

# **Отчёта по лабораторной работе 5**

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

Жуковский Илья Сергеевич НБИбд-01-22

# Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	17

## Список иллюстраций

3.1	Создание файлов в Midnight Commander . . . . .	7
3.2	Редактирование файла 1 в Midnight Commander . . . . .	8
3.3	Проверка программы 1 . . . . .	9
3.4	Файл in_out.asm . . . . .	10
3.5	Редактирование файла 2 в Midnight Commander . . . . .	11
3.6	Проверка программы 2 . . . . .	11
3.7	Редактирование файла 3 в Midnight Commander . . . . .	12
3.8	Проверка программы 3 . . . . .	12
3.9	Редактирование файла 4 в Midnight Commander . . . . .	13
3.10	Проверка программы 4 . . . . .	14
3.11	Редактирование файла 5 в Midnight Commander . . . . .	15
3.12	Проверка программы 5 . . . . .	16

## 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 Выполнение лабораторной работы

1. Создадим новый подкаталог с именем lab05 и в нем файл lab5-1.asm. (рис. 3.1)

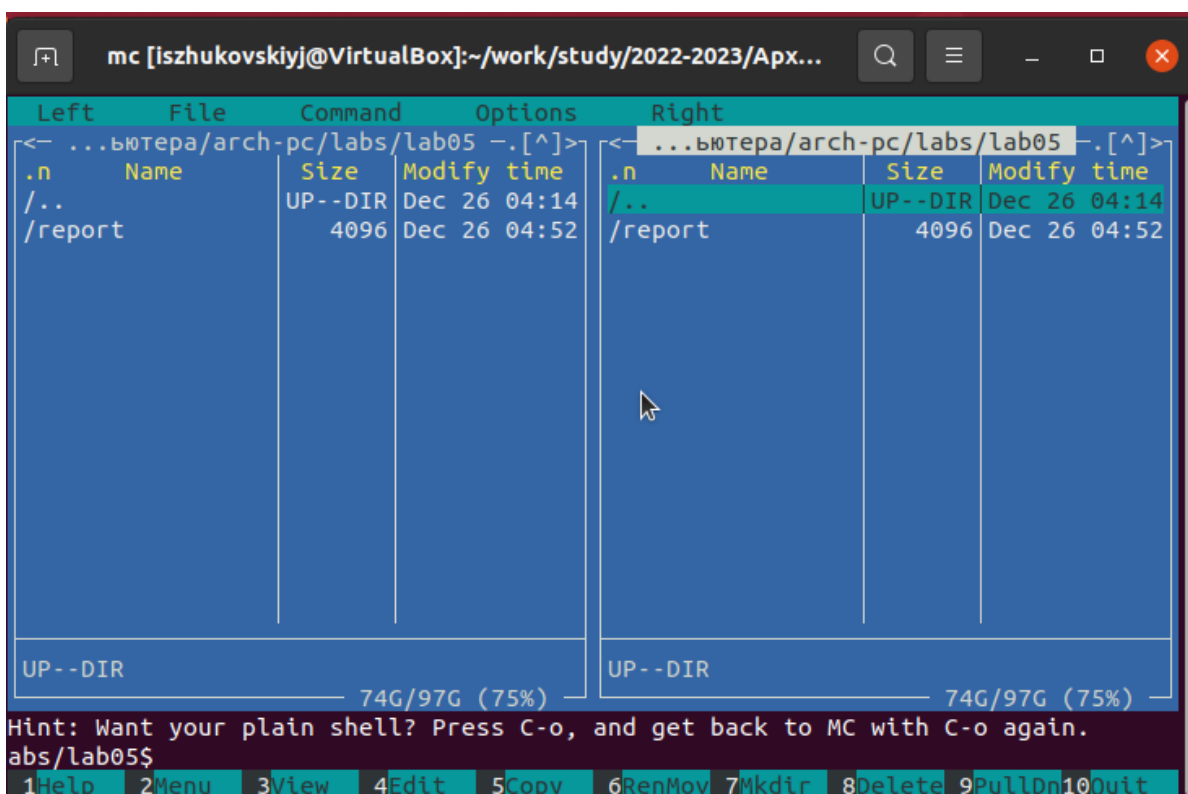
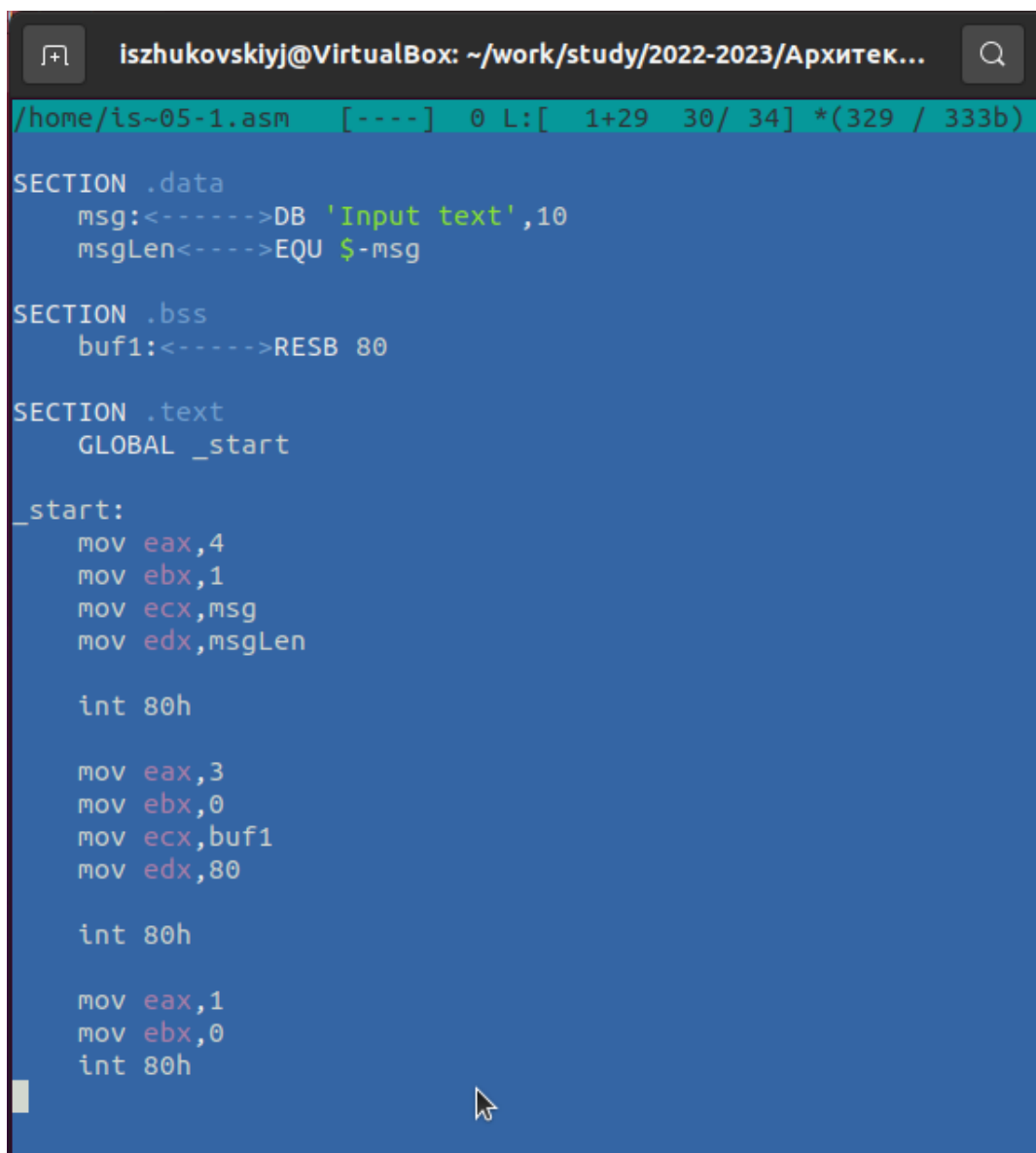


Рис. 3.1: Создание файлов в Midnight Commander

2. Введем в файл lab5-1.asm текст программы вывода сообщения на экран и ввода строки с клавиатуры (Листинг 1.). Создадим исполняемый файл и проверим его работу. (рис. 3.2, 3.3)



