

# **Отчёт по лабораторной работе 5**

**Архитектура компьютеров и операционные системы**

Горелашвили Лия Михайловна НКАбд-02-23

# 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>
4.1	Знакомство с Midnight Commander . . . . .	8
4.2	Подключение внешнего файла in_out.asm . . . . .	12
4.3	Выполнение заданий для самостоятельной работы . . . . .	16
<b>5</b>	<b>Выводы</b>	<b>19</b>

## Список иллюстраций

4.1	окно Midnight Commander . . . . .	8
4.2	Создание каталога . . . . .	9
4.3	Создала файл lab05-1.asm . . . . .	9
4.4	Редактирование файла lab05-1.asm . . . . .	10
4.5	Проверка кода lab05-1.asm . . . . .	11
4.6	Тестирование программы lab05-1.asm . . . . .	11
4.7	Копирование файла in_out.asm . . . . .	12
4.8	Копирование файла lab05-1.asm . . . . .	13
4.9	Редактирование файла lab05-2.asm . . . . .	14
4.10	Тестирование программы lab05-2.asm . . . . .	14
4.11	Редактирование файла lab05-2.asm . . . . .	15
4.12	Тестирование программы lab05-2.asm . . . . .	15
4.13	Редактирование файла lab05-3.asm . . . . .	16
4.14	Тестирование программы lab05-3.asm . . . . .	17
4.15	Редактирование файла lab05-4.asm . . . . .	17
4.16	Тестирование программы lab05-4.asm . . . . .	18

## 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

Открыла Midnight Commander и с помощью клавиш со стрелками и Enter перешла в каталог ~/work/arch-pc. Затем нажала F7 и создала каталог с названием lab05.

