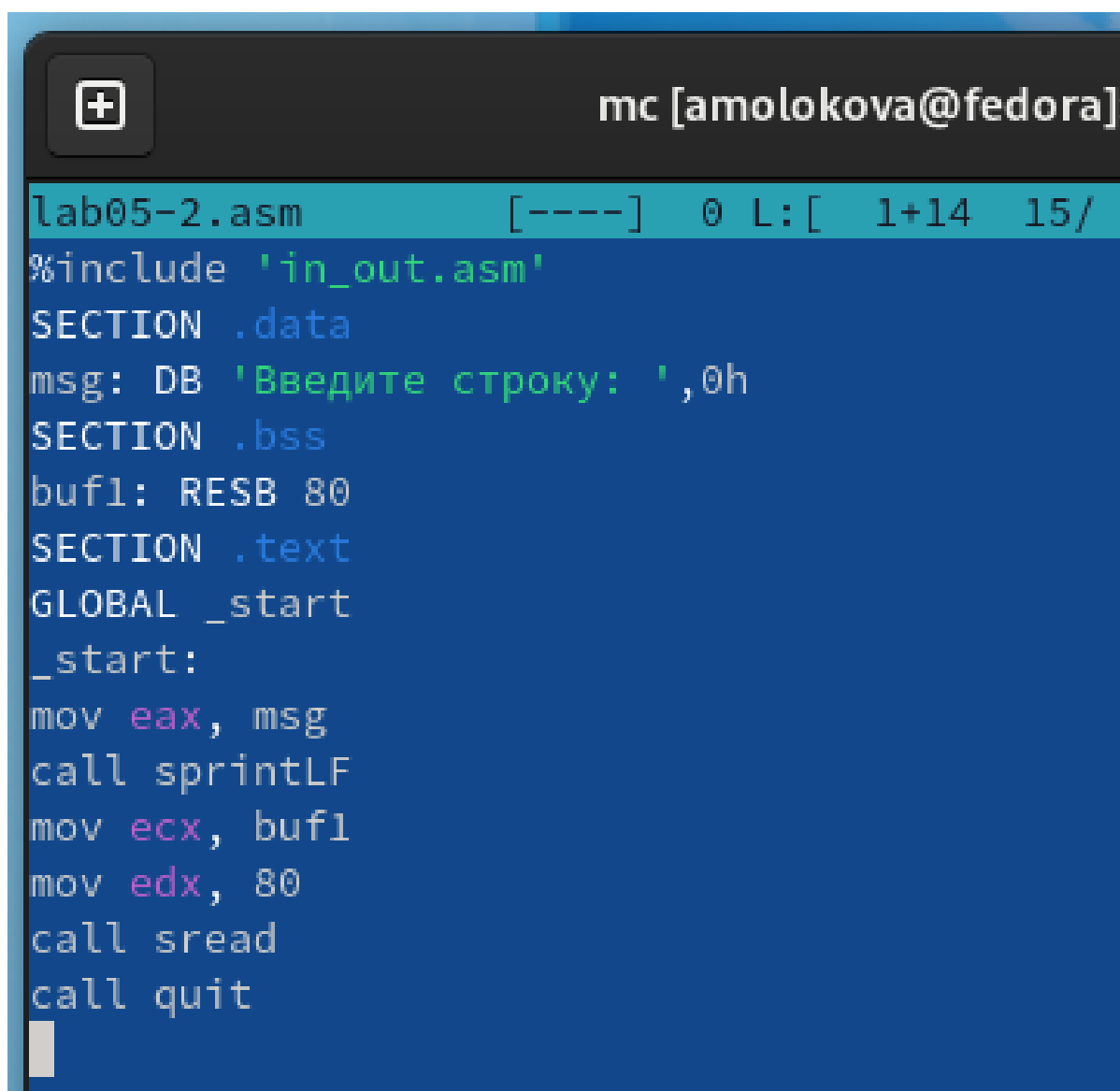


Рис. 3.9: Копирую файл lab05-1.asm

Пишу код программы lab05-2.asm с использованием подпрограмм из внешнего файла in_out.asm. (рис. [3.10])



```
lab05-2.asm [----] 0 L: [ 1+14 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 sprintLF
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рис. 3.10: Редактирую файл lab05-2.asm

