

Отчёт по лабораторной работе 5

Дисциплина: архитектура компьютера

Давлетова Мадина

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Знакомство с Midnight Commander	8
4.2	Подключение внешнего файла in_out.asm	14
4.3	Выполнение заданий для самостоятельной работы	19
5	Выводы	23

Список иллюстраций

4.1	окно Midnight Commander	9
4.2	Создание каталога	10
4.3	Создала файл lab05-1.asm	11
4.4	Изменение кода lab05-1.asm	12
4.5	Проверка кода lab05-1.asm	13
4.6	Компиляция текста программы lab05-1.asm	14
4.7	Копирование файла in_out.asm	15
4.8	Копирование файла lab05-1.asm	16
4.9	Изменение кода lab05-2.asm	17
4.10	Компиляция текста программы lab05-2.asm	17
4.11	Изменение кода lab05-2.asm	18
4.12	Компиляция текста программы lab05-2.asm	18
4.13	Изменение кода lab05-3.asm	19
4.14	Компиляция текста программы lab05-3.asm	20
4.15	Изменение кода lab05-4.asm	21
4.16	Компиляция текста программы lab05-4.asm	22

Список таблиц

1 Цель работы

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

2 Задание

1. Изучение возможностей Midnight Commander
2. Изучение файла in_out.asm
3. Выполнение заданий, рассмотрение примеров
4. Выполнение заданий для самостоятельной работы

3 Теоретическое введение

Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной.

Программа на языке ассемблера NASM, как правило, состоит из трёх секций: секция кода программы (SECTION .text), секция инициированных (известных во время компиляции) данных (SECTION .data) и секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение присваивается в ходе выполнения программы) (SECTION .bss).

4 Выполнение лабораторной работы

4.1 Знакомство с Midnight Commander

Я открыла Midnight Commander (рис. [4.1]) и с помощью клавиш со стрелками и Enter перешла в каталог `~/work/arch-рс`. Затем я нажала F7 и создала каталог lab05 (рис. [4.2]).