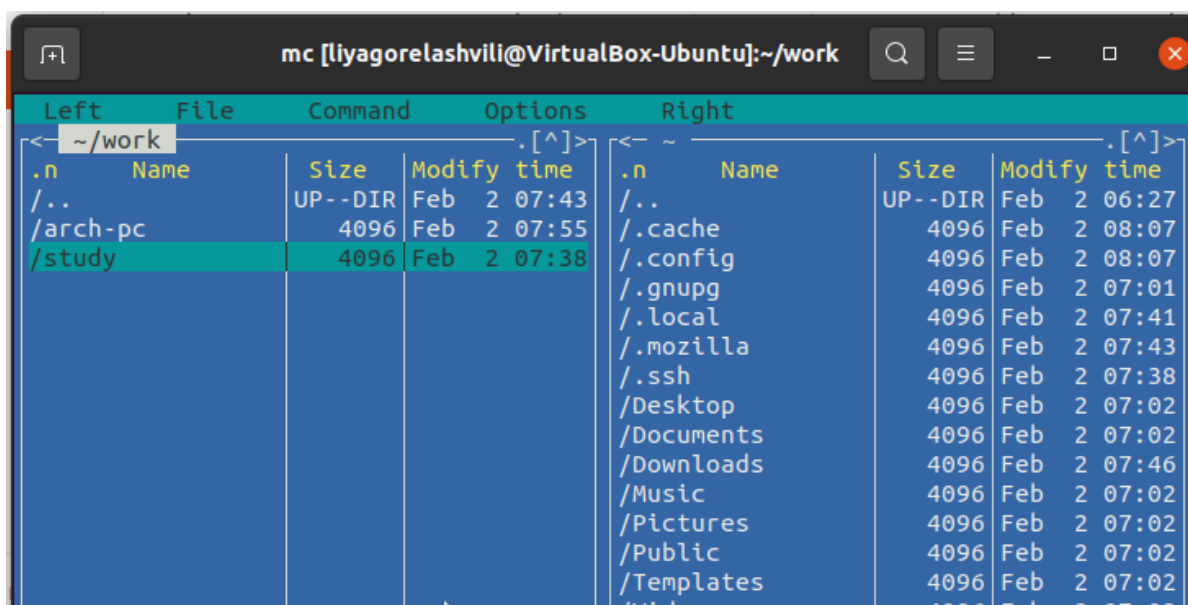


Рис. 4.1: окно Midnight Commander



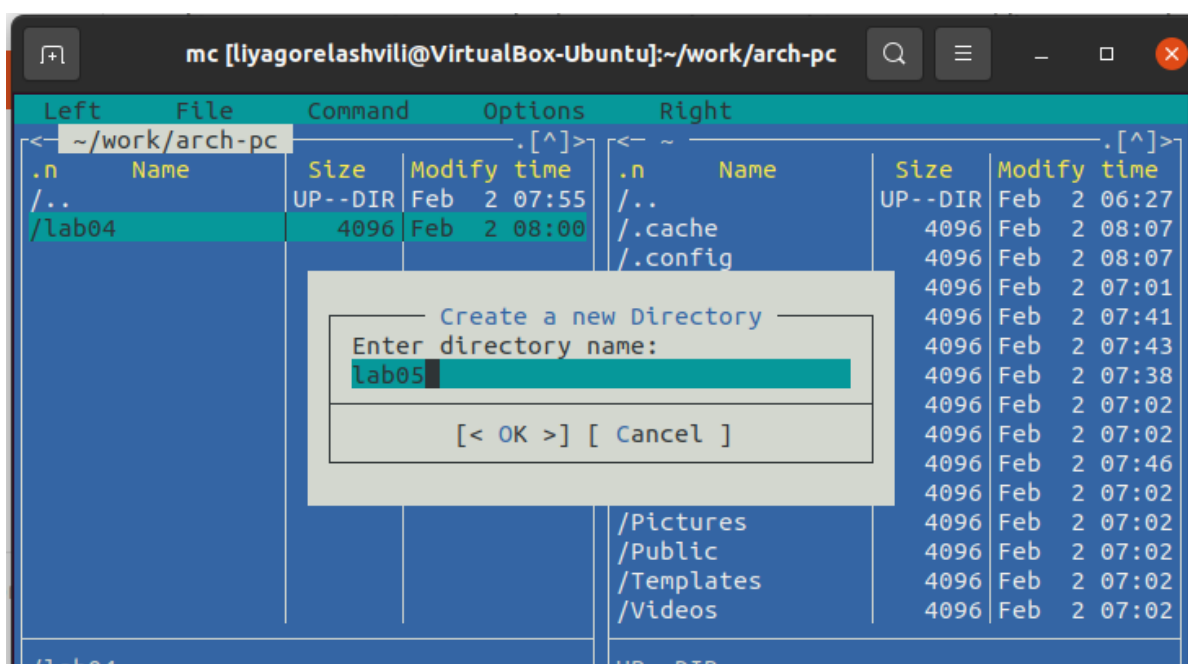


Рис. 4.2: Создание каталога

Используя команду touch, создала файл с именем lab05-1.asm.

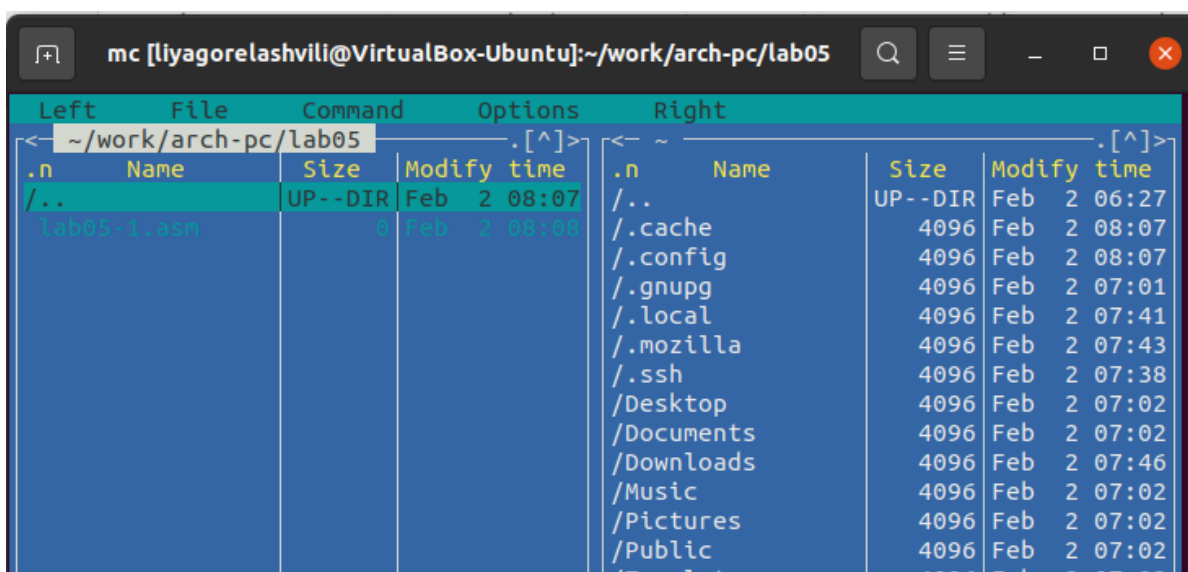
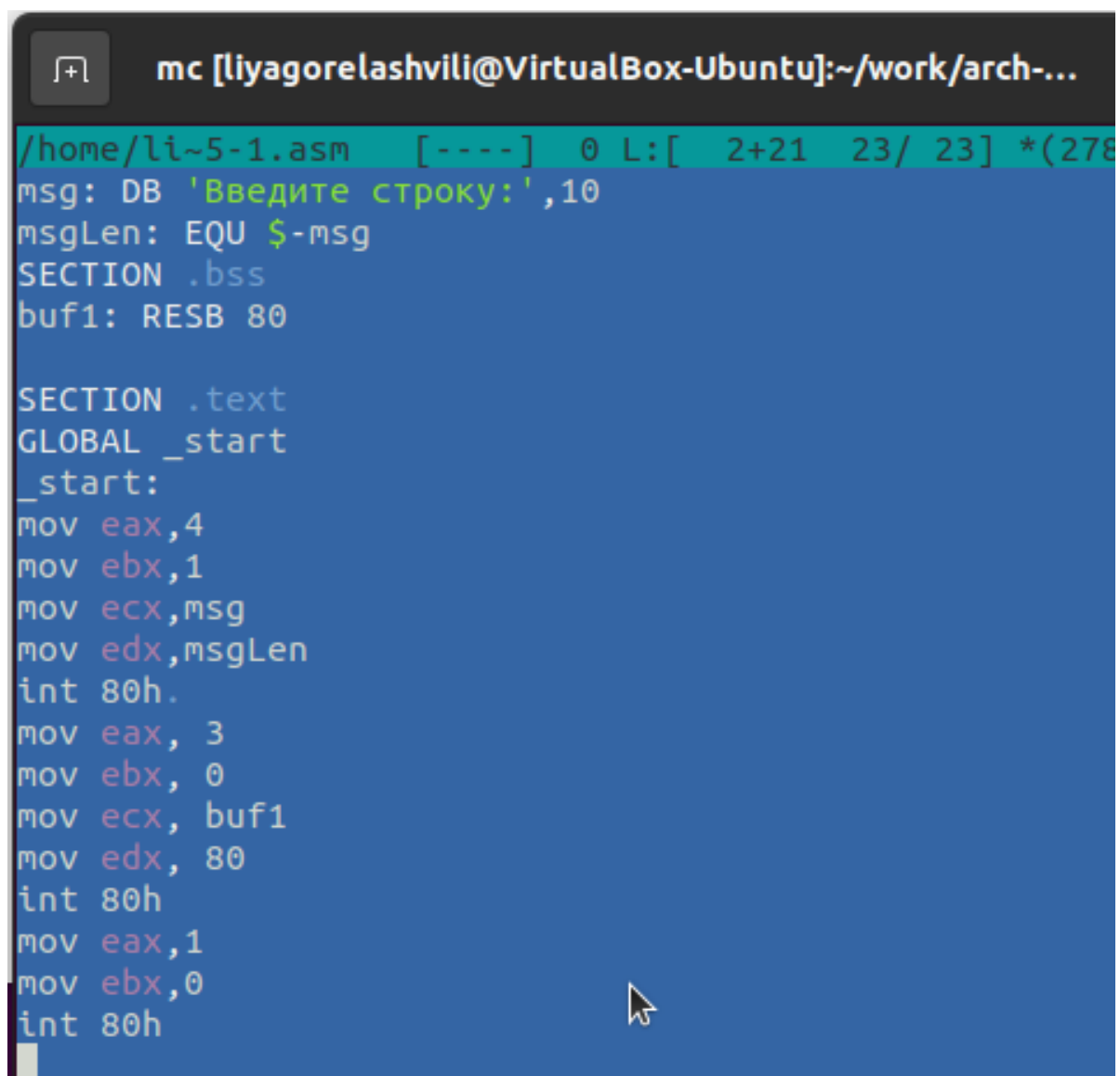