The image shows a screenshot of the Midnight Commander file manager and editor. The title bar at the top indicates the user is 'iszhukovskiyj@VirtualBox' and the current directory is '~/work/study/2022-2023/Архитек...'. The status bar at the top shows the file path '/home/is~05-1.asm', its size '[ --- ] 0 L:', and line information '[ 1+29 30/ 34 ] \*(329 / 333b)'. The main editing area contains assembly code for three sections: .data, .bss, and .text. The .data section defines 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 label, followed by register setup, two system calls (int 80h) for reading and writing, and a final exit call (int 80h).

```
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

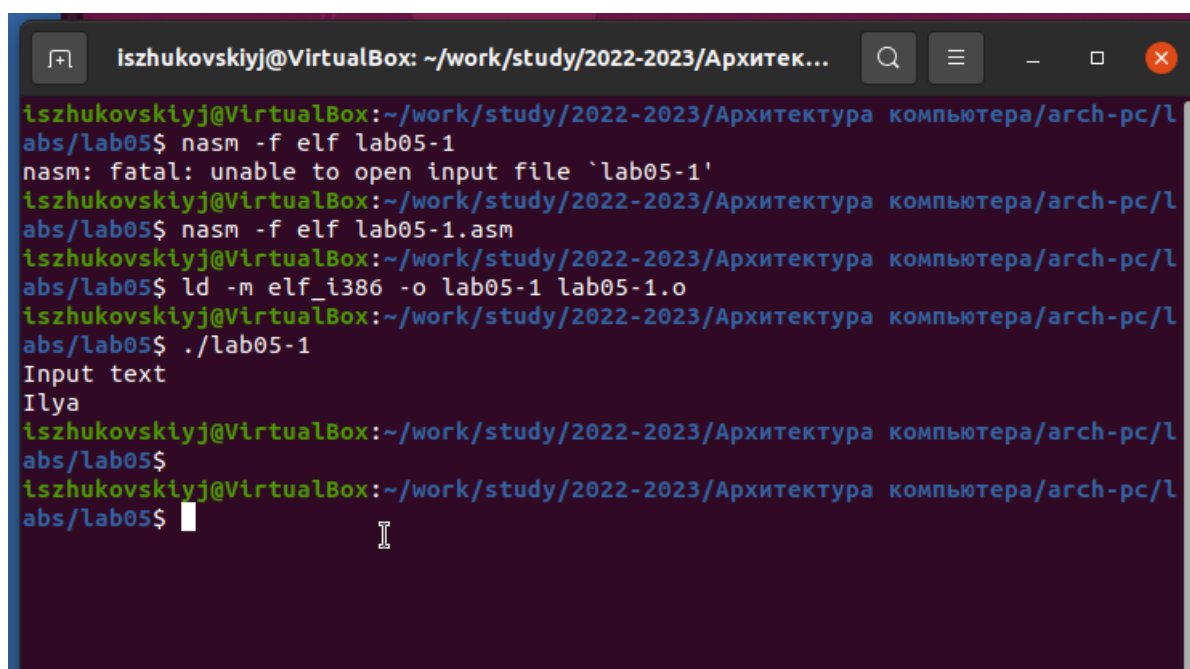
    mov eax,3
    mov ebx,0
    mov ecx,buf1
    mov edx,80

    int 80h

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

Рис. 3.2: Редактирование файла 1 в Midnight Commander