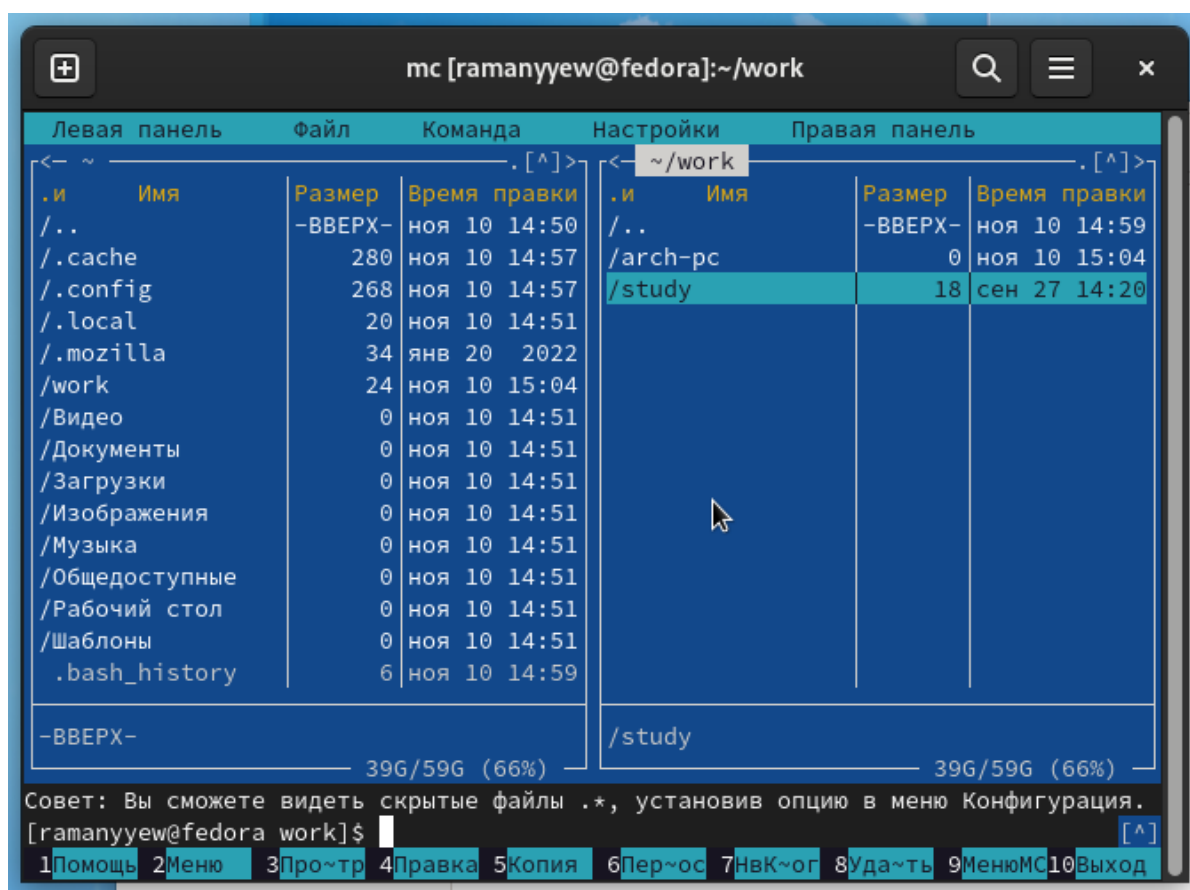


Рис. 4.1: окно Midnight Commander

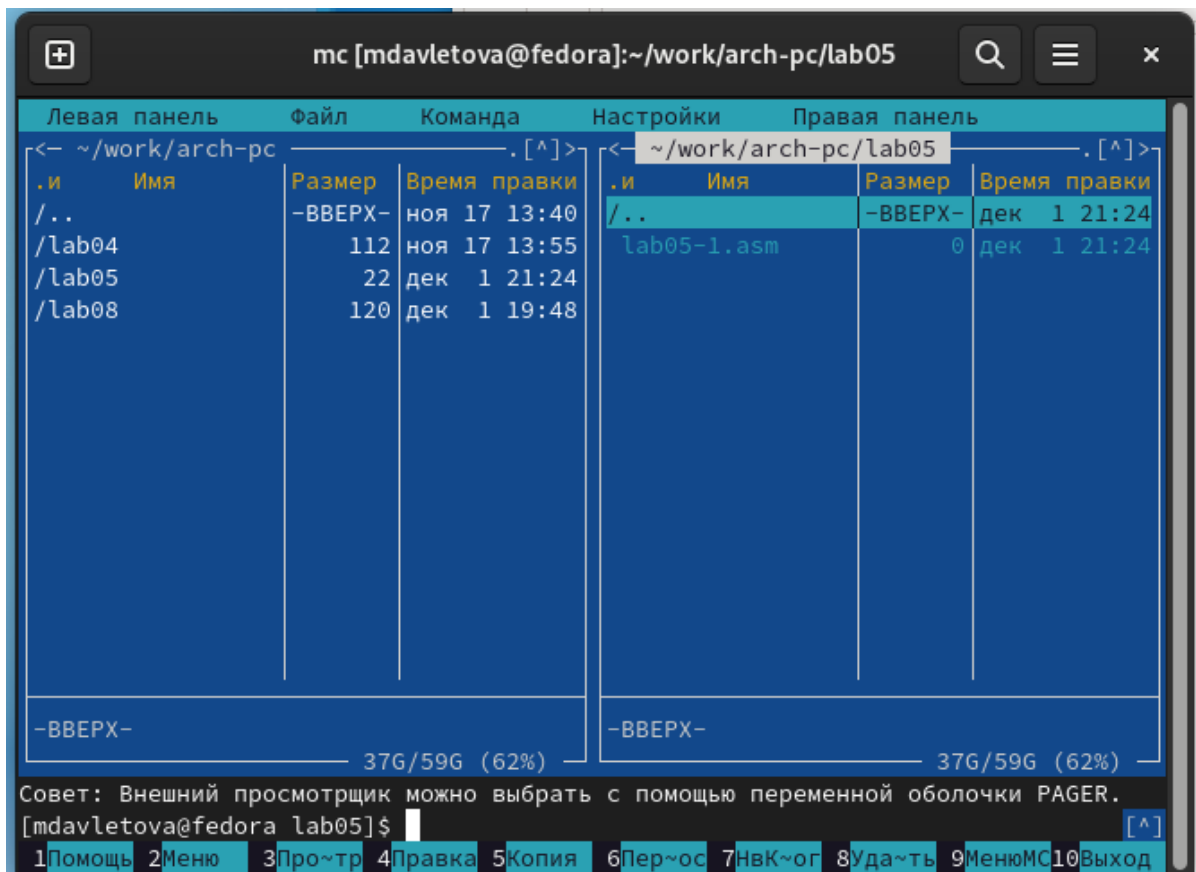
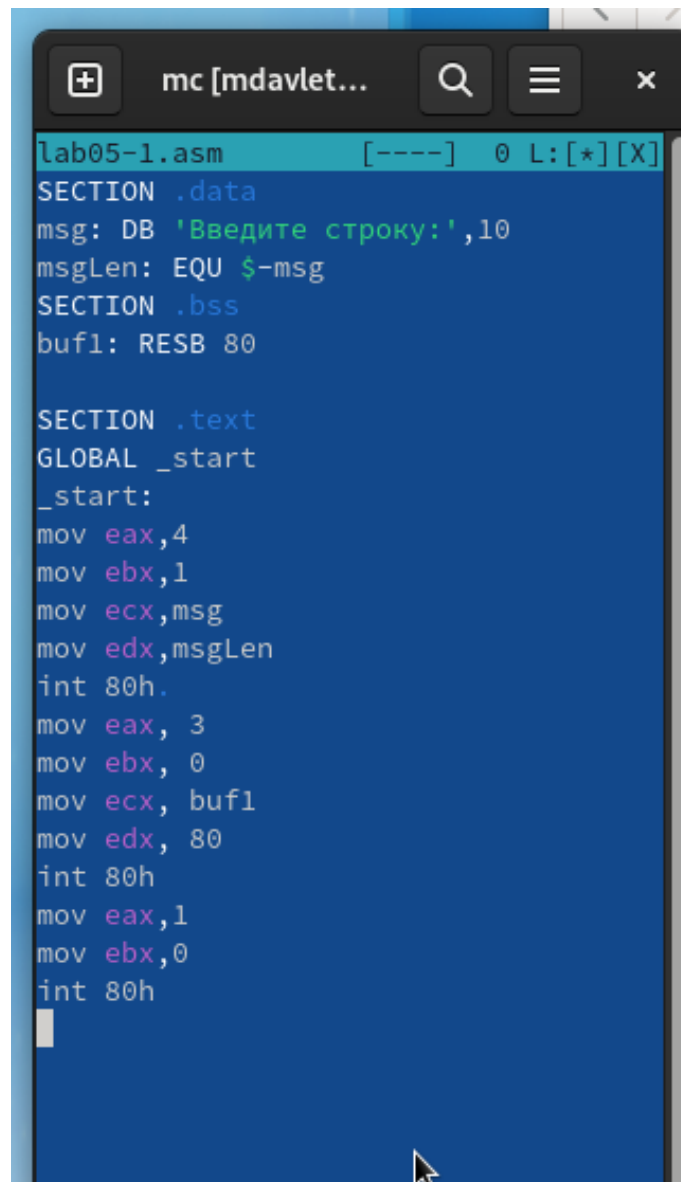


Рис. 4.3: Создала файл lab05-1.asm

Затем я открыла файл для редактирования, нажав клавишу F4, и выбрала редактор mceditor. Написала код программы, соответствующий заданию (рис. [4.4]).

