

# **Отчёта по лабораторной работе 6**

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

Грязнов Михаил Александрович НПИбд-01-22

# 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>14</b>

## Список иллюстраций

2.1	Создание файлов в Midnight Commander . . . . .	6
2.2	Редактирование файла 1 в Midnight Commander . . . . .	7
2.3	Проверка программы 1 . . . . .	8
2.4	Файл in_out.asm . . . . .	8
2.5	Редактирование файла 2 в Midnight Commander . . . . .	9
2.6	Проверка программы 2 . . . . .	9
2.7	Редактирование файла 3 в Midnight Commander . . . . .	10
2.8	Проверка программы 3 . . . . .	10
2.9	Редактирование файла 4 в Midnight Commander . . . . .	11
2.10	Проверка программы 4 . . . . .	12
2.11	Редактирование файла 5 в Midnight Commander . . . . .	12
2.12	Проверка программы 5 . . . . .	13

## **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. Создадим новый подкаталог с именем lab06 и в нем файл lab6-1.asm. (рис. 2.1)

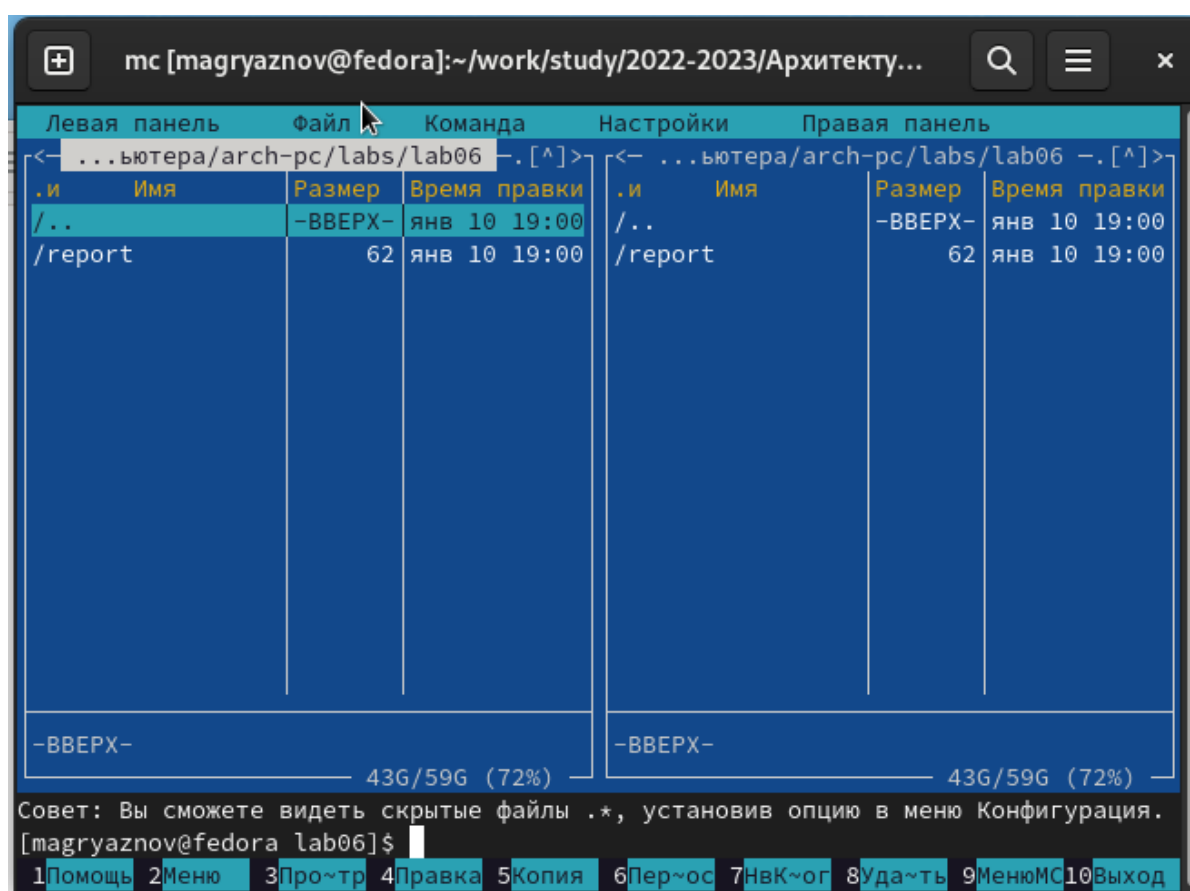
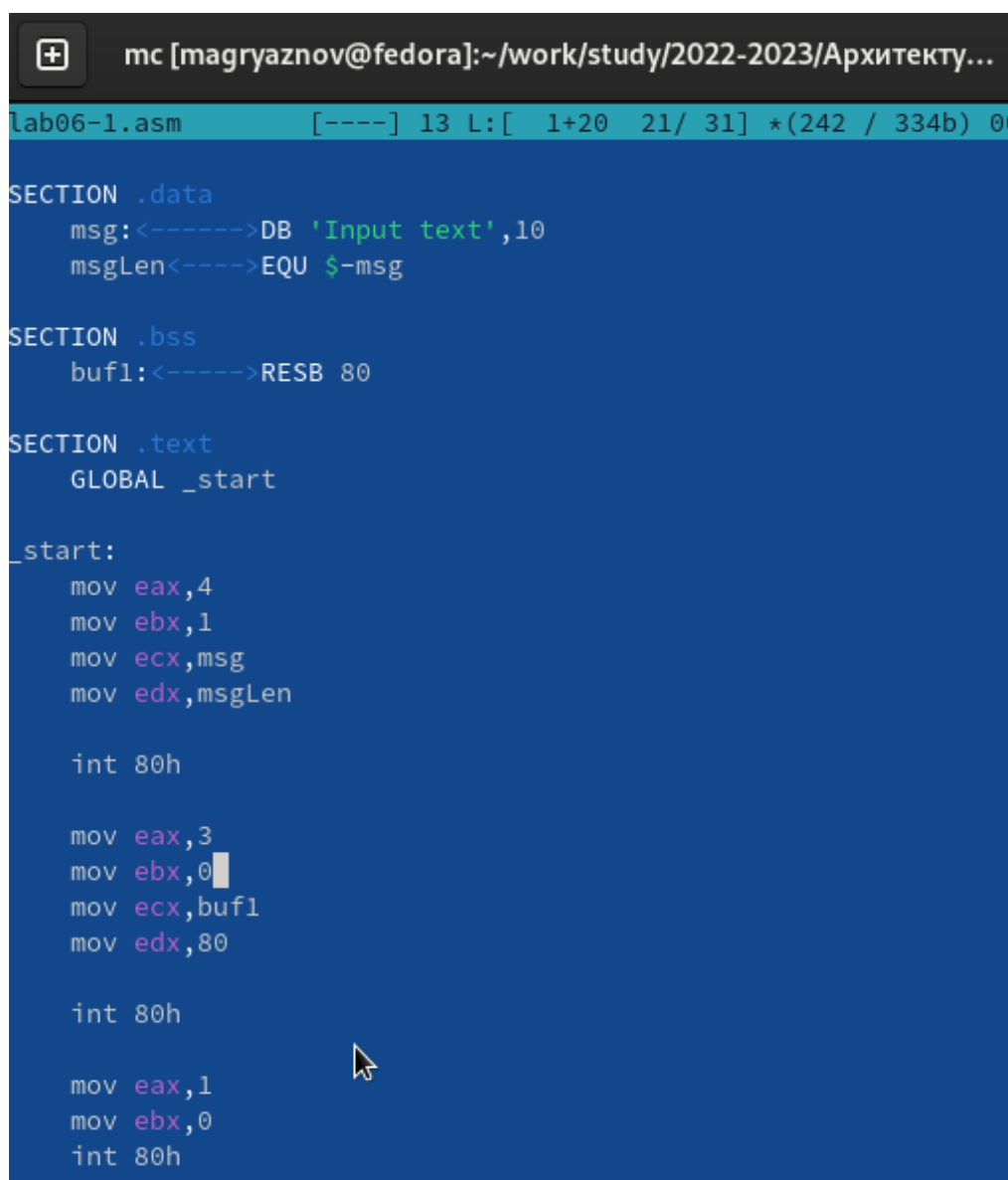