Скомпилирую программу и проверю запуск. (рис. [3.11])

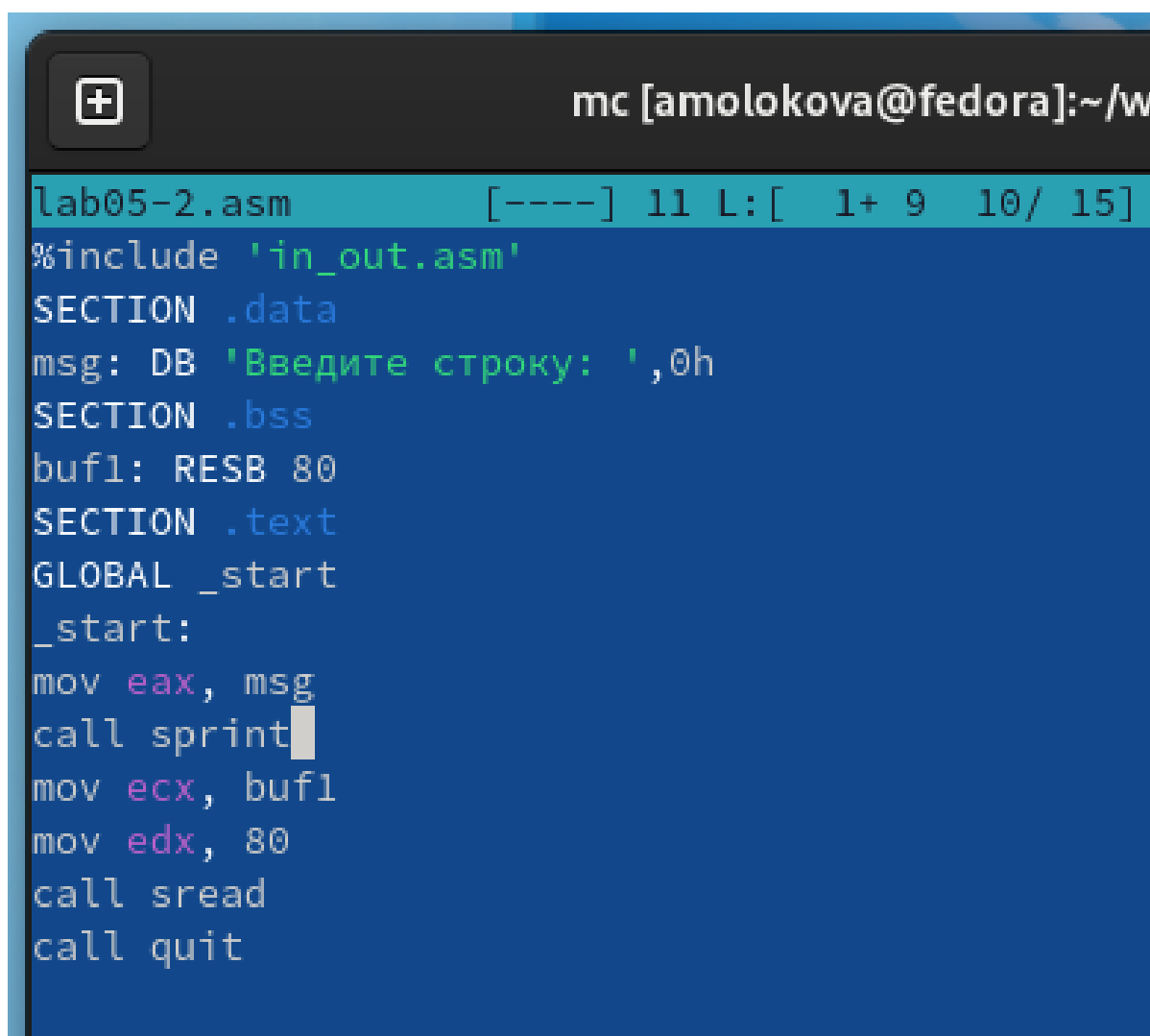
```

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

```

Рис. 3.11: Запуск программы lab05-2.asm

В файле lab5-2.asm заменила подпрограмму sprintLF на sprint. Заново собрала исполняемый файл. (рис. [3.12]) (рис. [3.13])