```
iszhukovskiyj@VirtualBox: ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05$ nasm -f elf lab05-1
nasm: fatal: unable to open input file `lab05-1'
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05$ nasm -f elf lab05-1.asm
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05$ ld -m elf_i386 -o lab05-1 lab05-1.o
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05$ ./lab05-1
Input text
Ilya
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05$
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05$
```

Рис. 3.3: Проверка программы 1

3. Скачали с туис доп файл, скопировали программу. (рис. 3.4)

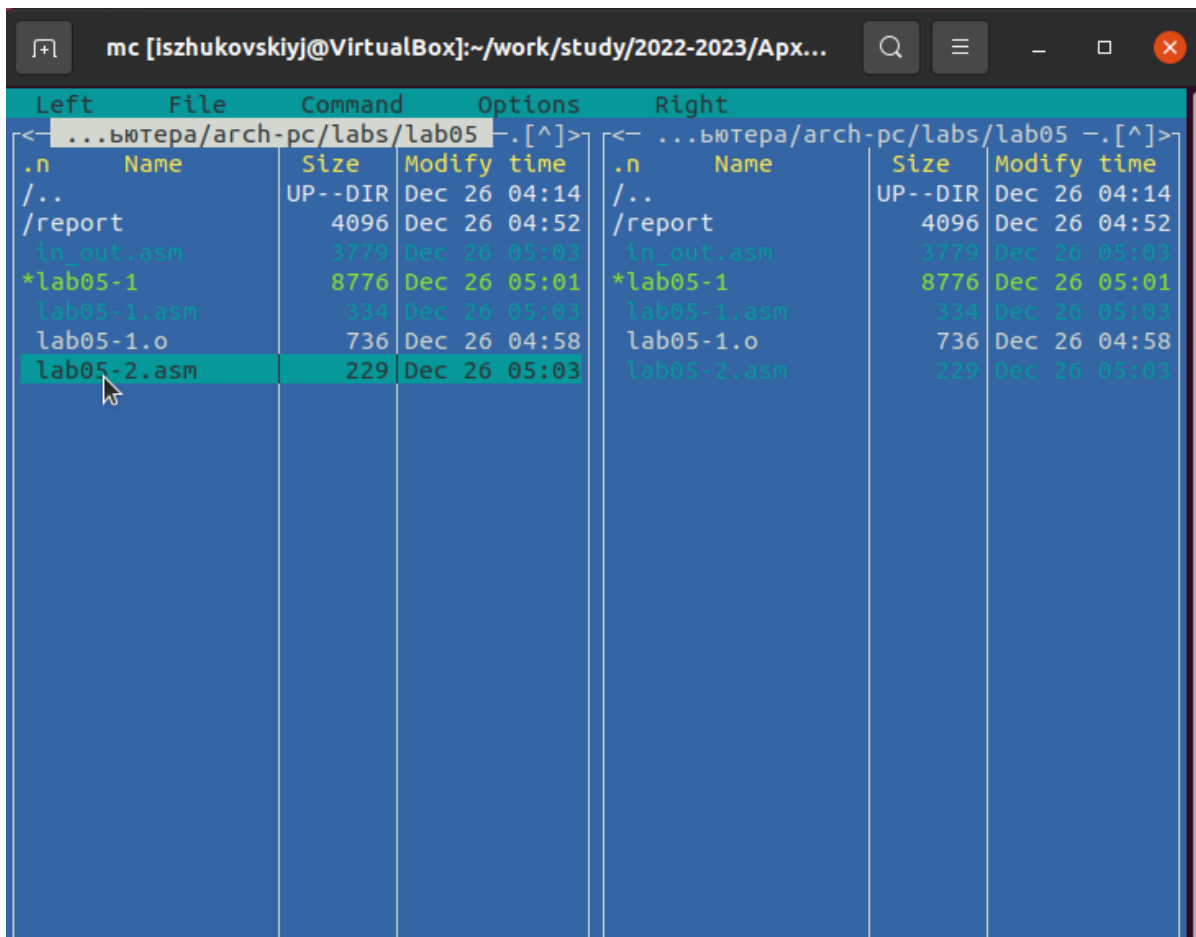
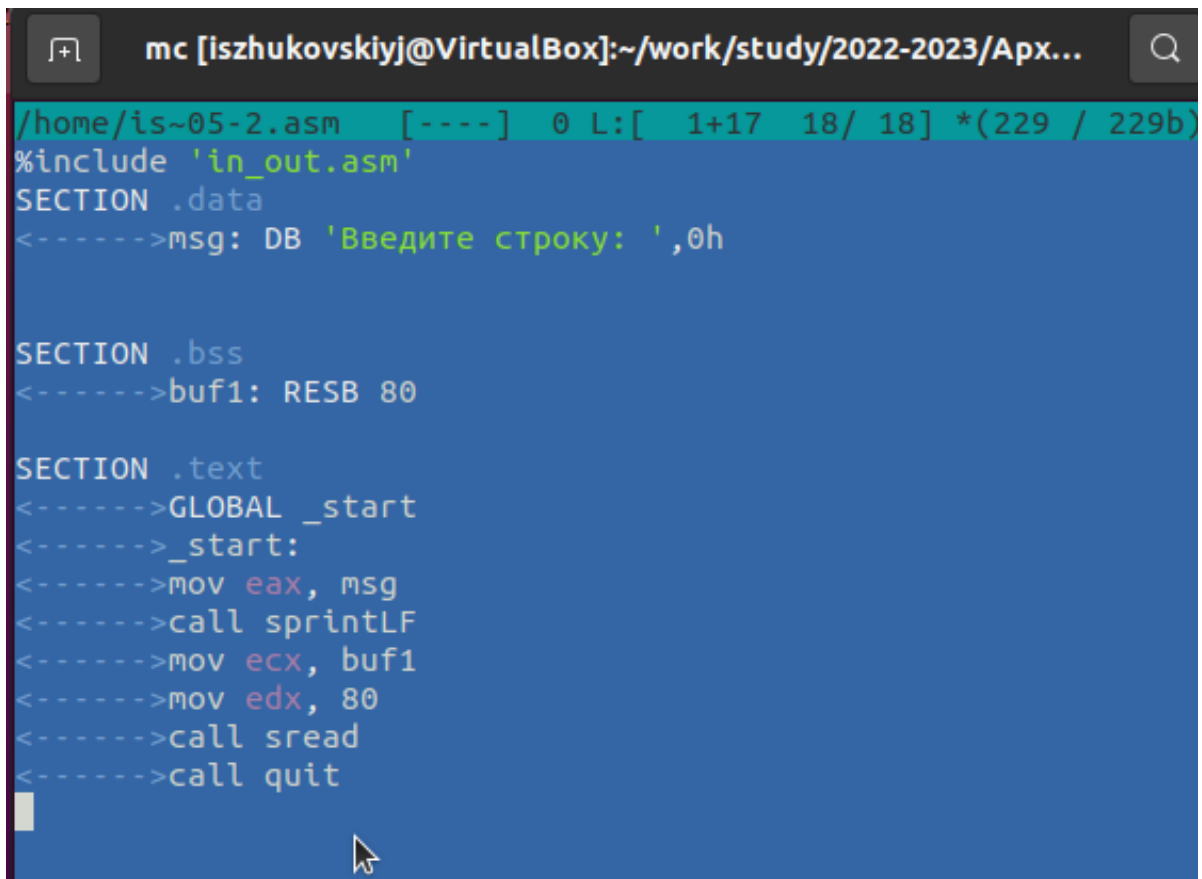


Рис. 3.4: Файл in\_out.asm

4. Изменили код программы. (рис. 3.5, 3.6)

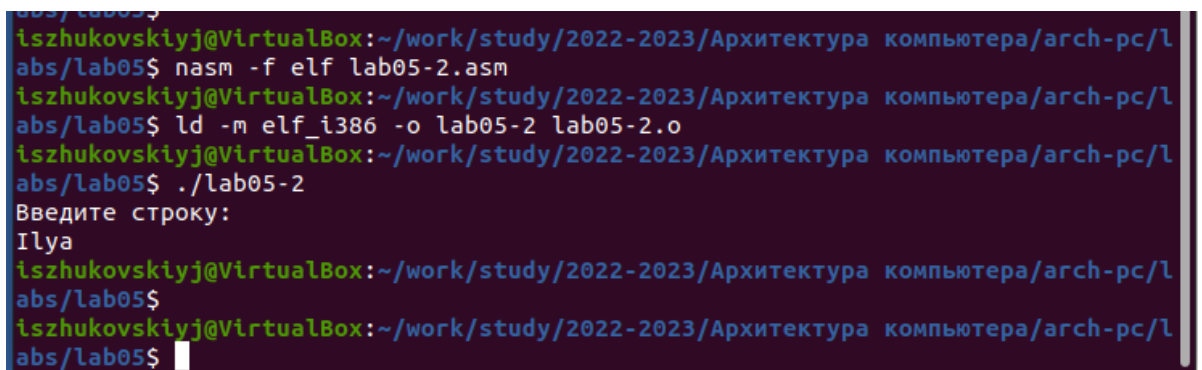