Рис. 2.1: Создание файлов в Midnight Commander

2. Введем в файл lab6-1.asm текст программы вывода сообщения на экран и ввода строки с клавиатуры (Листинг 1.). Создадим исполняемый файл и проверим его работу. (рис. 2.2, 2.3)

A screenshot of the Midnight Commander file manager. The title bar shows the user 'mc [magryaznov@fedora]' and the current directory '~/.work/study/2022-2023/Архитекту...'. The active file is 'lab06-1.asm'. The status bar at the top indicates line 13, column 1, with a cursor at position 21 of 31, and a file size of 242 bytes. The main window displays assembly code for sections .data, .bss, and .text. The .text section starts with a global \_start label, followed by several instructions: 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. A mouse cursor is visible over the code.

```
mc [magryaznov@fedora]:~/work/study/2022-2023/Архитекту...
lab06-1.asm [----] 13 L: [ 1+20 21/ 31] *(242 / 334b) 0

SECTION .data
    msg:<----->DB 'Input text',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: Редактирование файла 1 в Midnight Commander

```
magryaznov@fedora:~/work/study/2022-2023/Архитектура ко...
[magryaznov@fedora lab06]$ nasm -f elf lab06-1.asm
[magryaznov@fedora lab06]$ ld -m elf_i386 -o lab06-1 lab06-1.o
[magryaznov@fedora lab06]$ ./lab06-1
Input text
Mihail
[magryaznov@fedora lab06]$
```

Рис. 2.3: Проверка программы 1

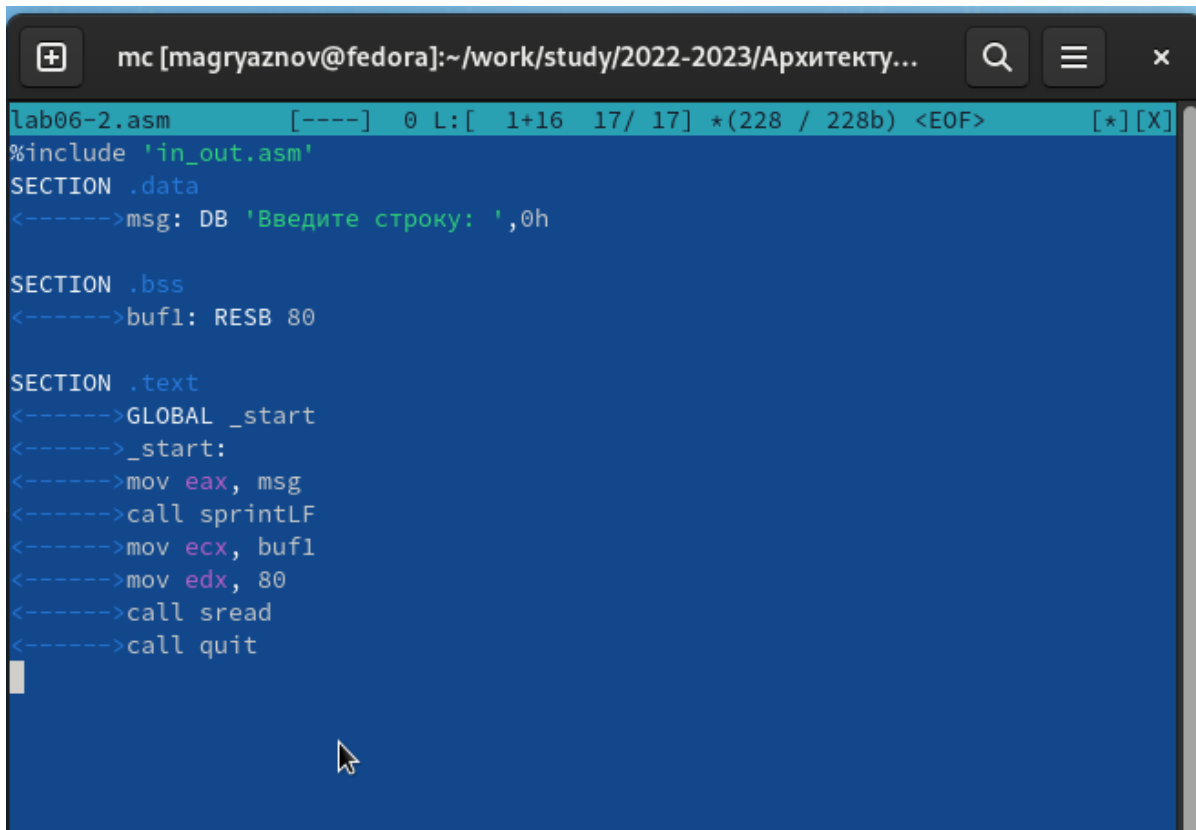
3. Скачали с туис доп файл, скопировали программу. (рис. 2.4)

Левая панель				Правая панель			
Имя	Размер	Время правки		Имя	Размер	Время правки	
../	-ВВЕРХ-	янв 10 19:00		../	-ВВЕРХ-	янв 10 19:00	
/report	62	янв 10 19:00		/report	62	янв 10 19:00	
*lab06-1	8728	янв 10 19:31		*lab06-1	8728	янв 10 19:31	
lab06-1.asm	334	янв 10 19:30		lab06-1.asm	334	янв 10 19:30	
lab06-1.o	736	янв 10 19:31		lab06-1.o	736	янв 10 19:31	
lab06-2.asm	228	янв 10 19:32		lab06-2.asm	228	янв 10 19:32	

Рис. 2.4: Файл in\_out.asm

4. Изменили код программы. (рис. 2.5, 2.6)





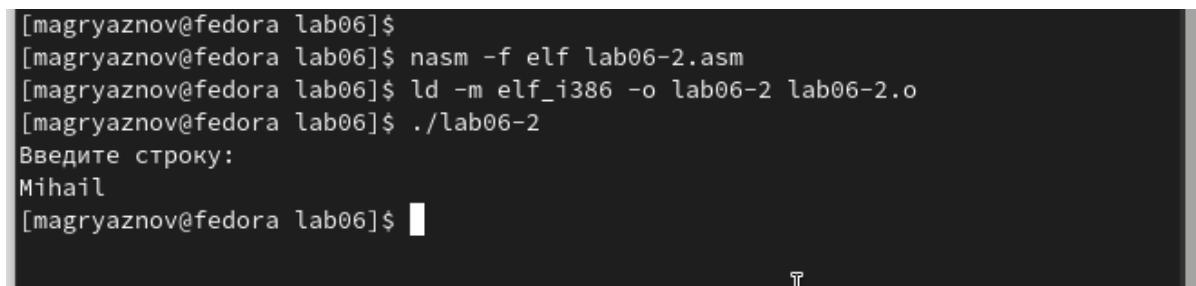
The screenshot shows the Midnight Commander file manager and editor. The title bar indicates the user is 'mc [magryaznov@fedora]:~/work/study/2022-2023/Архитекту...'. The editor window displays the file 'lab06-2.asm' with a status bar showing '0 L: [ 1+16 17/ 17] \*(228 / 228b) <EOF>'. The code is as follows:

```
lab06-2.asm
%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
```

Рис. 2.5: Редактирование файла 2 в Midnight Commander

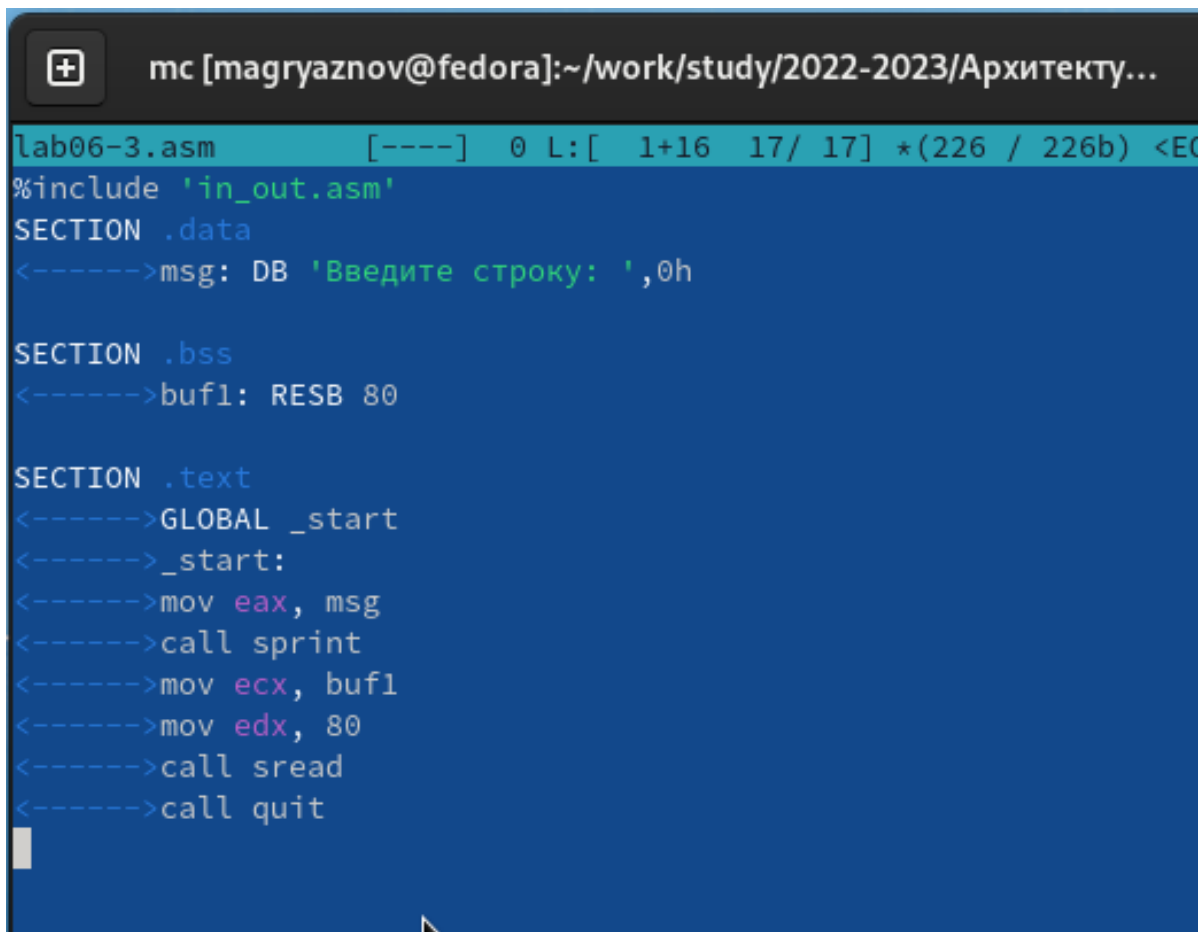


The screenshot shows a terminal window with the following commands and output:

```
[magryaznov@fedora lab06]$
[magryaznov@fedora lab06]$ nasm -f elf lab06-2.asm
[magryaznov@fedora lab06]$ ld -m elf_i386 -o lab06-2 lab06-2.o
[magryaznov@fedora lab06]$ ./lab06-2
Введите строку:
Mihail
[magryaznov@fedora lab06]$
```

Рис. 2.6: Проверка программы 2

5. Изменили вызов подпрограммы. Теперь ввод и вывод в одну строку. (рис. 2.7, 2.8)

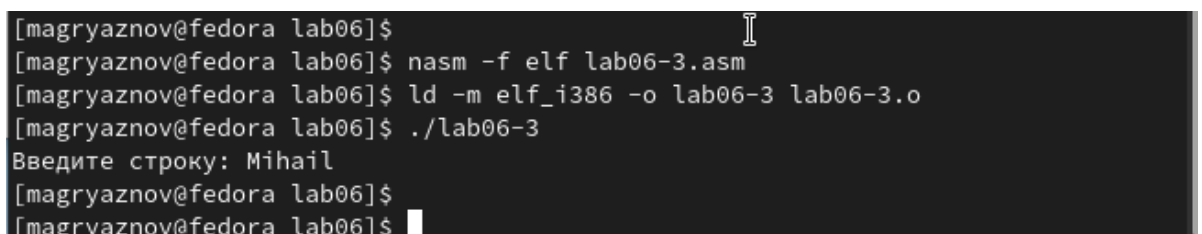


```
mc [magryaznov@fedora]:~/work/study/2022-2023/Архитекту...
lab06-3.asm  [----]  0 L:[ 1+16 17/ 17] *(226 / 226b) <E0
%include 'in_out.asm'
SECTION .data
<----->msg: DB 'Введите строку: ',0h

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. 2.7: Редактирование файла 3 в Midnight Commander



```
[magryaznov@fedora lab06]$
[magryaznov@fedora lab06]$ nasm -f elf lab06-3.asm
[magryaznov@fedora lab06]$ ld -m elf_i386 -o lab06-3 lab06-3.o
[magryaznov@fedora lab06]$ ./lab06-3
Введите строку: Mihail
[magryaznov@fedora lab06]$
[magryaznov@fedora lab06]$
```

Рис. 2.8: Проверка программы 3

6. Внесем изменения в программу (без использования внешнего файла `in_out.asm`), так чтобы она работала по следующему алгоритму: (рис. 2.9, 2.10)

- вывести приглашение типа “Введите строку:”;

- ввести строку с клавиатуры;
- вывести введенную строку на экран.