Рис. 4.3: Создала файл lab05-1.asm

Затем открыла файл для редактирования, нажав клавишу F4, и выбрала редактор mceditor. Написала код программы, соответствующий заданию.

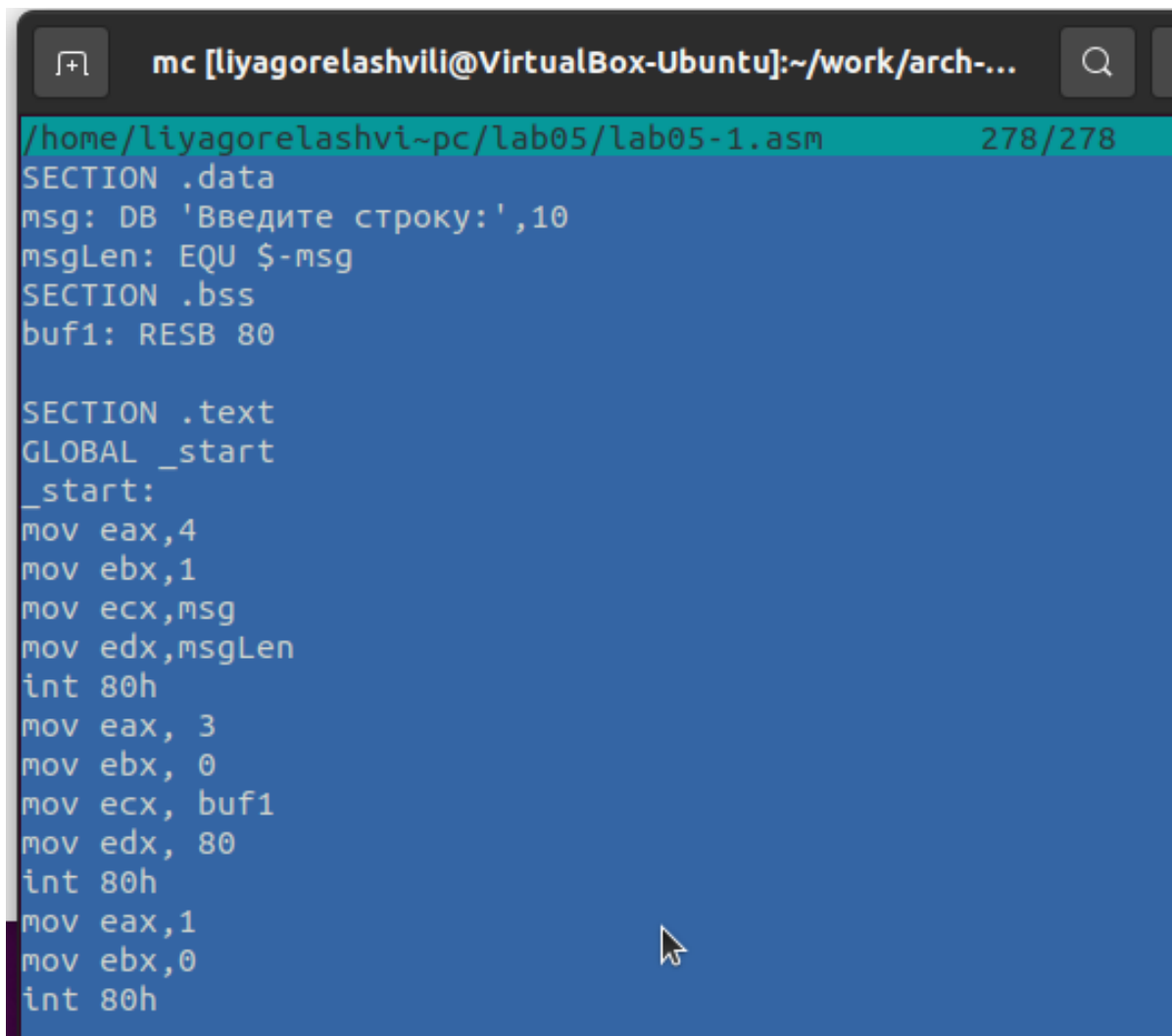
A screenshot of a terminal window with a dark background. The title bar at the top shows a window icon and the text 'mc [liyagorelashvili@VirtualBox-Ubuntu]:~/work/arch-...'. The terminal content shows assembly code being edited. The first line is a status bar: '/home/li~5-1.asm [----] 0 L:[ 2+21 23/ 23] \*(278'. Below it, the code is: 'msg: DB 'Введите строку:',10', 'msgLen: EQU \$-msg', 'SECTION .bss', 'buf1: RESB 80', a blank line, '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', and 'int 80h'. A mouse cursor is visible over the last line of code.

```
mc [liyagorelashvili@VirtualBox-Ubuntu]:~/work/arch-...
/home/li~5-1.asm [----] 0 L:[ 2+21 23/ 23] *(278
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: Редактирование файла lab05-1.asm

Далее открыла файл для просмотра, нажав клавишу F3, и убедилась, что он содержит написанный код.

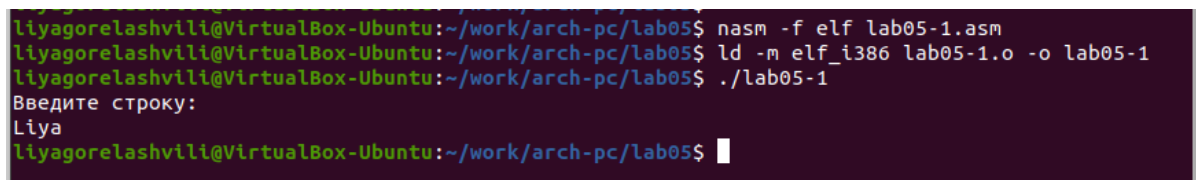
A screenshot of a code editor window titled 'mc [liyagorelashvili@VirtualBox-Ubuntu]:~/work/arch-...'. The editor displays assembly code for 'lab05-1.asm'. The code includes sections for data, bss, and text. The text section contains assembly instructions for setting up a string input routine. The code is as follows:

```
/home/liyagorelashvi~pc/lab05/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, проверила работу программы.

A screenshot of a terminal window showing the compilation and execution of the assembly program. The commands and output are as follows:

```
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab05-1.asm
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-1.o -o lab05-1
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$ ./lab05-1
Введите строку:
Liya
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$
```

Рис. 4.6: Тестирование программы lab05-1.asm

## 4.2 Подключение внешнего файла in\_out.asm

Скачала файл “in\_out.asm” и разместила его в рабочем каталоге.

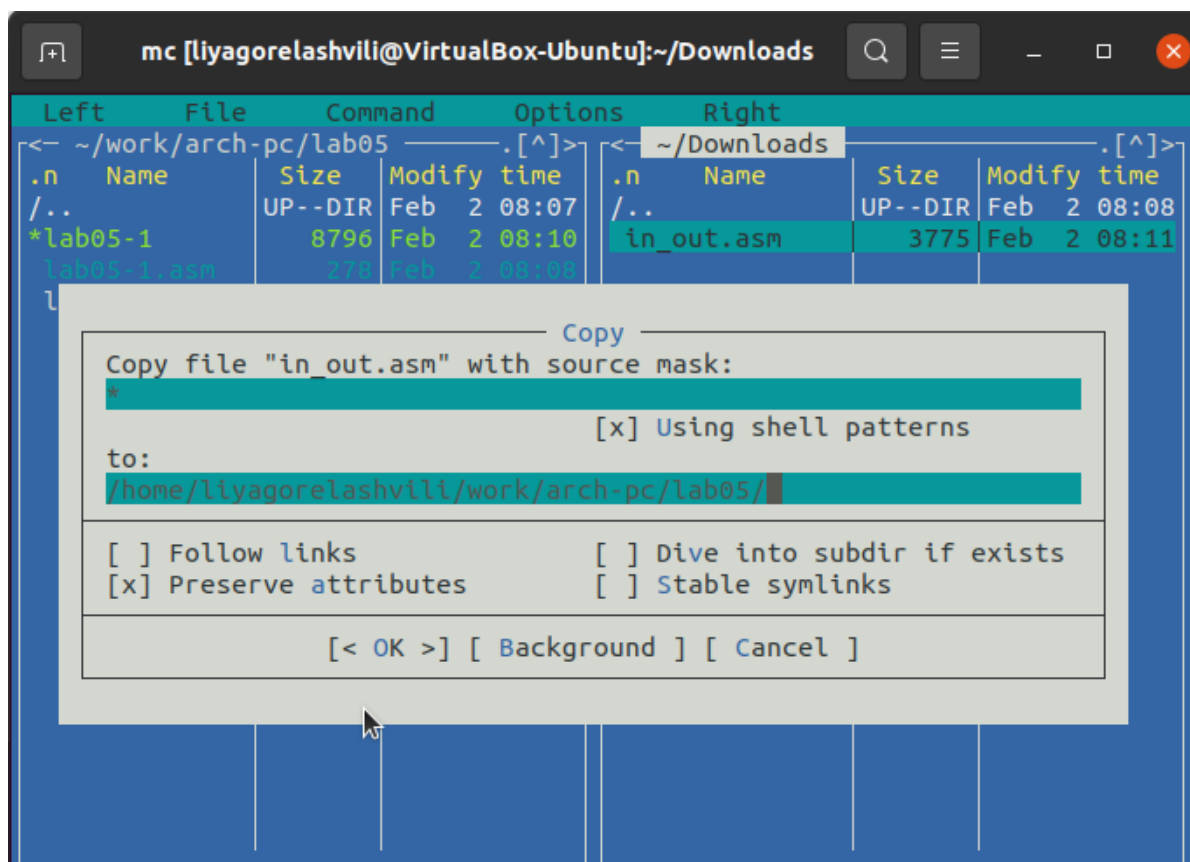


Рис. 4.7: Копирование файла in\_out.asm

С помощью клавиши F5 скопировала содержимое файла lab05-1.asm в файл lab05-2.asm.

