

# **Отчёта по лабораторной работе 6**

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

Джозеф Кервенс

# 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 . . . . .	9
2.5	Редактирование файла 2 в Midnight Commander . . . . .	10
2.6	Проверка программы 2 . . . . .	10
2.7	Редактирование файла 3 в Midnight Commander . . . . .	11
2.8	Проверка программы 3 . . . . .	11
2.9	Редактирование файла 4 в Midnight Commander . . . . .	12
2.10	Проверка программы 4 . . . . .	12
2.11	Редактирование файла 5 в Midnight Commander . . . . .	13
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)

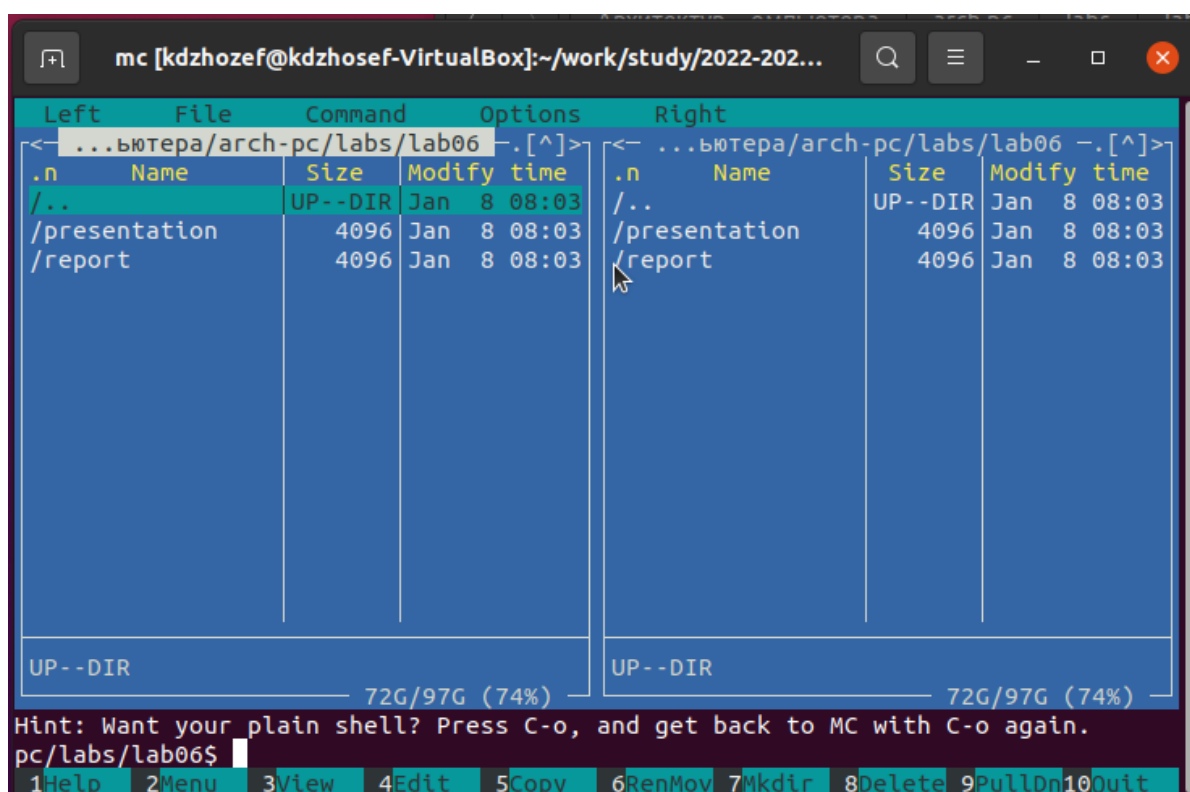
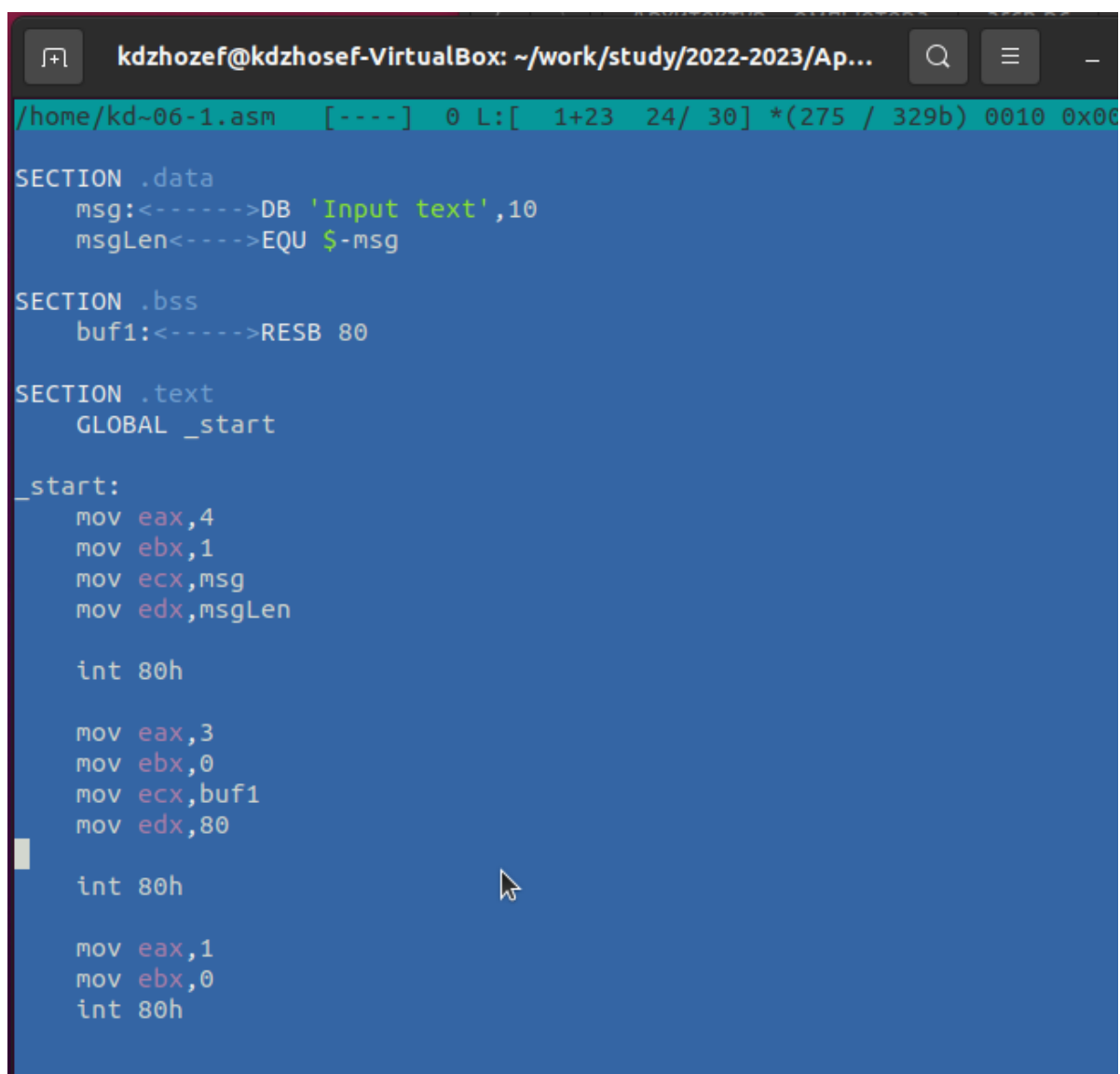