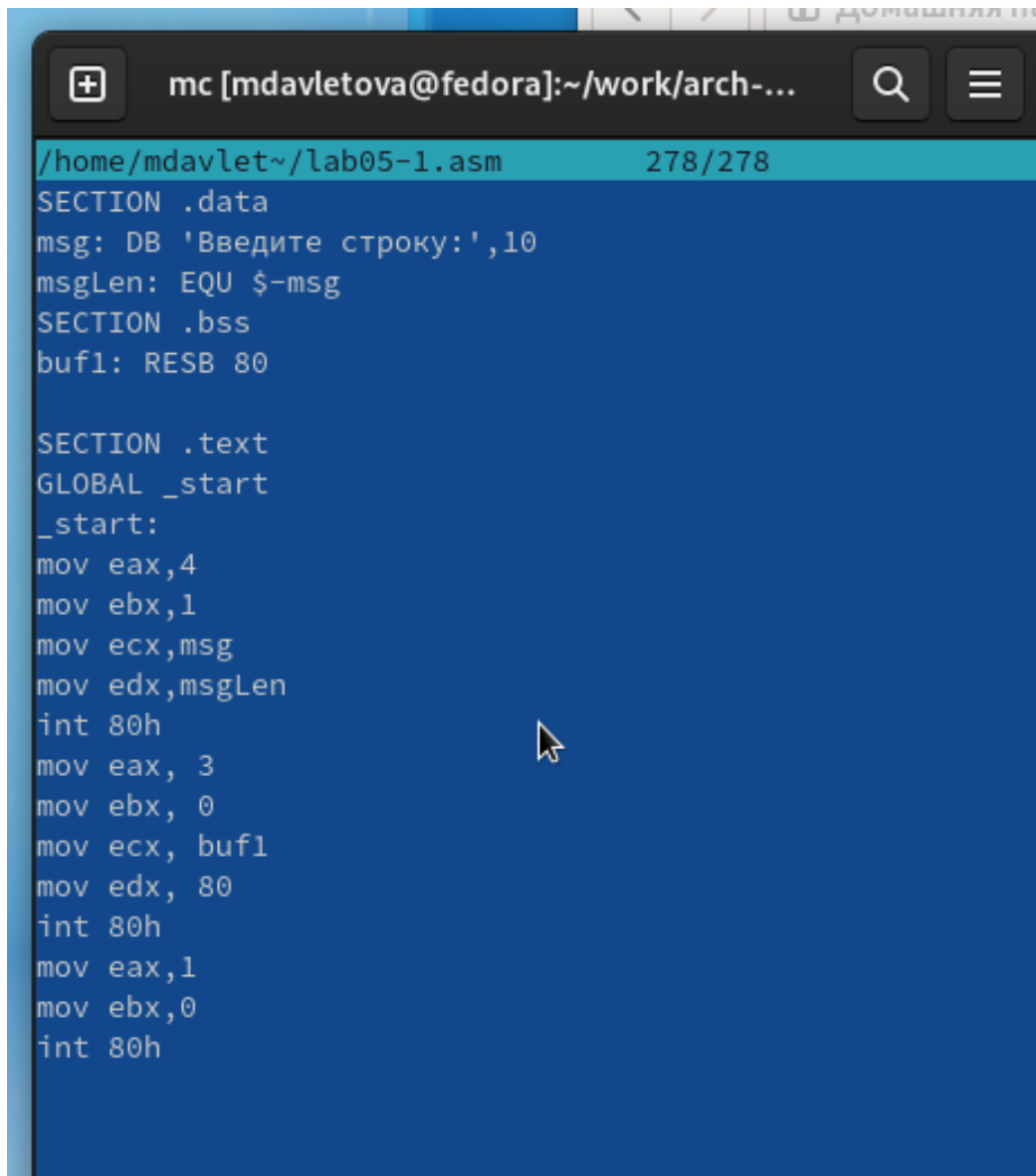


```
lab05-1.asm [----] 0 L: [*] [X]
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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. 4.4: Изменение кода lab05-1.asm

Далее я открыла файл для просмотра, нажав клавишу F3, и убедилась, что он содержит написанный код (рис. [4.5]).



```
mc [mdavletova@fedora]:~/work/arch-...
/home/mdavlet~ /lab05-1.asm 278/278
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. 4.5: Проверка кода lab05-1.asm

С помощью трансляции файла программы в объектный файл, выполнения компоновки объектного файла и получения исполняемого файла, я проверила работу программы (рис. [4.6]).

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

Рис. 4.6: Компиляция текста программы lab05-1.asm

4.2 Подключение внешнего файла in_out.asm

Я скачала файл in_out.asm (рис. [4.7]) и разместила его в рабочем каталоге.

