

# **Отчёта по лабораторной работе 6**

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

Туем Гислен НКАбд-03-22

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>15</b>
	<b>Список литературы</b>	<b>16</b>

## Список иллюстраций

4.1	Создание файлов в Midnight Commander . . . . .	8
4.2	Редактирование файла с программой в Midnight Commander . . .	9
4.3	Проверка программы . . . . .	10
4.4	Файл in_out.asm . . . . .	10
4.5	Редактирование файла с программой в Midnight Commander . . .	11
4.6	Проверка программы . . . . .	11
4.7	Редактирование файла с программой в Midnight Commander . . .	12
4.8	Проверка программы . . . . .	12
4.9	Редактирование файла с программой в Midnight Commander . . .	13
4.10	Проверка программы . . . . .	13
4.11	Редактирование файла с программой в Midnight Commander . . .	14
4.12	Проверка программы . . . . .	14

## **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. Изучите как работать в Midnight Commander.
2. Изучите примеры программ из задания к работе.
3. Дополните примеры в соответствии с заданием.
4. Загрузите файлы на GitHub.

### 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

1. Создадим новый подкаталог с именем lab06 и в нем файл lab6-1.asm. (рис. 4.1)

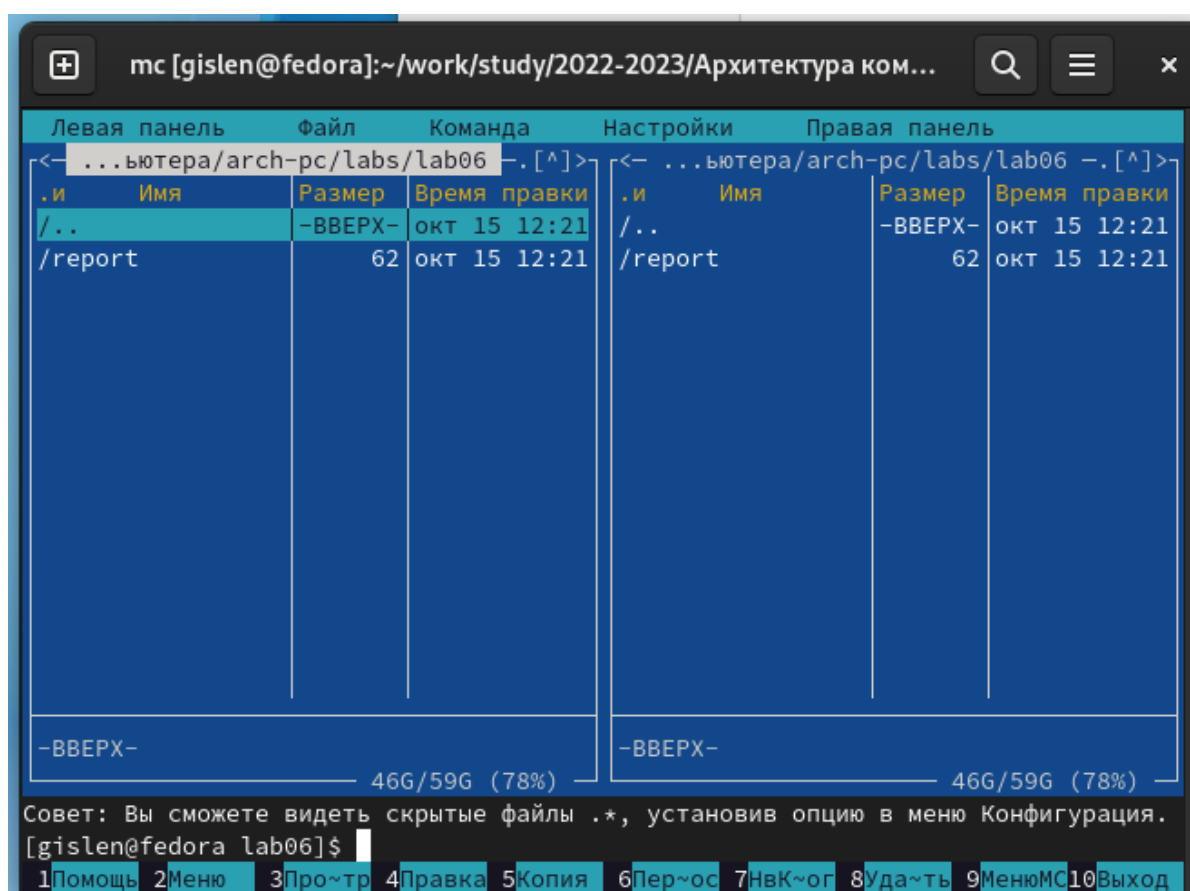
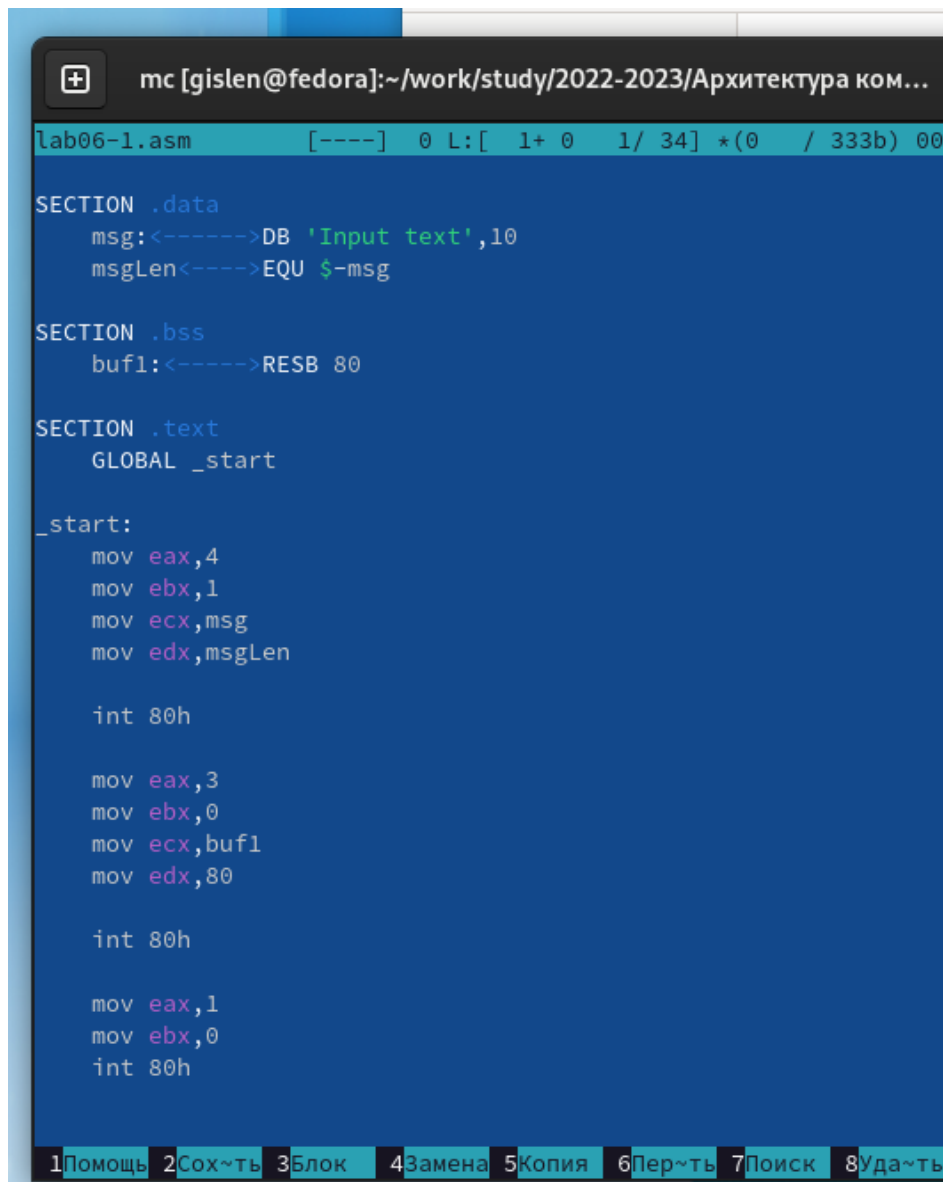