Рис. 2.1: Создание файлов в Midnight Commander

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)



```
kdzhosef@kdzhosef-VirtualBox: ~/work/study/2022-2023/Ap...
/home/kd~06-1.asm  [ - - - ]  0 L: [ 1+23 24/ 30] *(275 / 329b) 0010 0x00

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

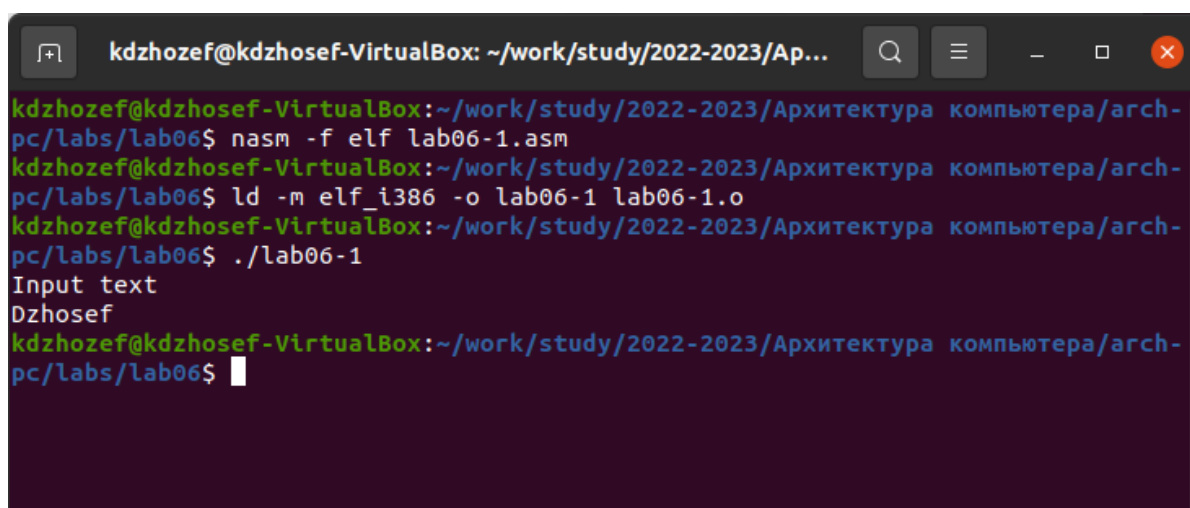
    int 80h

    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