```

mc [amolokova@fedora]:~/w
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. 3.12: Редактирую файл lab05-2.asm


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

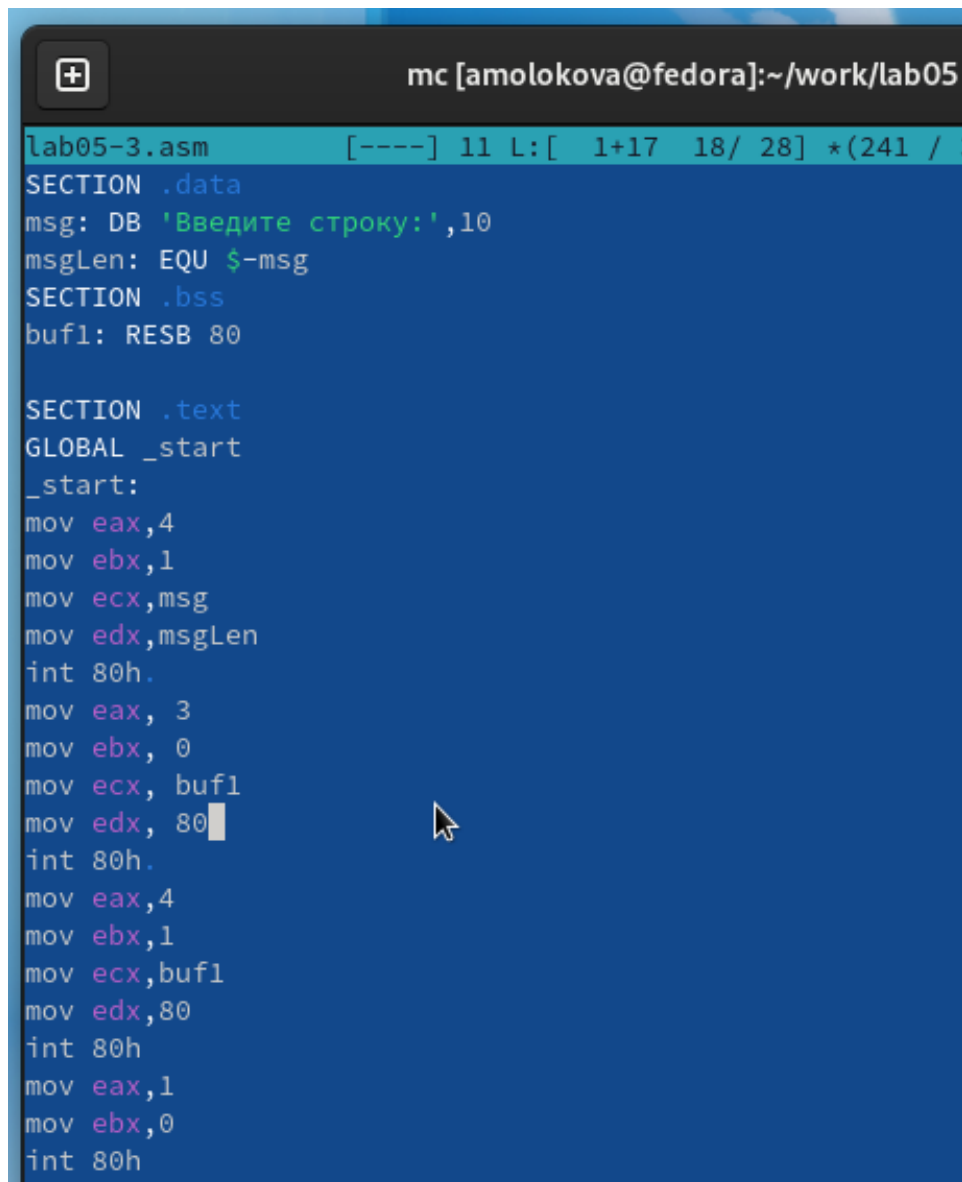
Рис. 3.13: Запуск программы lab05-2.asm

Теперь после вывода строки она не завершается символом перехода на новую строку.