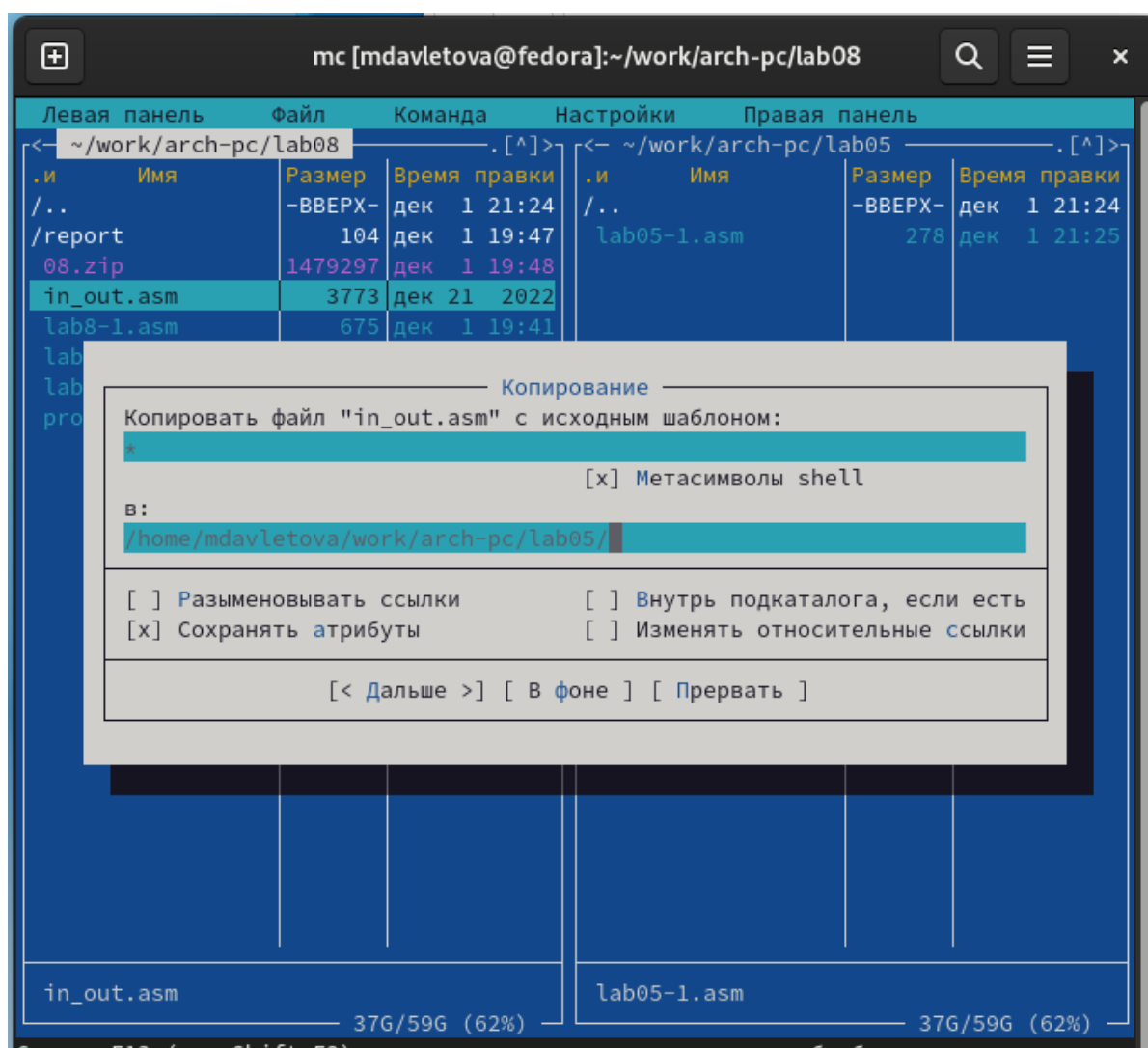


Рис. 4.7: Копирование файла in_out.asm

С помощью клавиши F5 скопировала содержимое файла lab05-1.asm в файл lab05-2.asm (рис. [4.8])