```
kdzhosef@kdzhosef-VirtualBox: ~/work/study/2022-2023/Архитектура компьютера/arch-  
pc/labs/lab06$ nasm -f elf lab06-1.asm  
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-  
pc/labs/lab06$ ld -m elf_i386 -o lab06-1 lab06-1.o  
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-  
pc/labs/lab06$ ./lab06-1  
Input text  
Dzhosef  
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-  
pc/labs/lab06$
```

Рис. 2.3: Проверка программы 1

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



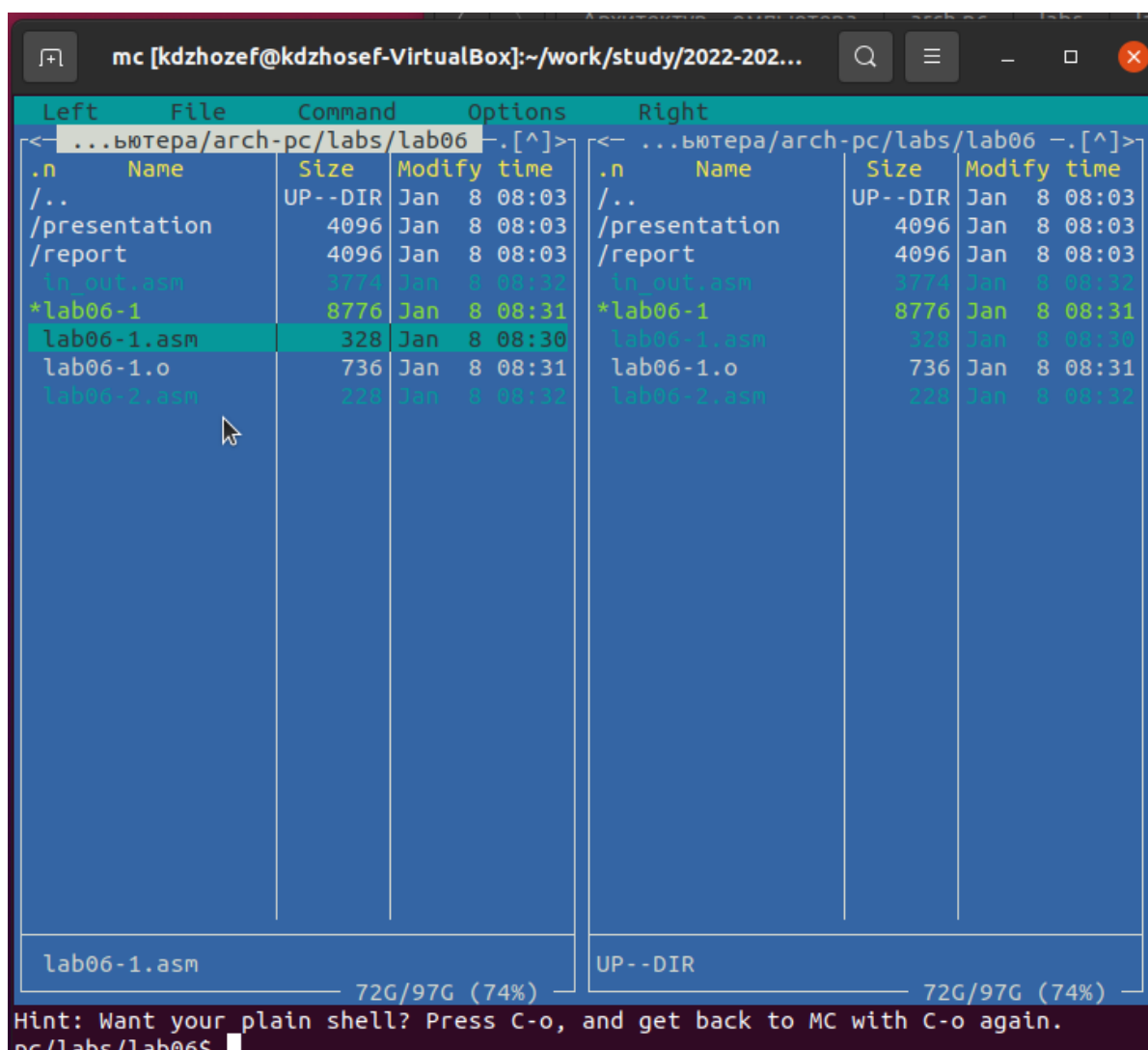
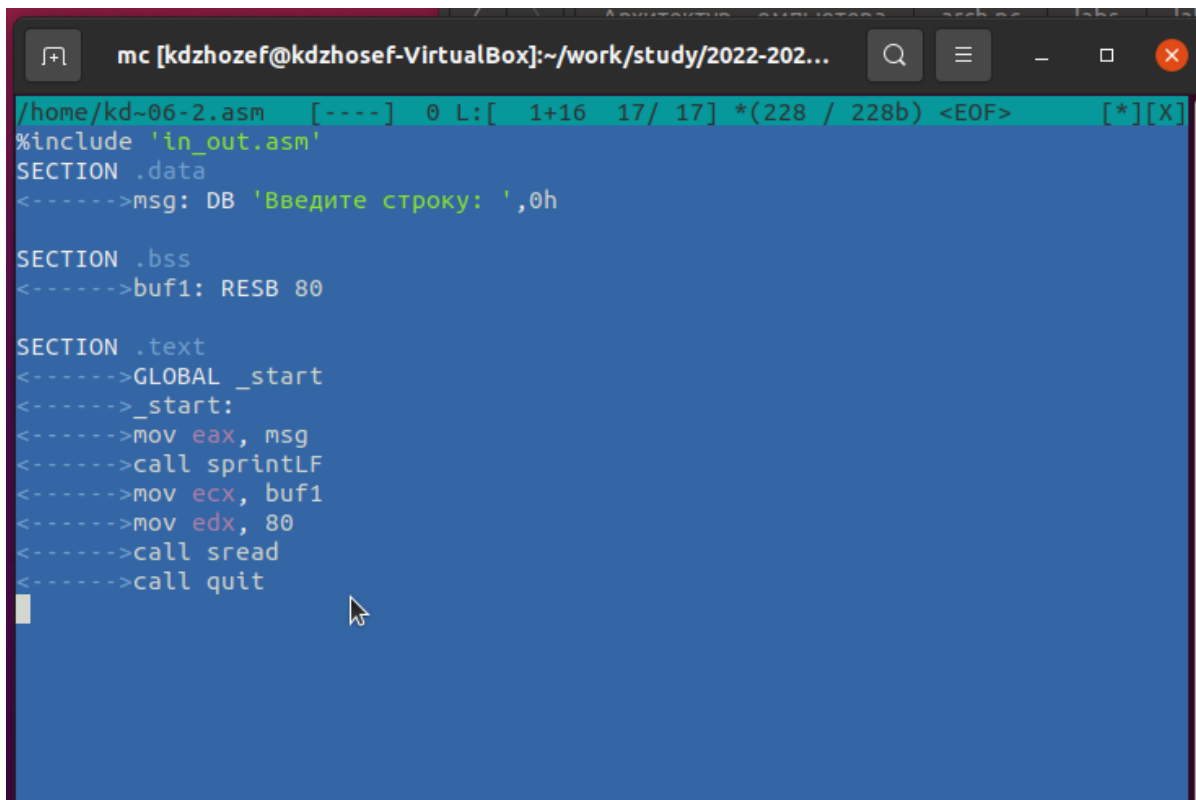


Рис. 2.4: Файл in\_out.asm

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

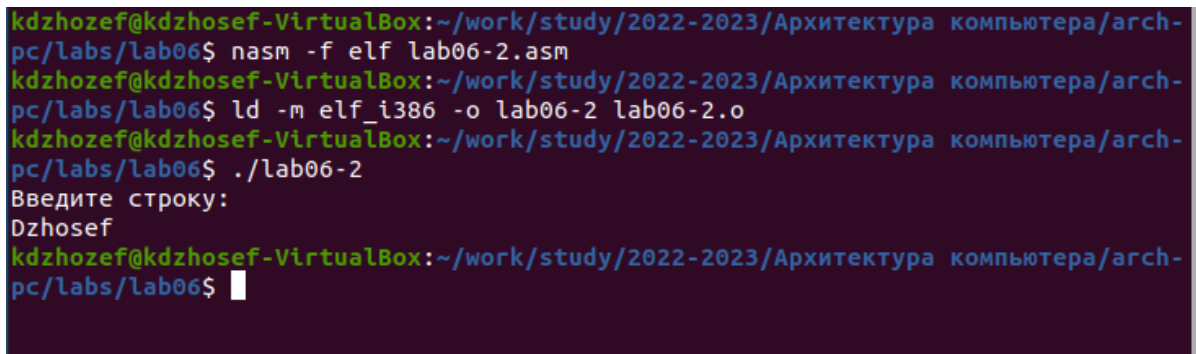