The image shows a terminal window with the Midnight Commander (MC) file editor. The title bar indicates the user is 'mc [magryaznov@fedora]:~/work/study/2022-2023/Архитекту...'. The editor is open to a file named 'lab06-4.asm'. The status bar at the top shows '[----] 0 L: [ 6+32 38/ 40] \*(404 / 406b) 0010 0'. The code is written in assembly language and is color-coded: blue for section names, green for global symbols, and purple for register names. The code defines a buffer 'buf1' in the .bss section and implements a loop in the .text section that reads a character from the keyboard (int 80h), stores it in the buffer, and prints it to the screen (mov eax, 3; mov ebx, 0; mov ecx, buf1; mov edx, 80; int 80h) until the user presses the Enter key (mov eax, 1; mov ebx, 0; int 80h).

```
lab06-4.asm [----] 0 L: [ 6+32 38/ 40] *(404 / 406b) 0010 0
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. 2.9: Редактирование файла 4 в Midnight Commander

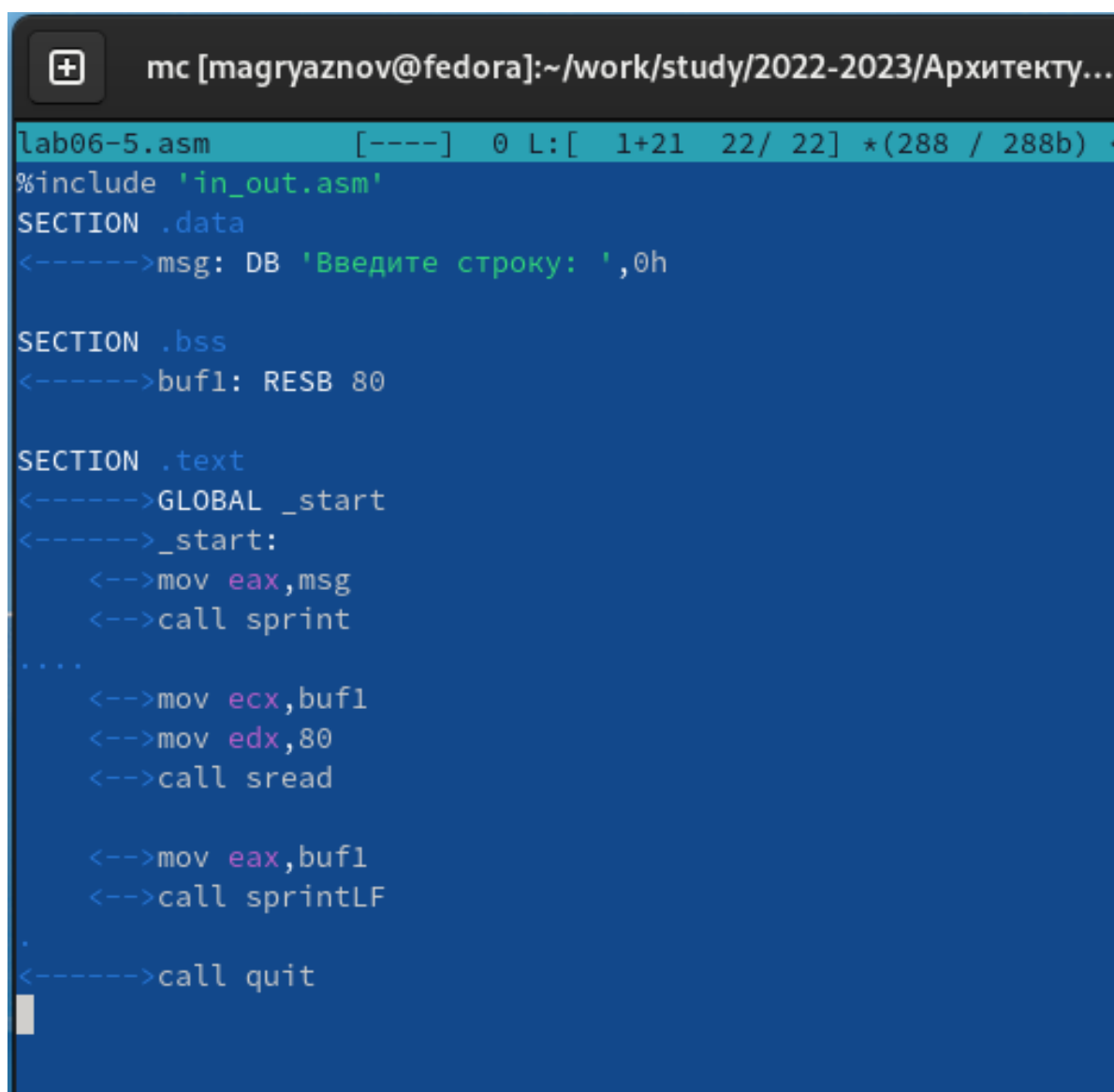
```

[magryaznov@fedora lab06]$
[magryaznov@fedora lab06]$ nasm -f elf lab06-4.asm
[magryaznov@fedora lab06]$ ld -m elf_i386 -o lab06-4 lab06-4.o
[magryaznov@fedora lab06]$ ./lab06-4
Input text
Mihail
Mihail
[magryaznov@fedora lab06]$

```

Рис. 2.10: Проверка программы 4

7. Сделаем тоже самое с файлом in\_out.asm (рис. 2.11, 2.12)



```

lab06-5.asm  [----]  0 L: [ 1+21  22/ 22] *(288 / 288b) <
#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 sprintfLF
    .
<----->call quit

```

Рис. 2.11: Редактирование файла 5 в Midnight Commander

```
[magryaznov@fedora lab06]$  
[magryaznov@fedora lab06]$ nasm -f elf lab06-5.asm  
[magryaznov@fedora lab06]$ ld -m elf_i386 -o lab06-5 lab06-5.o  
[magryaznov@fedora lab06]$ ./lab06-5  
Введите строку:  
  
[magryaznov@fedora lab06]$ ./lab06-5  
Введите строку: Mihail  
Mihail
```

Рис. 2.12: Проверка программы 5

## 3 Выводы

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