```
mc [iszhukovskiyj@VirtualBox]:~/work/study/2022-2023/Apx...
/home/is~05-2.asm  [----]  0 L:[ 1+17 18/ 18] *(229 / 229b)
%include 'in_out.asm'
SECTION .data
<----->msg: DB 'Введите строку: ',0h

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
```

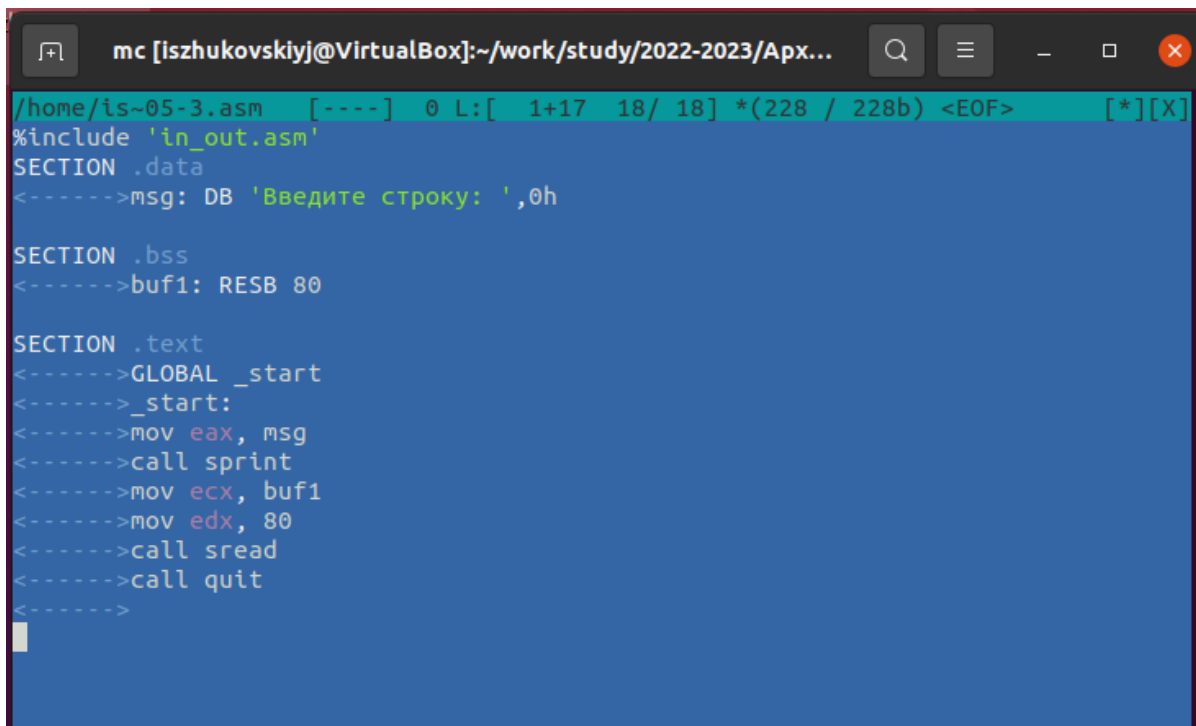
Рис. 3.5: Редактирование файла 2 в Midnight Commander