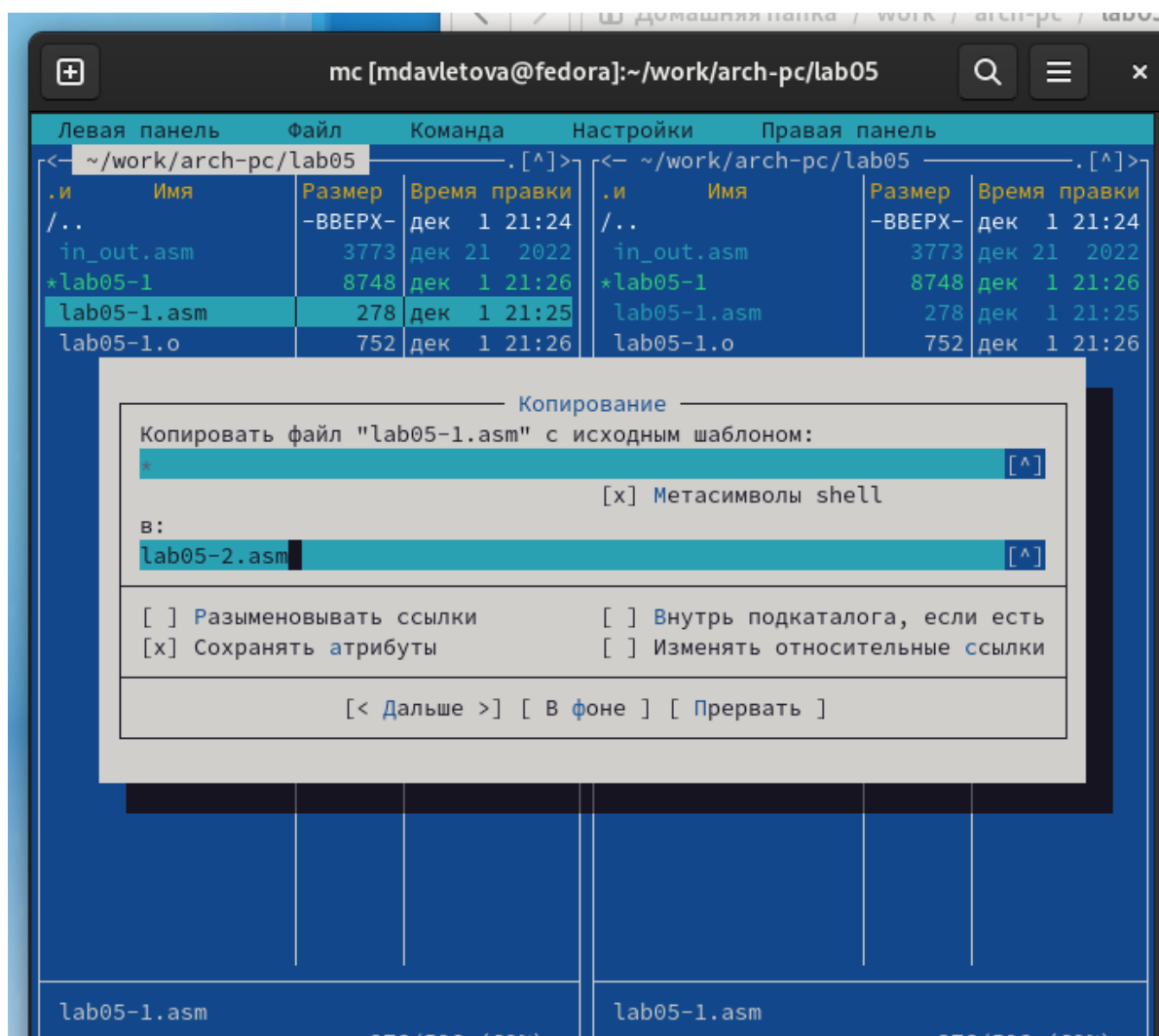
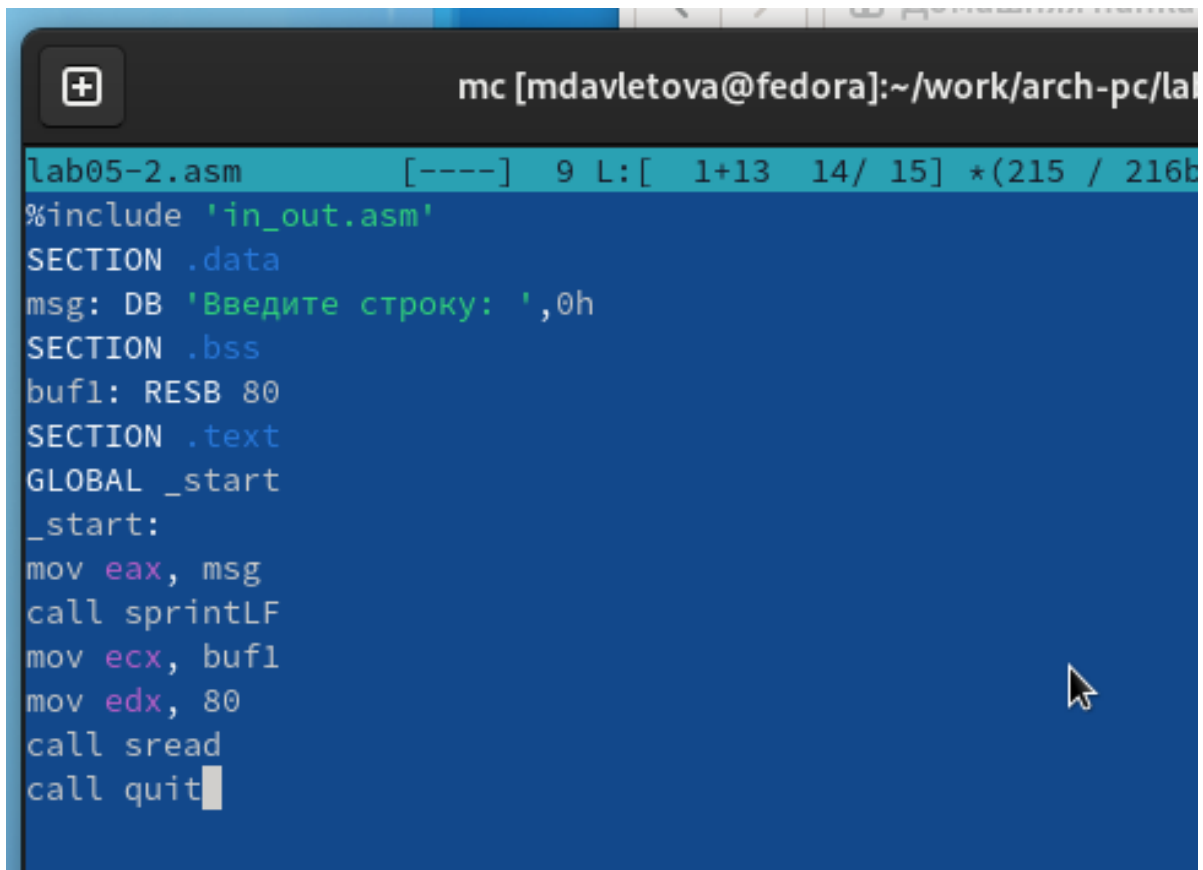