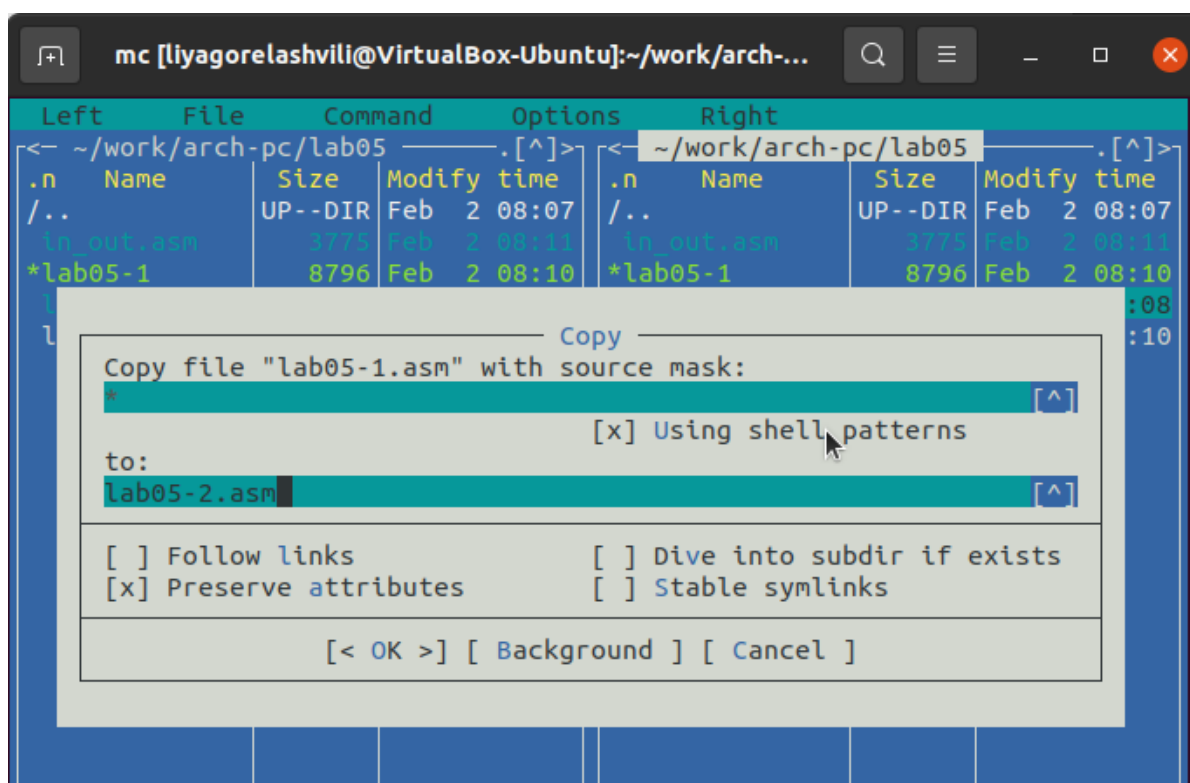
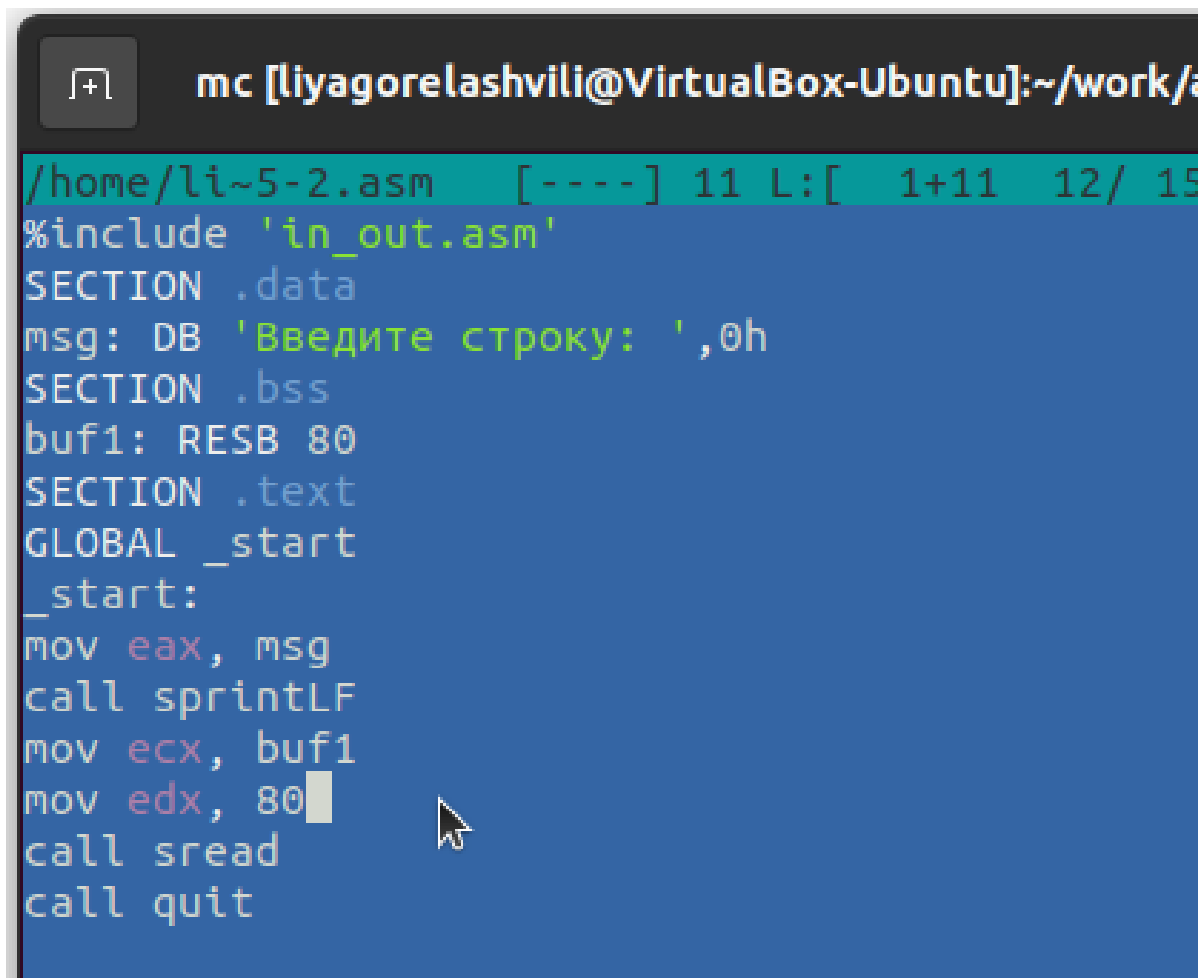