```
mc [kdzhosef@kdzhosef-VirtualBox]:~/work/study/2022-202...
/home/kd~06-2.asm  [----]  0 L:[ 1+16 17/ 17] *(228 / 228b) <EOF>  [*][X]
%include 'in_out.asm'
SECTION .data
<----->msg: DB 'Введите строку: ',0h

SECTION .bss
<----->buf1: RESB 80

SECTION .text
<----->GLOBAL _start
<----->_start:
<----->mov eax, msg
<----->call sprintf
<----->mov ecx, buf1
<----->mov edx, 80
<----->call sread
<----->call quit
```

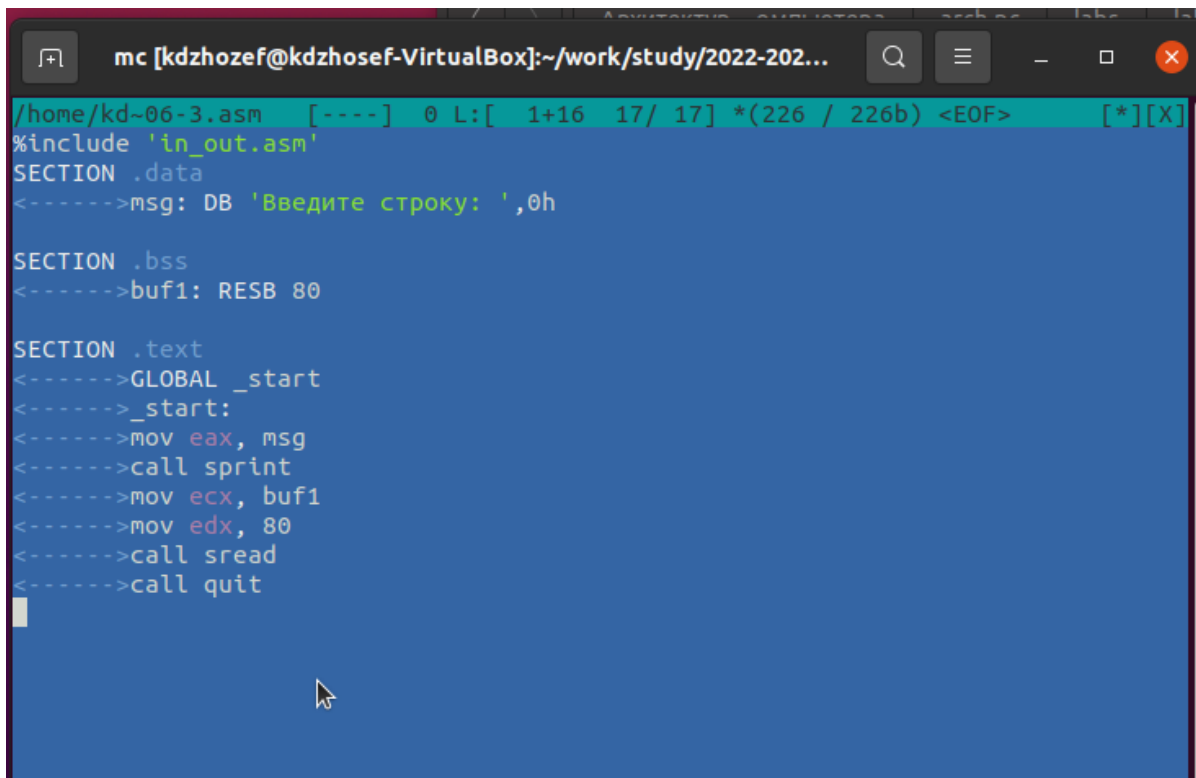
Рис. 2.5: Редактирование файла 2 в Midnight Commander