Рис. 4.8: Копирование файла lab05-1.asm

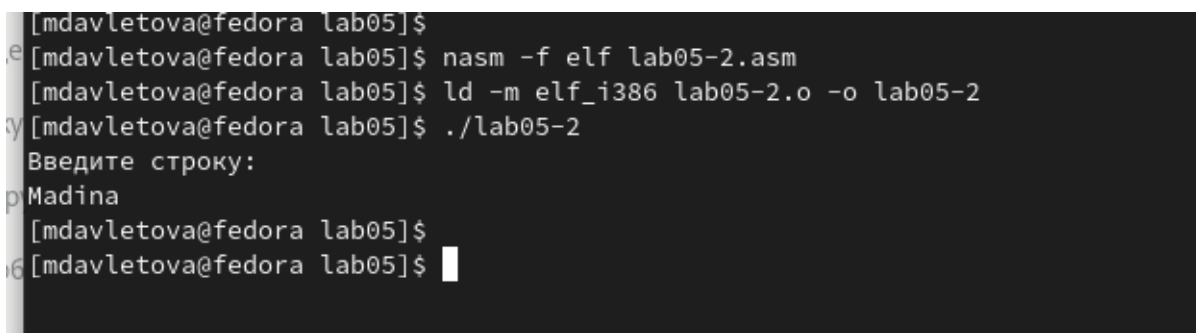
Затем я написала код программы lab05-2.asm, используя подпрограммы из внешнего файла in_out.asm (рис. [4.9]).



```
mc [mdavletova@fedora]:~/work/arch-pc/lab05-2.asm [----] 9 L:[ 1+13 14/ 15] *(215 / 216b
#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. 4.9: Изменение кода lab05-2.asm

Скомпилировала программу и проверила её запуск (рис. [4.10])

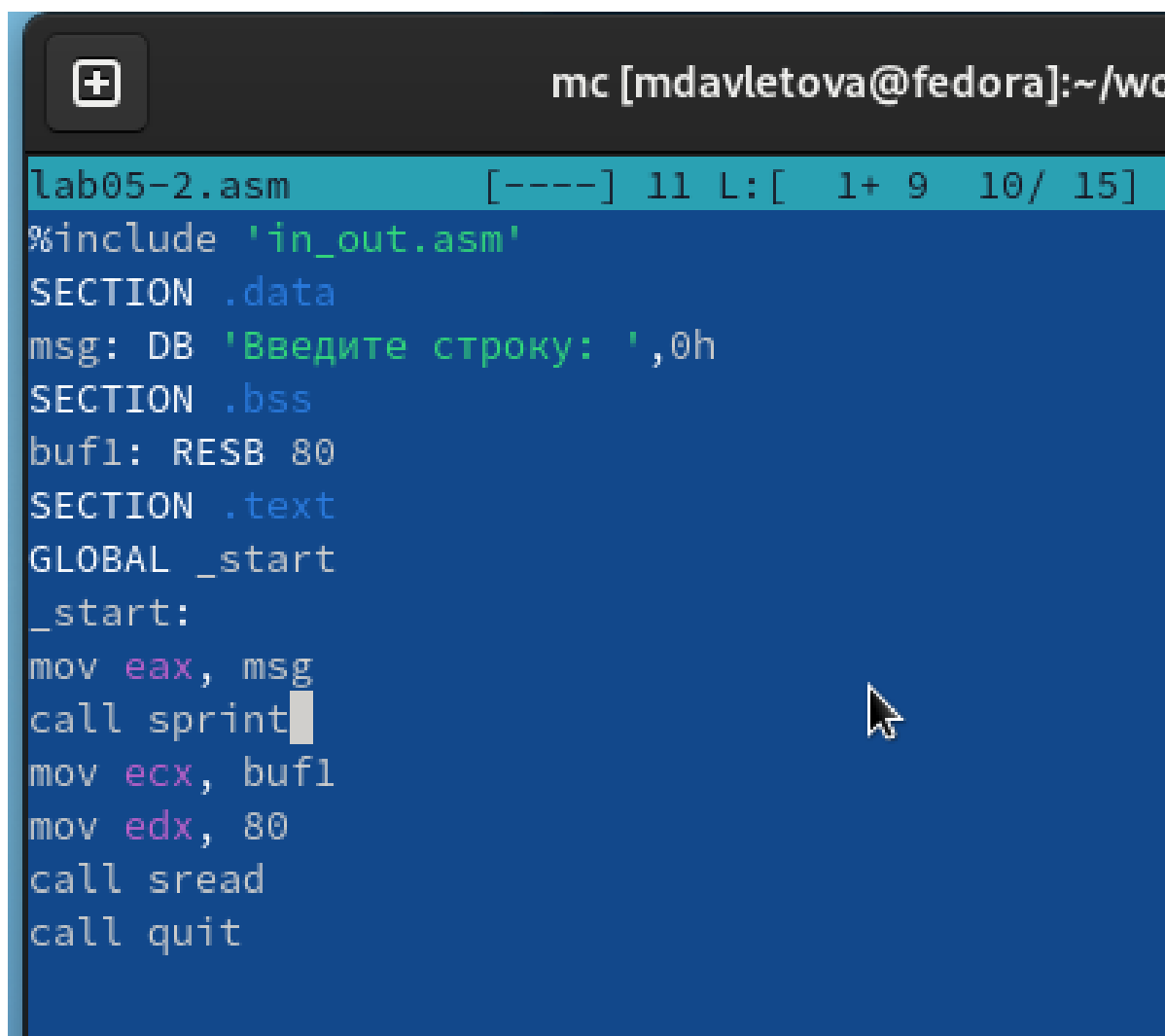