Рис. 4.8: Копирование файла lab05-1.asm

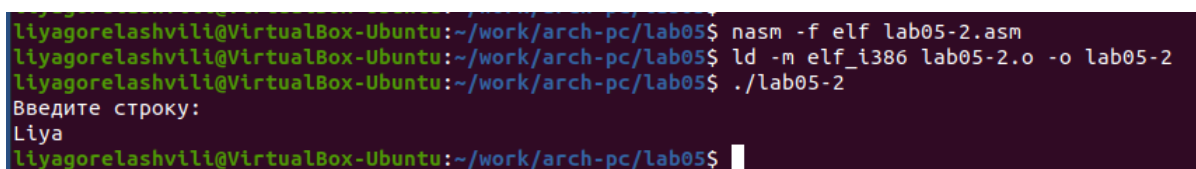
Затем написала код программы lab05-2.asm, используя подпрограммы из внешнего файла in\_out.asm.

A screenshot of a text editor window titled 'mc [liyagorelashvili@VirtualBox-Ubuntu]:~/work/a'. The editor shows the contents of a file named 'lab05-2.asm'. The code is written in assembly language and includes comments in Russian. The code defines a data section with a message, a bss section with a buffer, and a text section with a global \_start function. The \_start function prints the message and reads input from the user. The code is as follows:

```
/home/li~5-2.asm [----] 11 L:[ 1+11 12/ 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 sprintLF
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рис. 4.9: Редактирование файла lab05-2.asm

Скомпилировала программу и проверила ее запуск.

A screenshot of a terminal window showing the compilation and execution of the assembly program. The user runs the following commands:

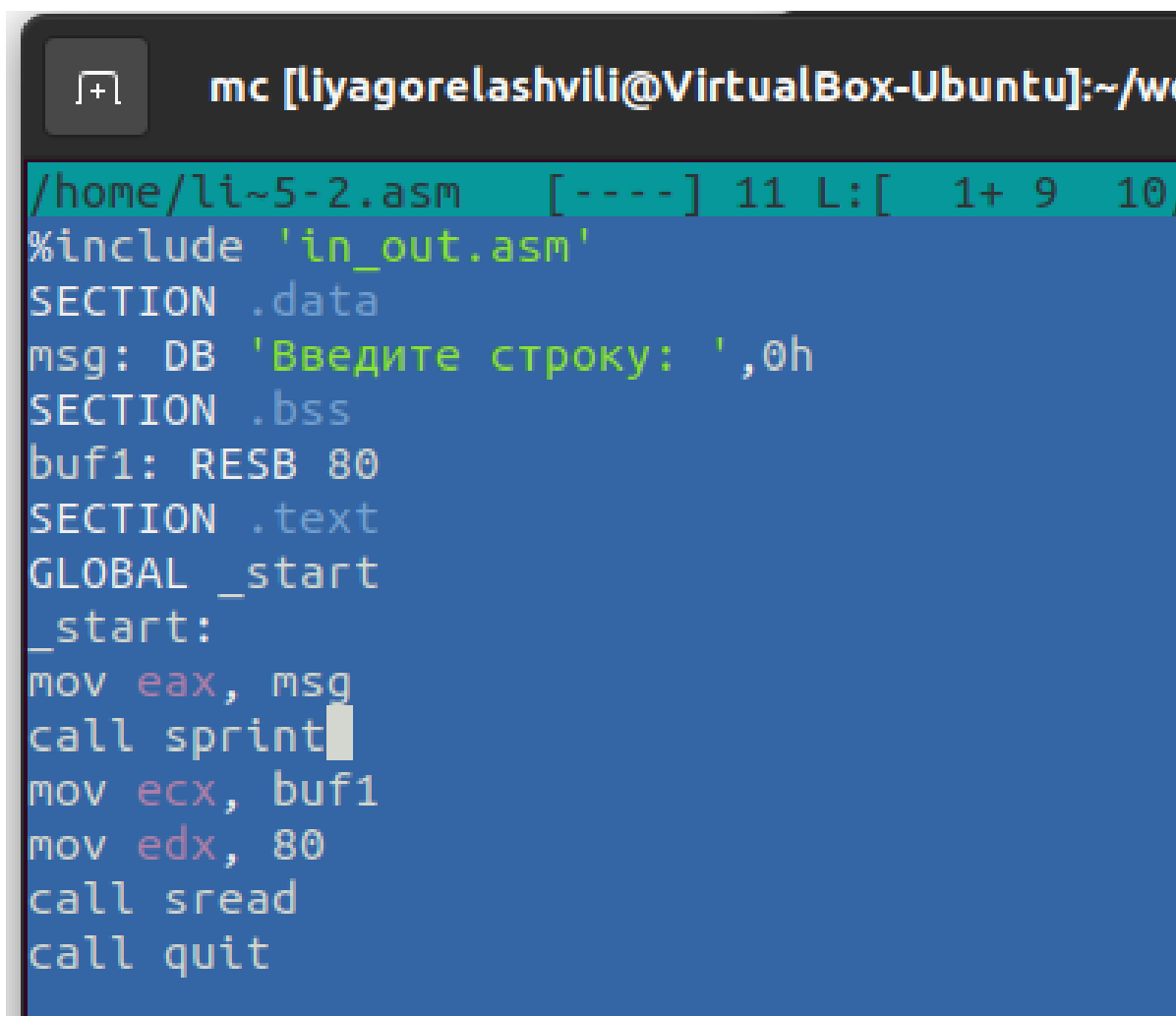
```
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$ ./lab05-2
```

The output of the program is shown as:

```
Введите строку:
Liya
```

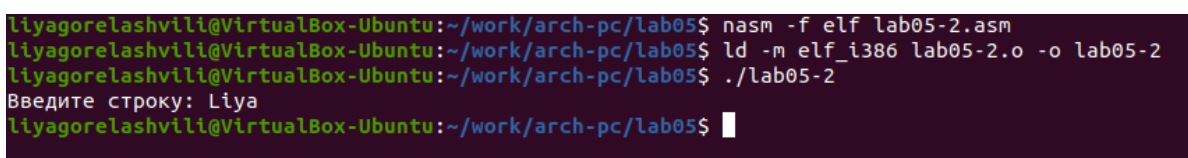
Рис. 4.10: Тестирование программы lab05-2.asm

В файле lab05-2.asm заменила вызов подпрограммы sprintLF на sprint. Пересобрала исполняемый файл. Теперь после вывода строки символ перехода на новую строку отсутствует.

A screenshot of a text editor window titled 'mc [liyagorelashvili@VirtualBox-Ubuntu]:~/w'. The editor shows the contents of a file named 'lab05-2.asm'. The code is written in assembly language and includes comments in Russian. The code defines a data section with a message, a bss section with a buffer, and a text section with a main routine that prints the message and reads input from the user. The code is as follows:

```
/home/li~5-2.asm [----] 11 L:[ 1+ 9 10]
%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: Редактирование файла lab05-2.asm

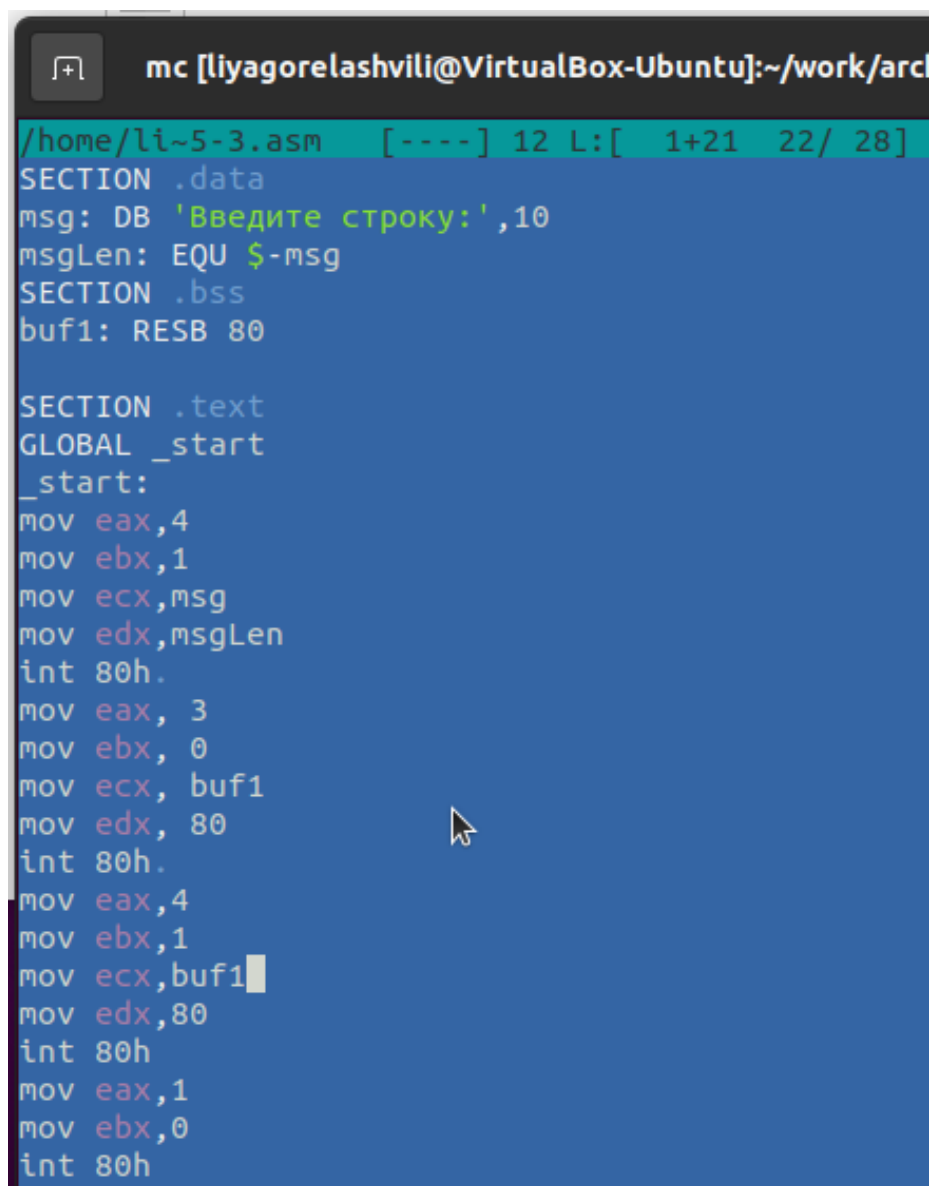
A screenshot of a terminal window showing the compilation and execution of the assembly program. The user runs 'nasm -f elf lab05-2.asm' to compile the assembly code into an ELF executable. Then, they run 'ld -m elf\_i386 lab05-2.o -o lab05-2' to link the object file into an executable. Finally, they run './lab05-2' to execute the program. The program prompts the user to enter a string, and the user enters 'Liya'. The terminal output is as follows:

```
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$ ./lab05-2
Введите строку: Liya
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$
```