```
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-
pc/labs/lab06$ nasm -f elf lab06-2.asm
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-
pc/labs/lab06$ ld -m elf_i386 -o lab06-2 lab06-2.o
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-
pc/labs/lab06$ ./lab06-2
Введите строку:
Dzhosef
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-
pc/labs/lab06$
```

Рис. 2.6: Проверка программы 2

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

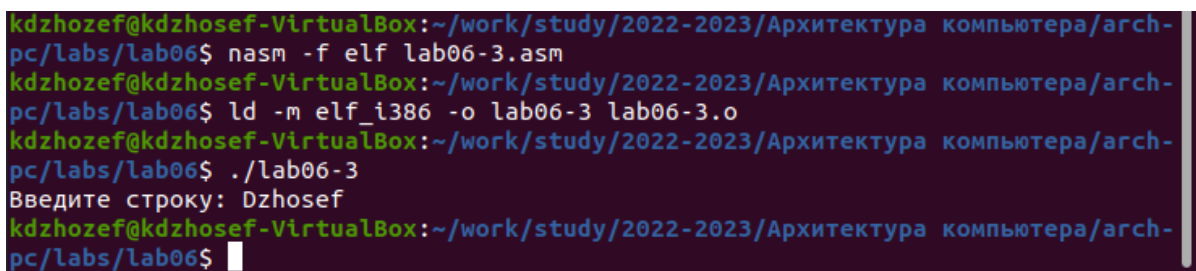
A screenshot of the Midnight Commander file manager in a terminal window. The window title is 'mc [kdzhosef@kdzhosef-VirtualBox]:~/work/study/2022-202...'. The editor shows the file '/home/kd~06-3.asm' with a status bar at the top indicating line 1 of 17. The code includes a directive to include 'in\_out.asm', a data section with a message string 'Введите строку: ', a bss section for a buffer, and a text section with assembly instructions to print the message, read input into the buffer, and quit.

```
mc [kdzhosef@kdzhosef-VirtualBox]:~/work/study/2022-202...
/home/kd~06-3.asm [----] 0 L:[ 1+16 17/ 17] *(226 / 226b) <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
```

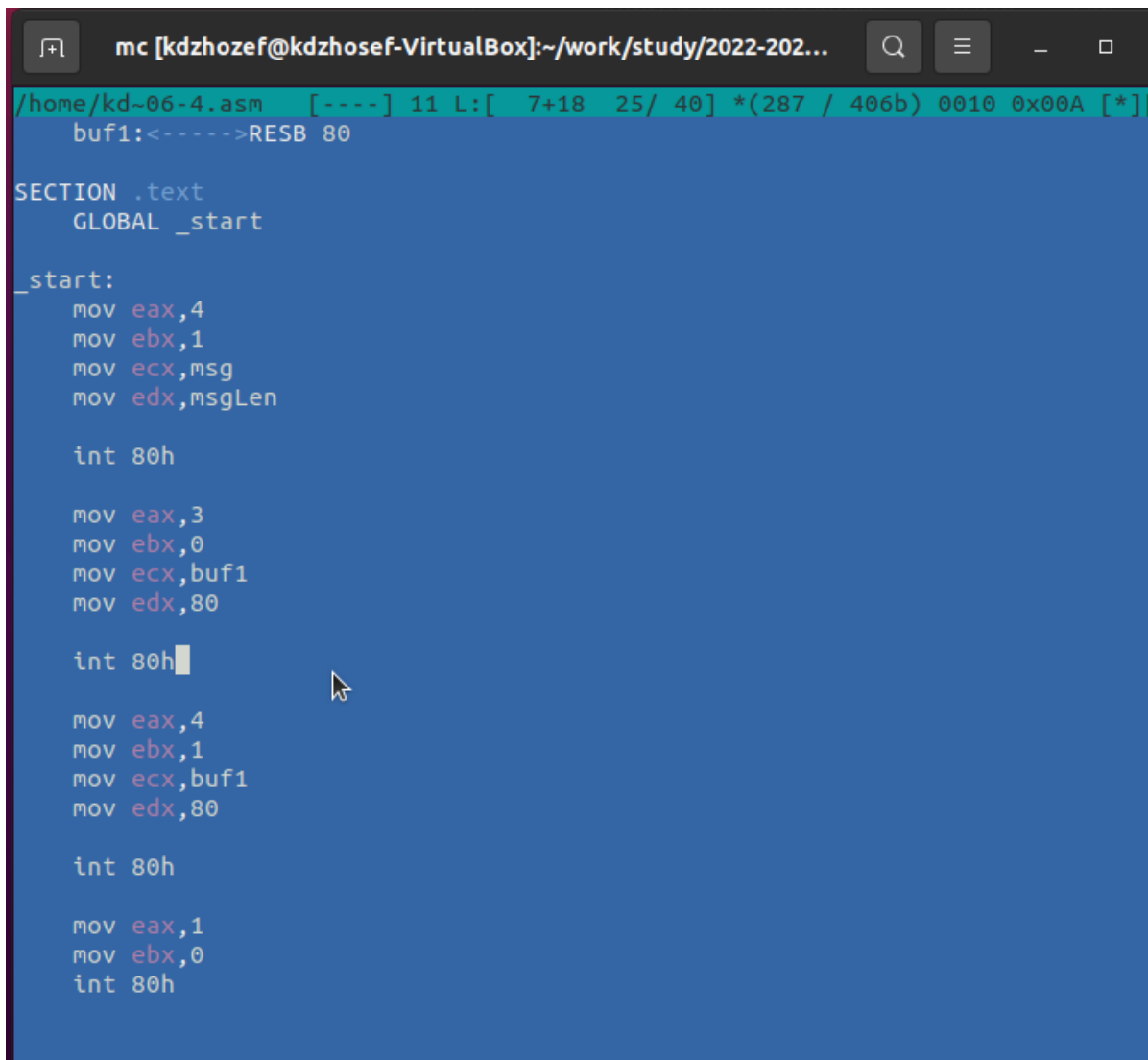
Рис. 2.7: Редактирование файла 3 в Midnight Commander

A screenshot of a terminal window showing the compilation and execution of the assembly program. The user runs 'nasm -f elf lab06-3.asm', then 'ld -m elf\_i386 -o lab06-3 lab06-3.o', and finally './lab06-3'. The program prompts 'Введите строку: ' and the user enters 'Dzhosef'.