```
[mdavletova@fedora lab05]$
[mdavletova@fedora lab05]$ nasm -f elf lab05-2.asm
[mdavletova@fedora lab05]$ ld -m elf_i386 lab05-2.o -o lab05-2
[mdavletova@fedora lab05]$ ./lab05-2
Введите строку:
Madina
[mdavletova@fedora lab05]$
[mdavletova@fedora lab05]$
```

Рис. 4.10: Компиляция текста программы lab05-2.asm

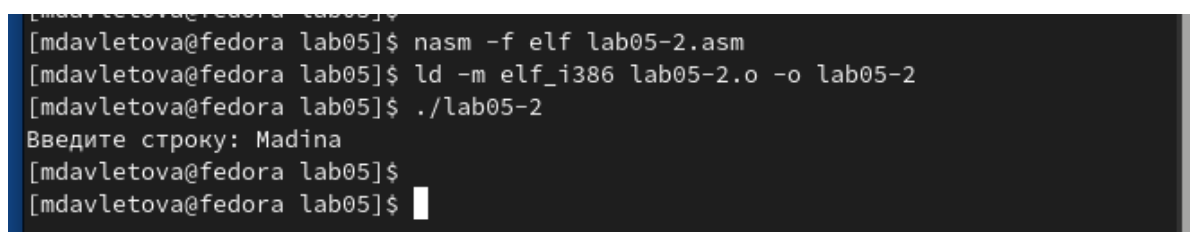
В файле lab05-2.asm заменила вызов подпрограммы sprintLF на sprint (рис. [4.11]). Я пересобрала исполняемый файл (рис. [4.12]). Теперь после вывода строки

символ перехода на новую строку отсутствует.



```
mc [mdavletova@fedora]:~/wo
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. 4.11: Изменение кода lab05-2.asm

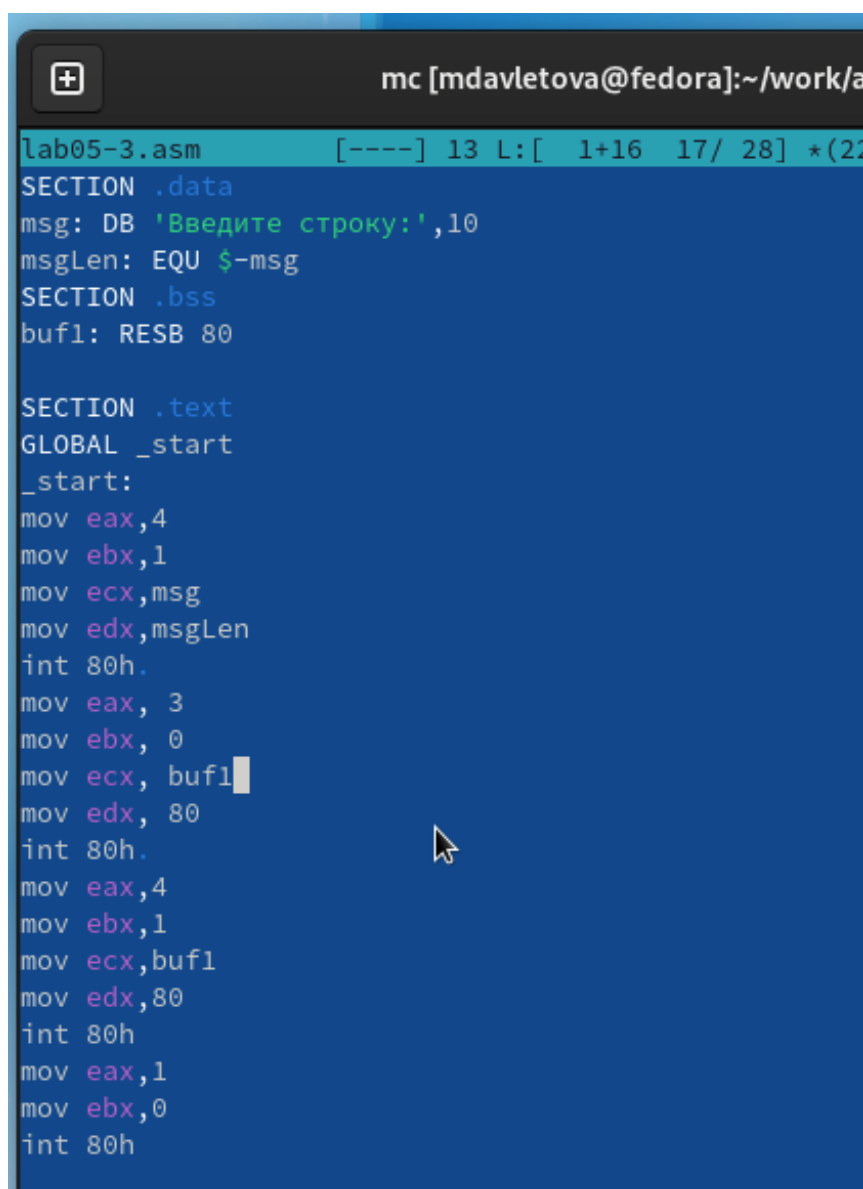


```
[mdavletova@fedora lab05]$
[mdavletova@fedora lab05]$ nasm -f elf lab05-2.asm
[mdavletova@fedora lab05]$ ld -m elf_i386 lab05-2.o -o lab05-2
[mdavletova@fedora lab05]$ ./lab05-2
Введите строку: Madina
[mdavletova@fedora lab05]$
[mdavletova@fedora lab05]$
```