Рис. 4.1: Создание файлов в Midnight Commander

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. 4.2, 4.3)





The image shows a screenshot of the Midnight Commander (mc) file manager and editor. The title bar indicates the user is gislen@fedora and the current directory is ~/work/study/2022-2023/Архитектура ком... The active file is lab06-1.asm. The editor displays assembly code for three sections: .data, .bss, and .text. The .data section defines a message 'Input text' and its length. The .bss section reserves 80 bytes for a buffer. The .text section contains assembly instructions for a program that moves values into registers, prints a message, and then prints the buffer. At the bottom, a menu bar provides options in Russian: 1Помощь, 2Сох~ть, 3Блок, 4Замена, 5Копия, 6Пер~ть, 7Поиск, 8Уда~ть.

```
mc [gislen@fedora]:~/work/study/2022-2023/Архитектура ком...
lab06-1.asm  [----]  0 L:[ 1+ 0 1/ 34] *(0 / 333b) 00

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

1Помощь 2Сох~ть 3Блок 4Замена 5Копия 6Пер~ть 7Поиск 8Уда~ть
```

Рис. 4.2: Редактирование файла с программой в Midnight Commander

```

gislen@fedora:~/work/study/2022-2023/Архитектура компью...
[gislen@fedora lab06]$ nasm -f elf lab06-1.asm
[gislen@fedora lab06]$ ld -m elf_i386 -o lab06-1 lab06-1.o
[gislen@fedora lab06]$ ./lab06-1
Input text
Gislen
[gislen@fedora lab06]$