```
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-
pc/labs/lab06$ nasm -f elf lab06-3.asm
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-
pc/labs/lab06$ ld -m elf_i386 -o lab06-3 lab06-3.o
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-
pc/labs/lab06$ ./lab06-3
Введите строку: Dzhosef
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-
pc/labs/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 введённую строку на экран.



```
mc [kdzhosef@kdzhosef-VirtualBox]:~/work/study/2022-202...
/home/kd~06-4.asm [---] 11 L:[ 7+18 25/ 40] *(287 / 406b) 0010 0x00A [*]
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

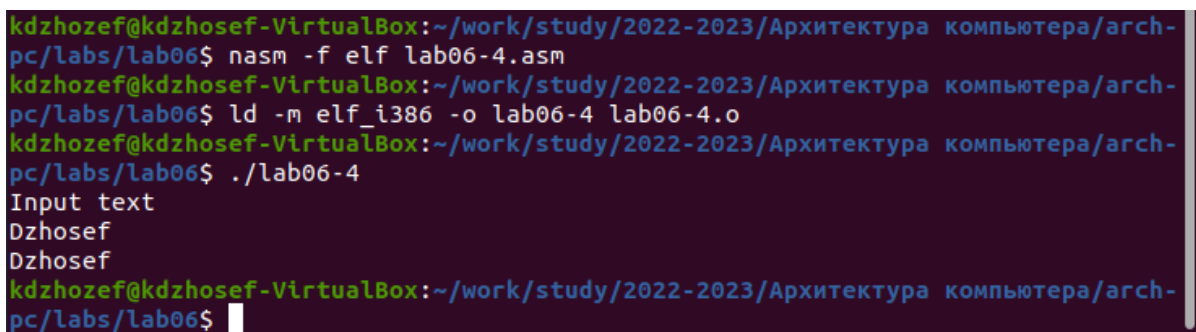
    int 80h

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

    int 80h

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

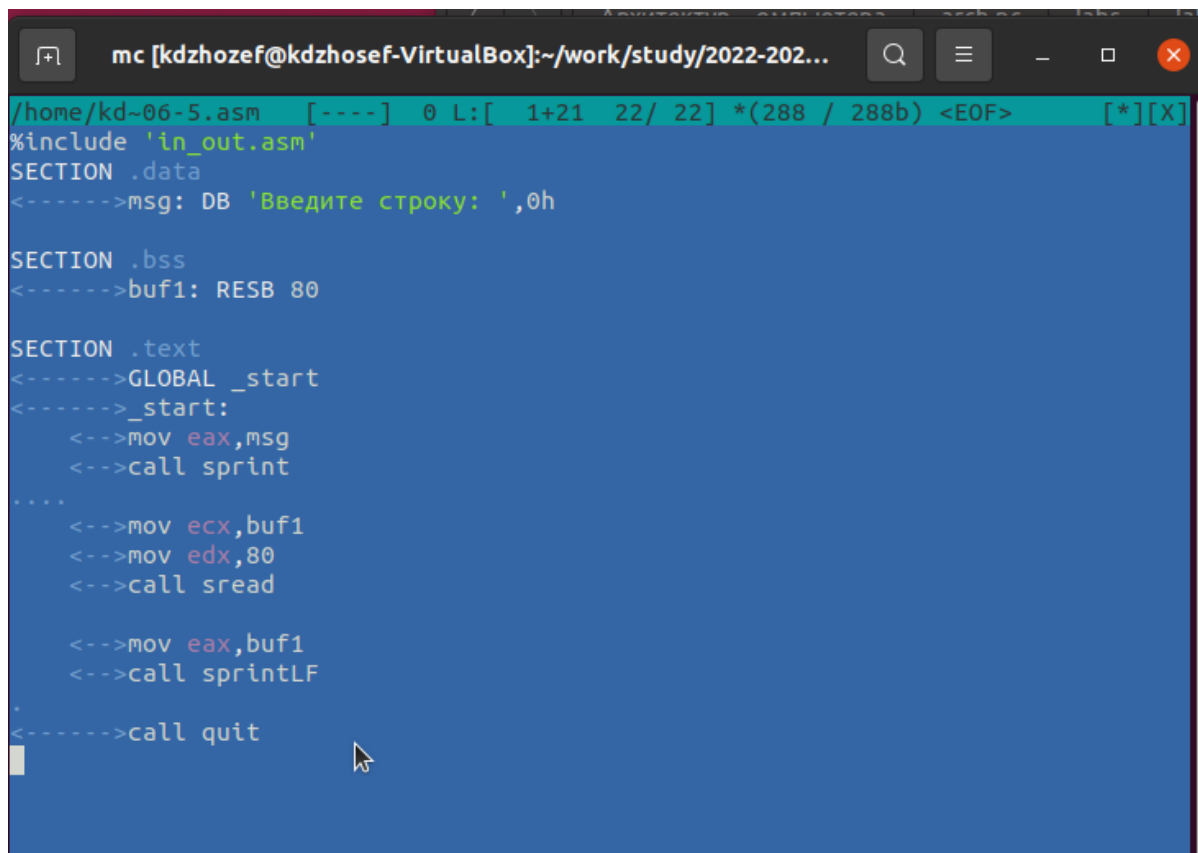
Рис. 2.9: Редактирование файла 4 в Midnight Commander