3.3 Задание для самостоятельной работы

Скопировала программу lab05-1.asm и изменила код, так чтобы она работала по следующему алгоритму: (рис. [3.14]) (рис. [3.15])

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;
- вывести введенную строку на экран.



```
mc [amolokova@fedora]:~/work/lab05
lab05-3.asm [----] 11 L:[ 1+17 18/ 28] *(241 / 3
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

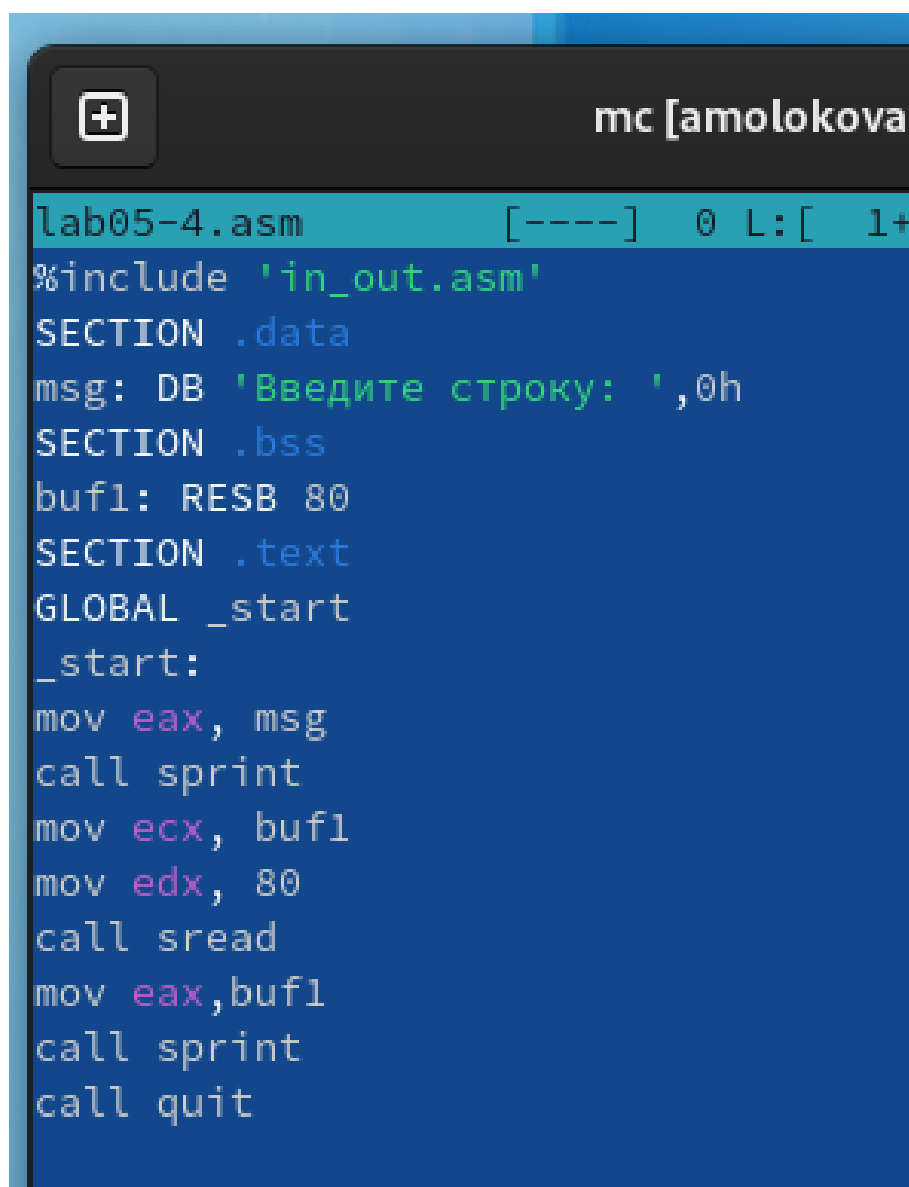
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h.
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h.
mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,80
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рис. 3.14: Редактирую файл lab05-3.asm

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

Рис. 3.15: Запуск программы lab05-3.asm

Аналогично скопировала программу lab05-2.asm и изменила код, но теперь использовал подпрограммы из файла in_out.asm. (рис. [3.16]) (рис. [3.17])



```
lab05-4.asm [----] 0 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. 3.16: Редактирую файл lab05-4.asm

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

Рис. 3.17: Запуск программы lab05-4.asm

4 Выводы

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