```

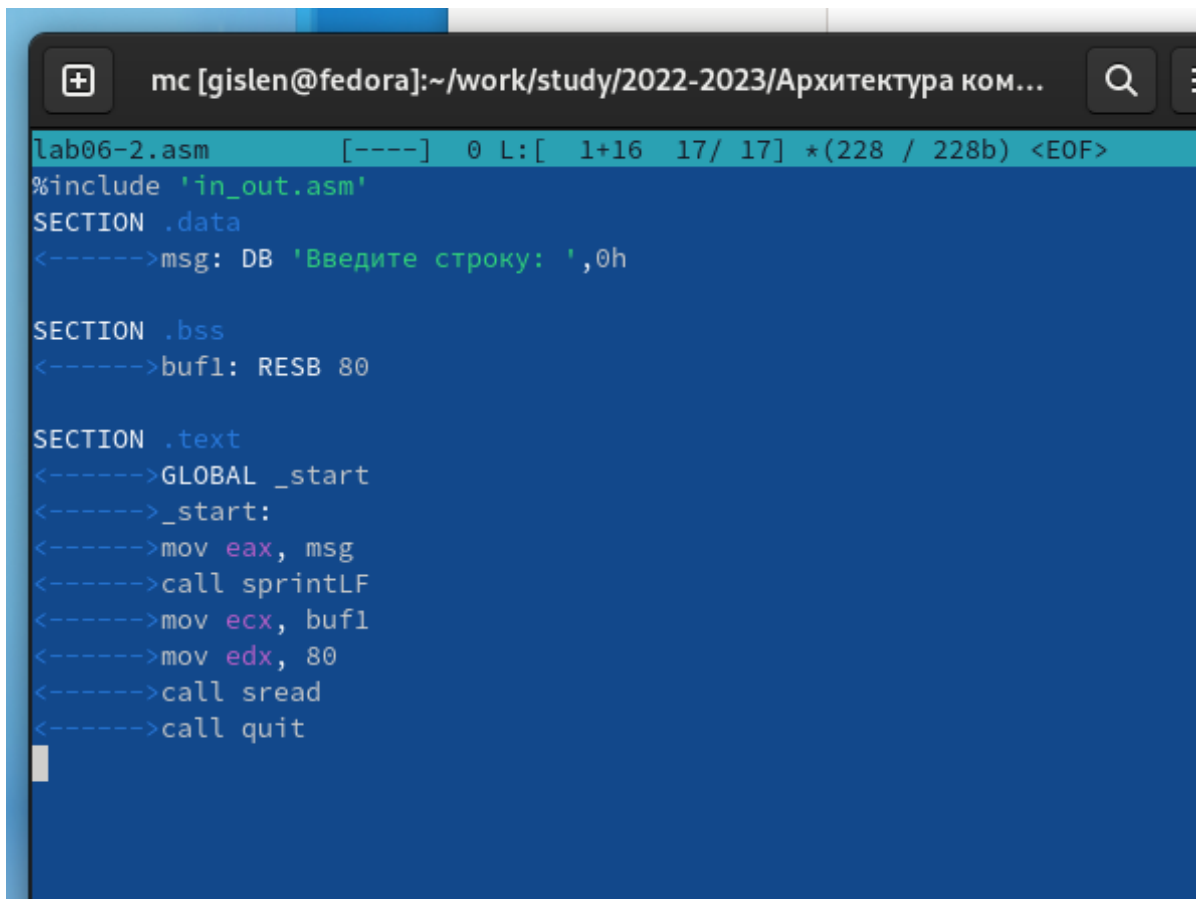
Рис. 4.3: Проверка программы

3. Скачали с туис доп файл, скопировали программу. (рис. 4.4)

Левая панель			Правая панель		
Имя	Размер	Время правки	Имя	Размер	Время правки
../	-ВВЕРХ-	окт 15 12:21	../	-ВВЕРХ-	окт 15 12:21
/report	62	окт 15 12:21	/report	62	окт 15 12:21
in_out.asm	3773	ноя 6 15:38	in_out.asm	3773	ноя 6 15:38
*lab06-1	8728	ноя 16 15:40	*lab06-1	8728	ноя 16 15:40
lab06-1.asm	333	ноя 6 15:38	lab06-1.asm	333	ноя 6 15:38
lab06-1.o	736	ноя 16 15:40	lab06-1.o	736	ноя 16 15:40
lab06-2.asm	226	ноя 16 15:16	lab06-2.asm	226	ноя 16 15:16

Рис. 4.4: Файл in\_out.asm

4. Изменили код программы. (рис. 4.5, 4.6)

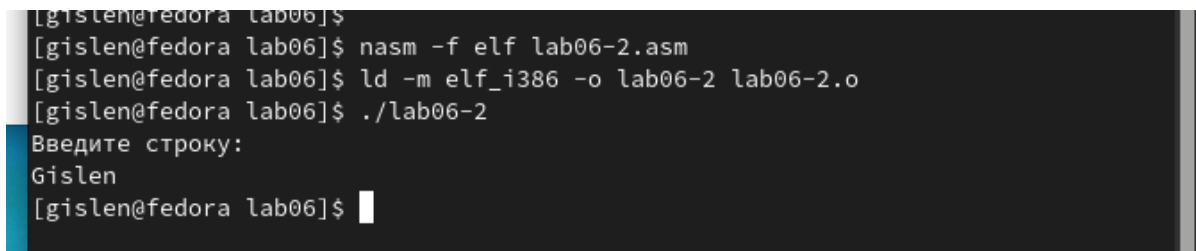
A screenshot of the Midnight Commander file manager and editor. The title bar shows the user 'gislen@fedora' and the file path '~/work/study/2022-2023/Архитектура ком...'. The editor window displays assembly code for 'lab06-2.asm'. The code includes a header line with file statistics, followed by an include directive for 'in\_out.asm'. It defines a data section with a message string 'Введите строку: ', a bss section with a buffer 'buf1' of size 80, and a text section containing assembly instructions to print the message and read input into the buffer. The code ends with a 'quit' instruction.

```
lab06-2.asm      [----]  0 L:[ 1+16 17/ 17] *(228 / 228b) <EOF>
#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
```