```
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-
pc/labs/lab06$ nasm -f elf lab06-4.asm
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-
pc/labs/lab06$ ld -m elf_i386 -o lab06-4 lab06-4.o
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-
pc/labs/lab06$ ./lab06-4
Input text
Dzhosef
Dzhosef
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-
pc/labs/lab06$
```

Рис. 2.10: Проверка программы 4

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



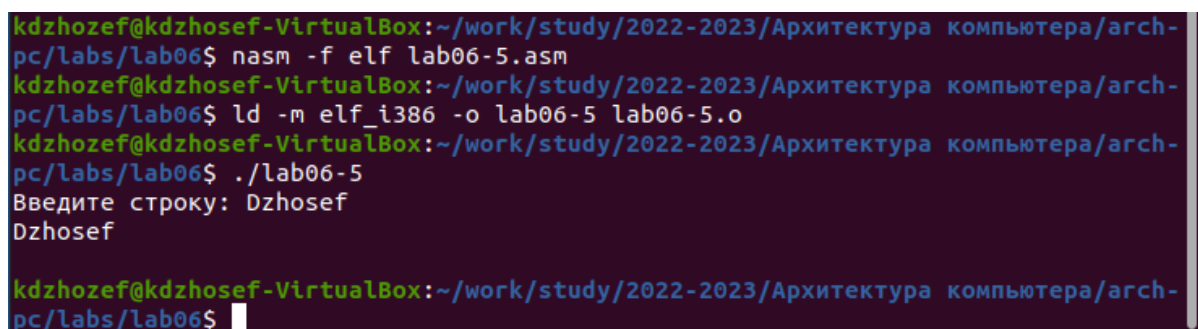
```
mc [kdzhosef@kdzhosef-VirtualBox]:~/work/study/2022-2023...
/home/kd~06-5.asm  [----] 0 L:[ 1+21 22/ 22] *(288 / 288b) <EOF>  [*][X]
%include 'in_out.asm'
SECTION .data
<----->msg: DB 'Введите строку: ',0h

SECTION .bss
<----->buf1: RESB 80

SECTION .text
<----->GLOBAL _start
<----->_start:
    <-->mov eax,msg
    <-->call sprint
    ....
    <-->mov ecx,buf1
    <-->mov edx,80
    <-->call sread

    <-->mov eax,buf1
    <-->call sprintLF
    *
<----->call quit
```

Рис. 2.11: Редактирование файла 5 в Midnight Commander



```
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-
pc/labs/lab06$ nasm -f elf lab06-5.asm
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-
pc/labs/lab06$ ld -m elf_i386 -o lab06-5 lab06-5.o
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-
pc/labs/lab06$ ./lab06-5
Введите строку: Dzhosef
Dzhosef
kdzhosef@kdzhosef-VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-
pc/labs/lab06$
```

Рис. 2.12: Проверка программы 5

## 3 Выводы

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