Рис. 4.12: Тестирование программы lab05-2.asm

## 4.3 Выполнение заданий для самостоятельной работы

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.



```
mc [liyagorelashvili@VirtualBox-Ubuntu]:~/work/arc  
/home/li~5-3.asm [----] 12 L:[ 1+21 22/ 28]  
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: Редактирование файла lab05-3.asm



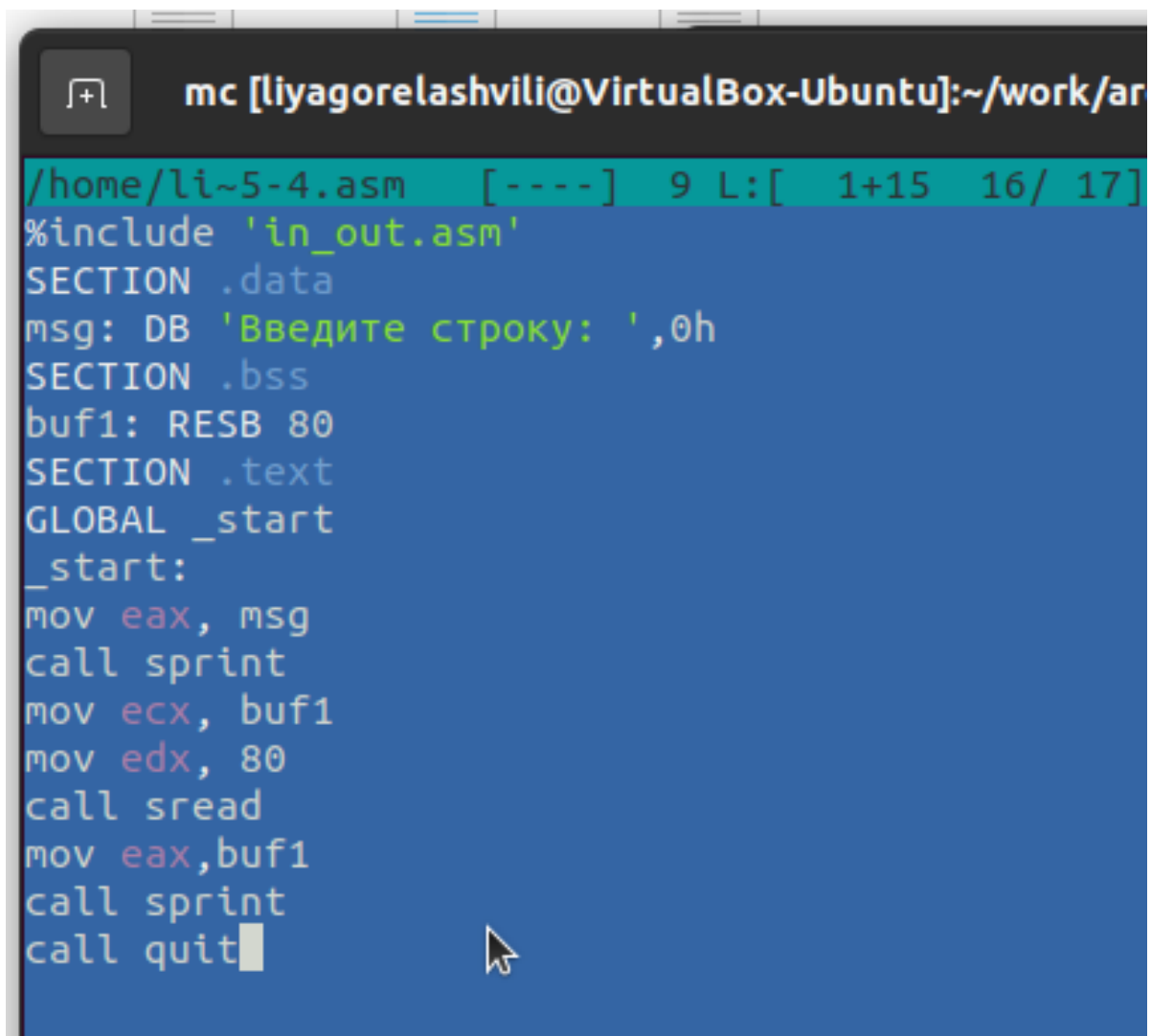
```

liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab05-3.asm
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$ ./lab05-3
Введите строку:
Liya
Liya
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$

```

Рис. 4.14: Тестирование программы lab05-3.asm

Аналогично скопировала программу lab05-2.asm и изменила код, но теперь использовала подпрограммы из файла in\_out.asm.



```

mc [liyagorelashvili@VirtualBox-Ubuntu]:~/work/ar
/home/li~5-4.asm [----] 9 L:[ 1+15 16/ 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: Редактирование файла lab05-4.asm

```
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab05-4.asm
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-4.o -o lab05-4
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$ ./lab05-4
Введите строку: Liya
Liya
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab05$
```

Рис. 4.16: Тестирование программы lab05-4.asm

## 5 Выводы

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