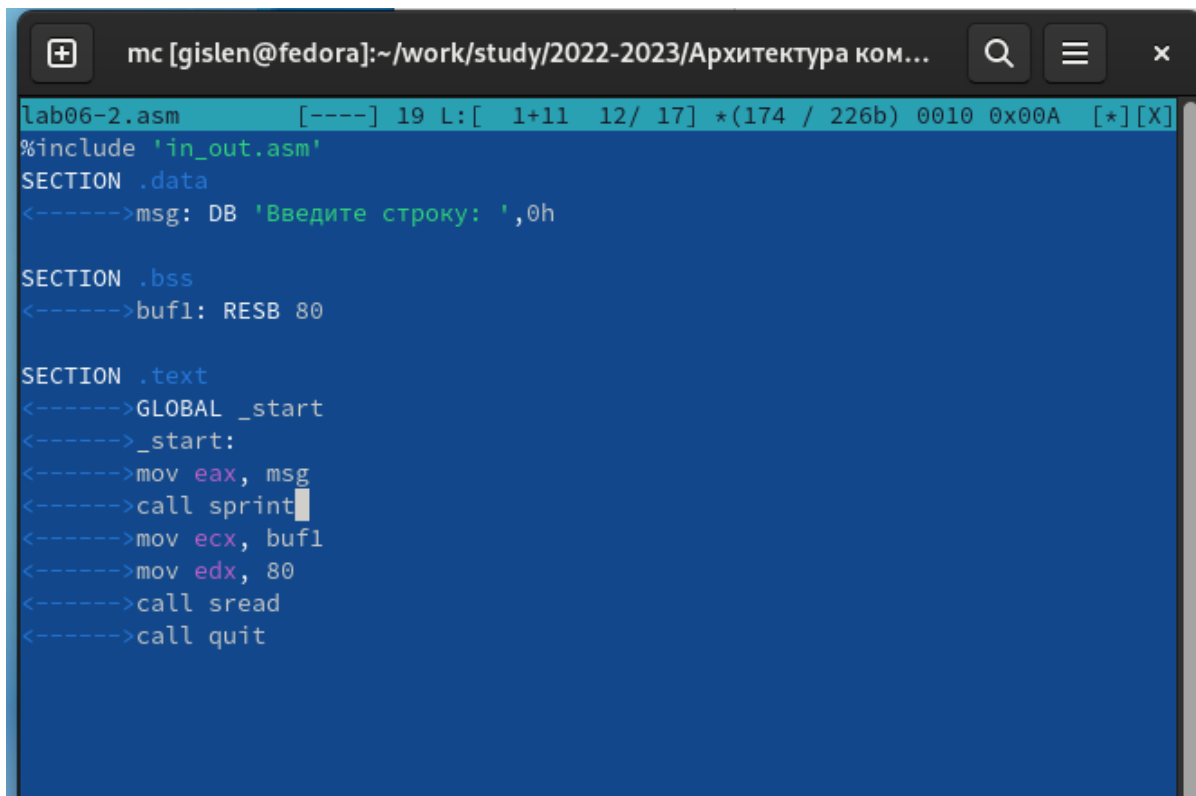
Рис. 4.5: Редактирование файла с программой в Midnight Commander

A screenshot of a terminal window showing the compilation and execution of the assembly program. The user runs 'nasm' to compile 'lab06-2.asm' into 'lab06-2.o', then 'ld' to link it into an executable 'lab06-2', and finally runs the executable. The program prompts for input, and the user enters 'Gislen'.