```
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab05$ nasm -f elf lab05-2.asm
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab05$ ld -m elf_i386 -o lab05-2 lab05-2.o
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab05$ ./lab05-2
Введите строку:
Ilya
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab05$
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab05$
```

Рис. 3.6: Проверка программы 2

5. Изменили вызов подпрограммы. Теперь ввод и вывод в одну строку. (рис. 3.7, 3.8)

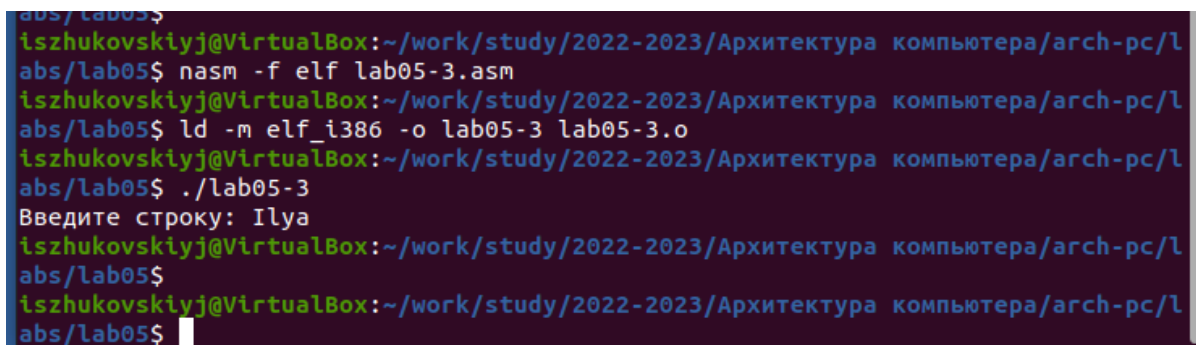


```
mc [iszhukovskiyj@VirtualBox]:~/work/study/2022-2023/Apx...
/home/is~05-3.asm  [----]  0 L:[ 1+17 18/ 18] *(228 / 228b) <EOF>  [*][X]
%include 'in_out.asm'
SECTION .data
<----->msg: DB 'Введите строку: ',0h

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. 3.7: Редактирование файла 3 в Midnight Commander




```
abs/lab05$
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab05$ nasm -f elf lab05-3.asm
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab05$ ld -m elf_i386 -o lab05-3 lab05-3.o
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab05$ ./lab05-3
Введите строку: Ilya
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab05$
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab05$
```

Рис. 3.8: Проверка программы 3

6. Внесем изменения в программу (без использования внешнего файла `in_out.asm`), так чтобы она работала по следующему алгоритму: (рис. 3.9, 3.10)

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;

- вывести введённую строку на экран.