Рис. 4.12: Компиляция текста программы lab05-2.asm

4.3 Выполнение заданий для самостоятельной работы

Я скопировала программу lab05-1.asm и внесла изменения в код, чтобы программа работала по следующему алгоритму: она выводит приглашение вида “Введите строку:”, считывает строку с клавиатуры и выводит введенную строку на экран. (рис. [4.13]) (рис. [4.14])



```
mc [mdavletova@fedora]:~/work/a
lab05-3.asm [-----] 13 L: [ 1+16 17/ 28] *(22
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

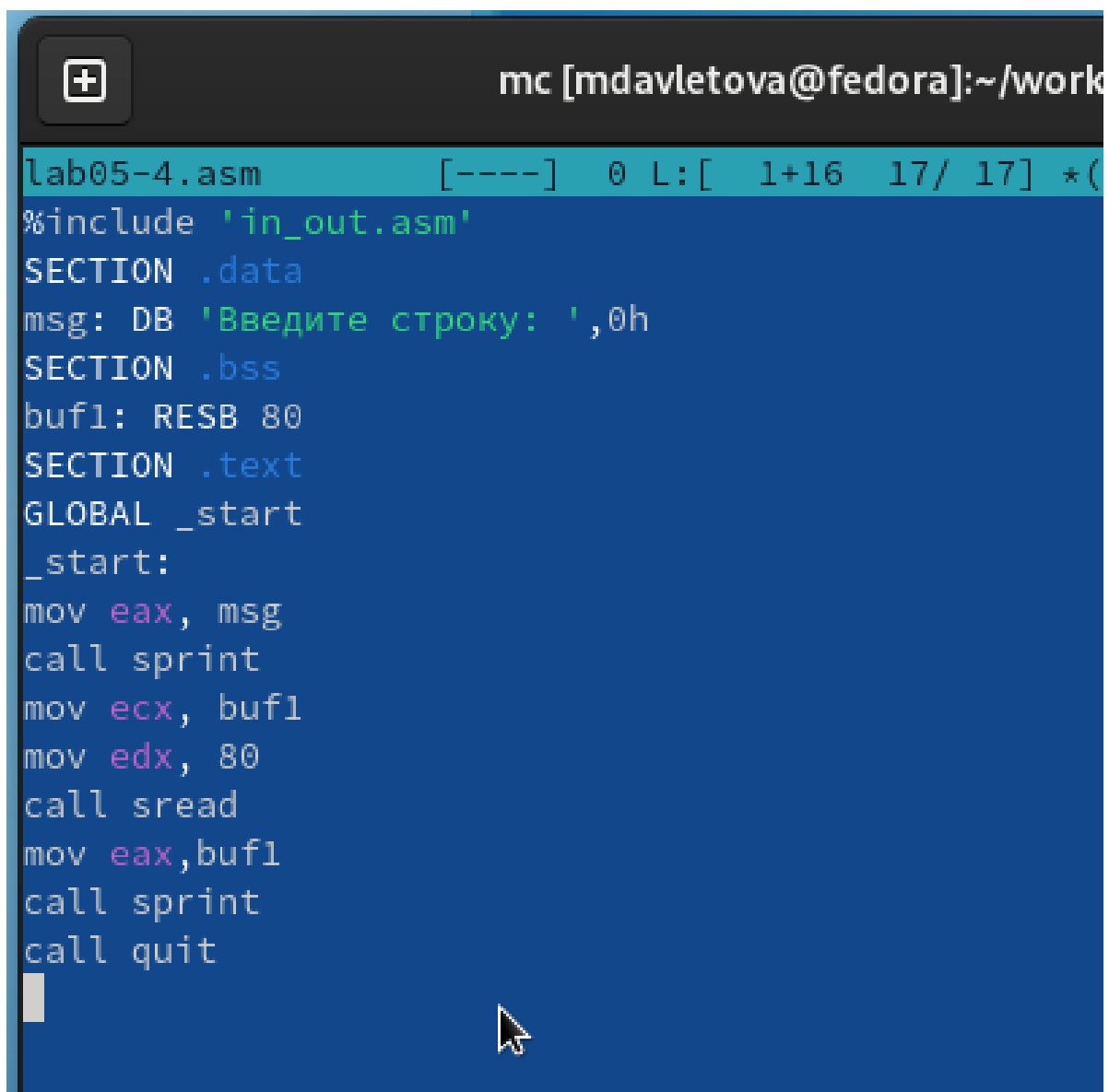
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h.
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h.
mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,80
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рис. 4.13: Изменение кода lab05-3.asm

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

Рис. 4.14: Компиляция текста программы lab05-3.asm

Аналогично я скопировала программу lab05-2.asm и изменила код, но теперь использовала подпрограммы из файла in_out.asm. (рис. [4.15]) (рис. [4.16])



```
lab05-4.asm [----] 0 L:[ 1+16 17/ 17] *(
#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. 4.15: Изменение кода lab05-4.asm

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

Рис. 4.16: Компиляция текста программы lab05-4.asm

5 Выводы

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