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

Рис. 4.6: Проверка программы

5. Изменили вызов подпрограммы. Теперь ввод и вывод в одну строку. (рис. 4.7, 4.8)



```
lab06-2.asm [----] 19 L: [ 1+11 12/ 17] *(174 / 226b) 0010 0x00A [*] [X]
#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. 4.7: Редактирование файла с программой в Midnight Commander

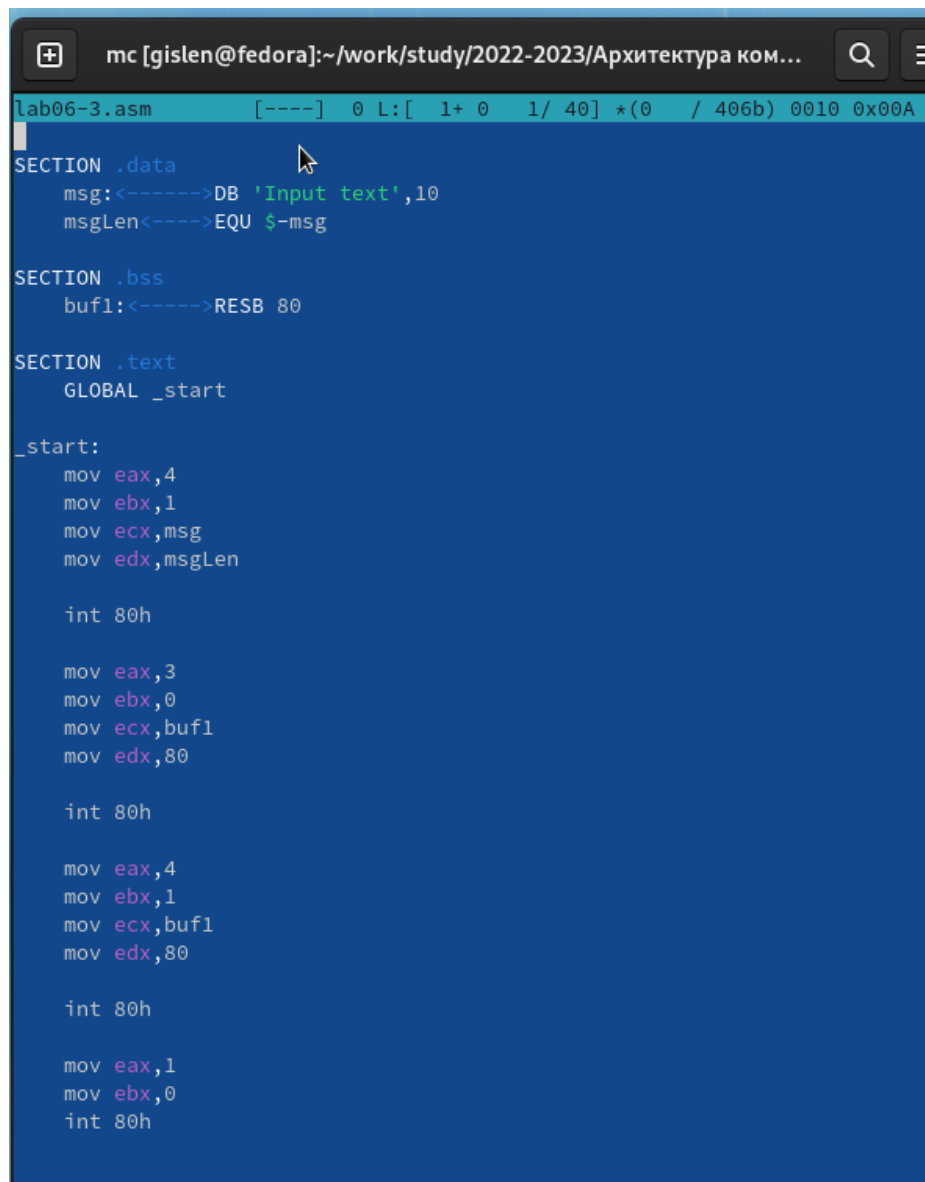