A screenshot of the Midnight Commander (mc) file manager interface. The title bar shows the user 'mc [iszhukovskiy]@VirtualBox' and the current directory '~/work/study/2022-2023/Apx...'. The main window displays the contents of a file named '/home/is~05-4.asm'. The code is written in assembly language and is color-coded: keywords like 'SECTION', 'GLOBAL', and 'int' are in blue, registers and memory locations like 'eax', 'ebx', 'ecx', 'edx', 'buf1', and 'msg' are in red, and string literals like 'Input text' are in green. The code defines a data section with a message 'Input text' of length 10, a bss section with a buffer 'buf1' of size 80, and a text section starting at '\_start'. The '\_start' routine consists of several instructions: setting up registers for a system call (eax=4, ebx=1, ecx=msg, edx=msgLen), performing an interrupt (int 80h), setting up registers for another system call (eax=3, ebx=0, ecx=buf1, edx=80), performing another interrupt (int 80h), and finally setting up registers for a third system call (eax=1, ebx=0) and performing a final interrupt (int 80h).

```
mc [iszhukovskiy]@VirtualBox:~/work/study/2022-2023/Apx...
/home/is~05-4.asm [----] 11 L:[ 1+24 25/ 41] *(287 / 411b)

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

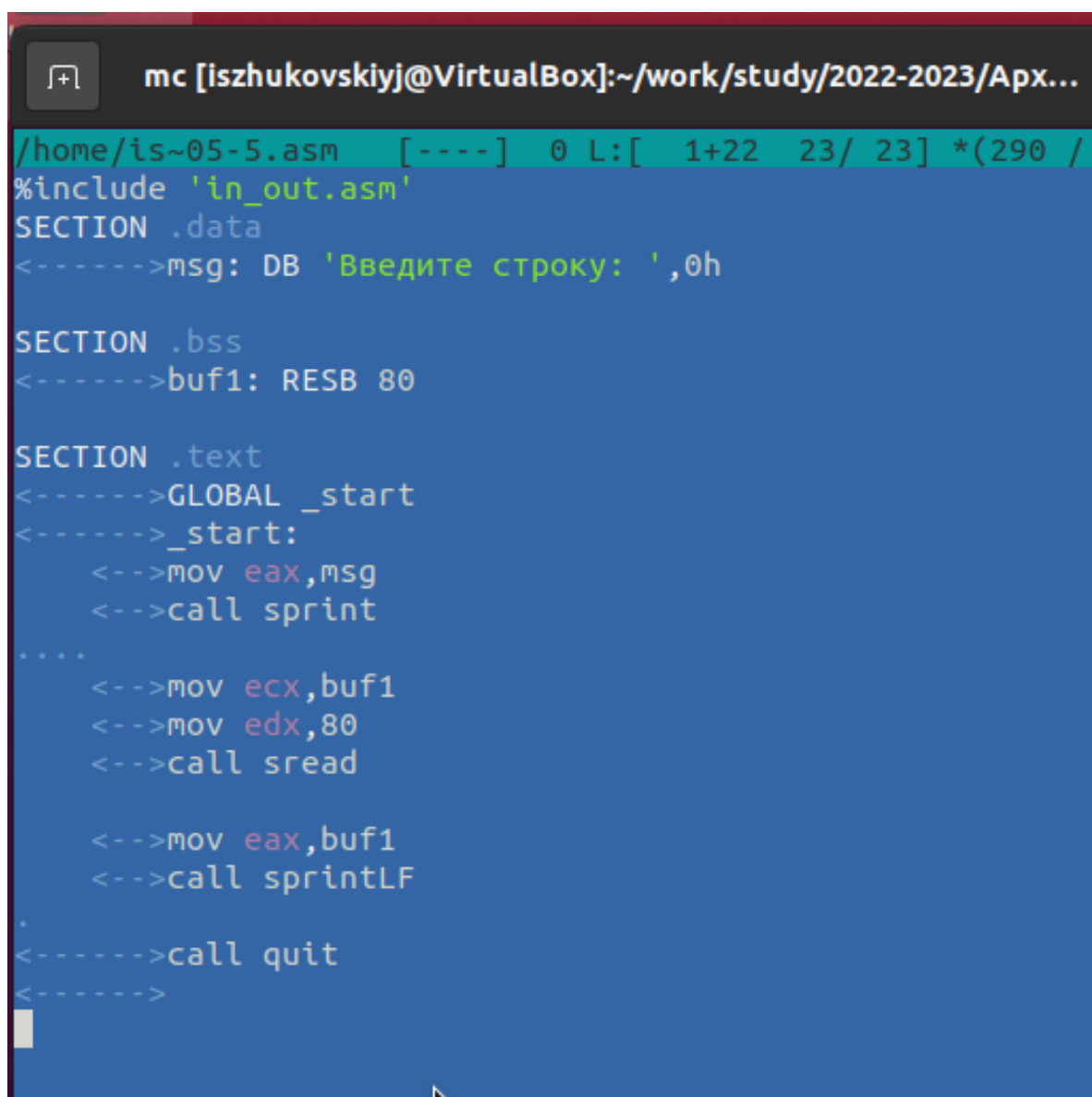
    mov eax,1
    mov ebx,0
    int 80h
. . . .
```

Рис. 3.9: Редактирование файла 4 в Midnight Commander

```
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab05-4.asm
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab05-4 lab05-4.o
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05$ ./lab05-4
Input text
Ilya
Ilya
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05$
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05$
```

Рис. 3.10: Проверка программы 4

7. Сделаем тоже самое с файлом in\_out.asm (рис. 3.11, 3.12)



The image shows a screenshot of the Midnight Commander (mc) file manager and editor. The title bar at the top indicates the user is 'iszhukovskiyj@VirtualBox' and the current directory is '~/work/study/2022-2023/Apx...'. The editor window displays an assembly file named '5.asm'. The code includes a header section, data and bss sections, and a text section with assembly instructions for printing a message and reading input. The status bar at the bottom shows the current line is 1 of 22, and the file size is 23 bytes.

```
/home/is~05-5.asm  [----]  0 L:[ 1+22 23/ 23] *(290 /
%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. 3.11: Редактирование файла 5 в Midnight Commander

```
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab05-5.asm
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab05-5 lab05-5.o
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05$ ./lab05-5
Введите строку: Ilya
Ilya

iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05$
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05$
```

Рис. 3.12: Проверка программы 5



## 4 Выводы

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