```
[gislen@fedora lab06]$
[gislen@fedora lab06]$
[gislen@fedora lab06]$ nasm -f elf lab06-2.asm
[gislen@fedora lab06]$ ld -m elf_i386 -o lab06-2 lab06-2.o
[gislen@fedora lab06]$ ./lab06-2
Введите строку: Gislen
[gislen@fedora lab06]$
[gislen@fedora lab06]$
```

Рис. 4.8: Проверка программы

6. Внесем изменения в программу (без использования внешнего файла `in_out.asm`), так чтобы она работала по следующему алгоритму: (рис. 4.9, 4.10)

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



The image shows a screenshot of the Midnight Commander (MC) file manager and editor. The title bar indicates the user is 'gislen@fedora' and the current directory is '~/work/study/2022-2023/Архитектура ком...'. The editor is open to a file named 'lab06-3.asm'. The code is written in assembly language and is color-coded. It defines three sections: .data, .bss, and .text. The .data section contains a string 'Input text' and its length. The .bss section reserves 80 bytes for a buffer. The .text section contains the main program logic, starting with a global \_start symbol. The program sets up registers (eax, ebx, ecx, edx) and performs several interrupt calls (int 80h) to display the string and its length.

```
lab06-3.asm [----] 0 L: [ 1+ 0 1/ 40] *(0 / 406b) 0010 0x00A

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

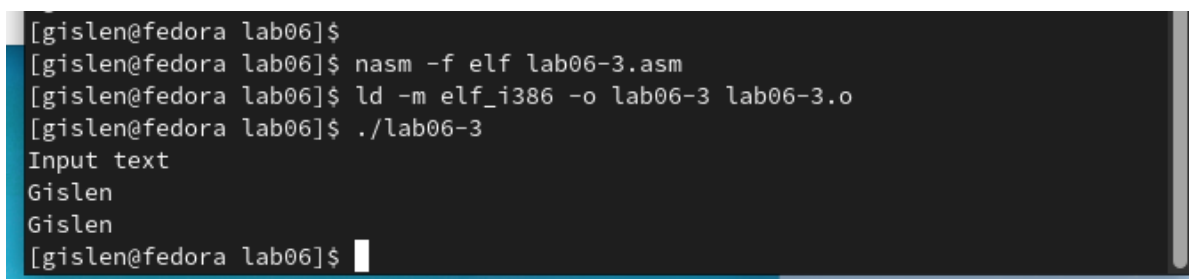
    int 80h

    mov eax,4
    mov ebx,1
    mov ecx,buf1
    mov edx,80

    int 80h

    mov eax,1
    mov ebx,0
    int 80h
```

Рис. 4.9: Редактирование файла с программой в Midnight Commander

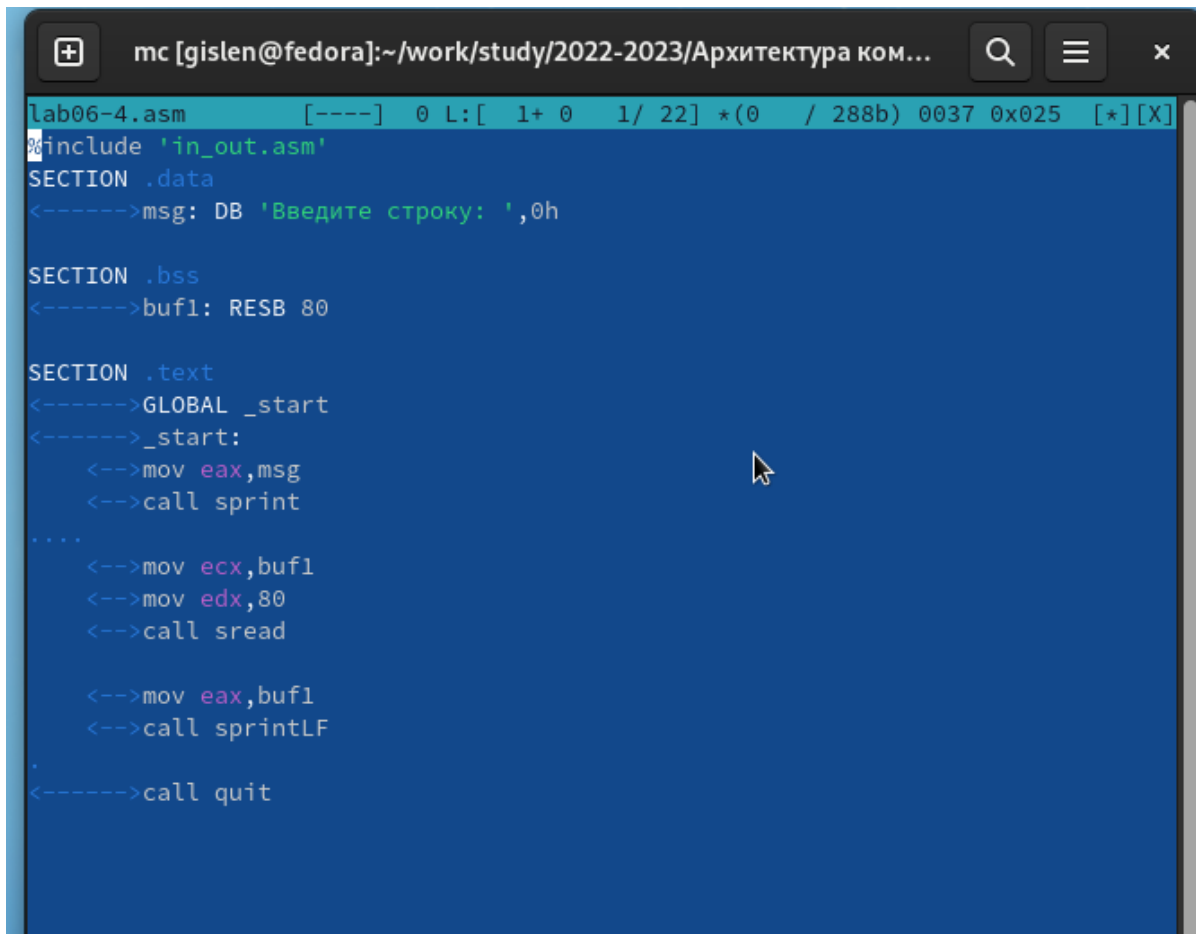


The image shows a terminal window with the following commands and output:

```
[gislen@fedora lab06]$
[gislen@fedora lab06]$ nasm -f elf lab06-3.asm
[gislen@fedora lab06]$ ld -m elf_i386 -o lab06-3 lab06-3.o
[gislen@fedora lab06]$ ./lab06-3
Input text
Gislen
Gislen
[gislen@fedora lab06]$
```

Рис. 4.10: Проверка программы

7. Сделаем тоже самое с файлом in\_out.asm (рис. 4.11, 4.12)



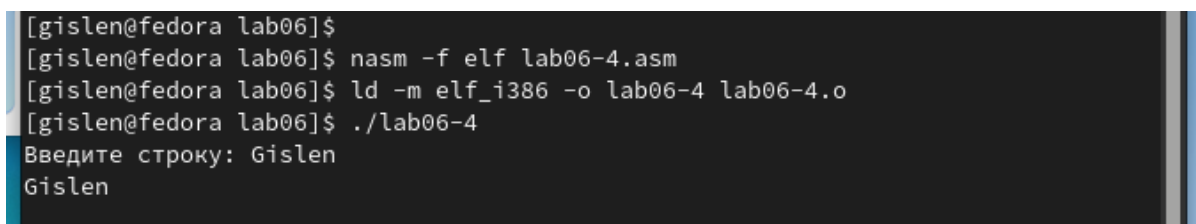
```
lab06-4.asm [----] 0 L: [ 1+ 0 1/ 22] *(0 / 288b) 0037 0x025 [*] [X]
%include 'in_out.asm'
SECTION .data
<----->msg: DB 'Введите строку: ',0h

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 sprintLF
    .
<----->call quit
```

Рис. 4.11: Редактирование файла с программой в Midnight Commander



```
[gislen@fedora lab06]$
[gislen@fedora lab06]$ nasm -f elf lab06-4.asm
[gislen@fedora lab06]$ ld -m elf_i386 -o lab06-4 lab06-4.o
[gislen@fedora lab06]$ ./lab06-4
Введите строку: Gislen
Gislen
```

Рис. 4.12: Проверка программы

## 5 Выводы

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

# Список литературы

1. Расширенный ассемблер: NASM
2. MASM, TASM, FASM, NASM под Windows и Linux