

# **Отчёт по лабораторной работе 5**

**Архитектура компьютеров и операционных систем**

Игнатова Анастасия НБИбд-01-23

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задания</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>18</b>

## Список иллюстраций

4.1	Запуск Midnight Commander . . . . .	8
4.2	Создание файла lab05-1.asm . . . . .	9
4.3	Программа в файле lab05-1.asm . . . . .	10
4.4	Просмотр файла lab05-1.asm . . . . .	11
4.5	Запуск программы lab05-1.asm . . . . .	12
4.6	Копирование файла lab05-1.asm . . . . .	12
4.7	Программа в файле lab05-2.asm . . . . .	13
4.8	Запуск программы lab05-2.asm . . . . .	13
4.9	Программа в файле lab05-2.asm . . . . .	14
4.10	Запуск программы lab05-2.asm . . . . .	14
4.11	Программа в файле lab05-3.asm . . . . .	15
4.12	Запуск программы lab05-3.asm . . . . .	16
4.13	Программа в файле lab05-4.asm . . . . .	17
4.14	Запуск программы lab05-4.asm . . . . .	17

## **Список таблиц**

# 1 Цель работы

Целью работы является приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.

## 2 Задания

1. Освоить возможности Midnight Commander
2. Изучить примеры программ с использованием внешнего файла `in_out.asm`
3. Выполнить задание по программе
4. Подготовить отчет и загрузить 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`).

Инструкция языка ассемблера `mov` предназначена для дублирования данных источника в приёмнике. В общем виде эта инструкция записывается в виде `mov dst,src` Здесь операнд `dst` — приёмник, а `src` — источник

Инструкция языка ассемблера `int` предназначена для вызова прерывания с указанным номером. В общем виде она записывается в виде `int n` Здесь `n` — номер прерывания, принадлежащий диапазону 0–255

## 4 Выполнение лабораторной работы

Открыла Midnight Commander, с помощью клавиш со стрелками и Enter перехожу в каталог ~/work/arch-рс. Далее нажимаю F7 и создаю каталог lab05

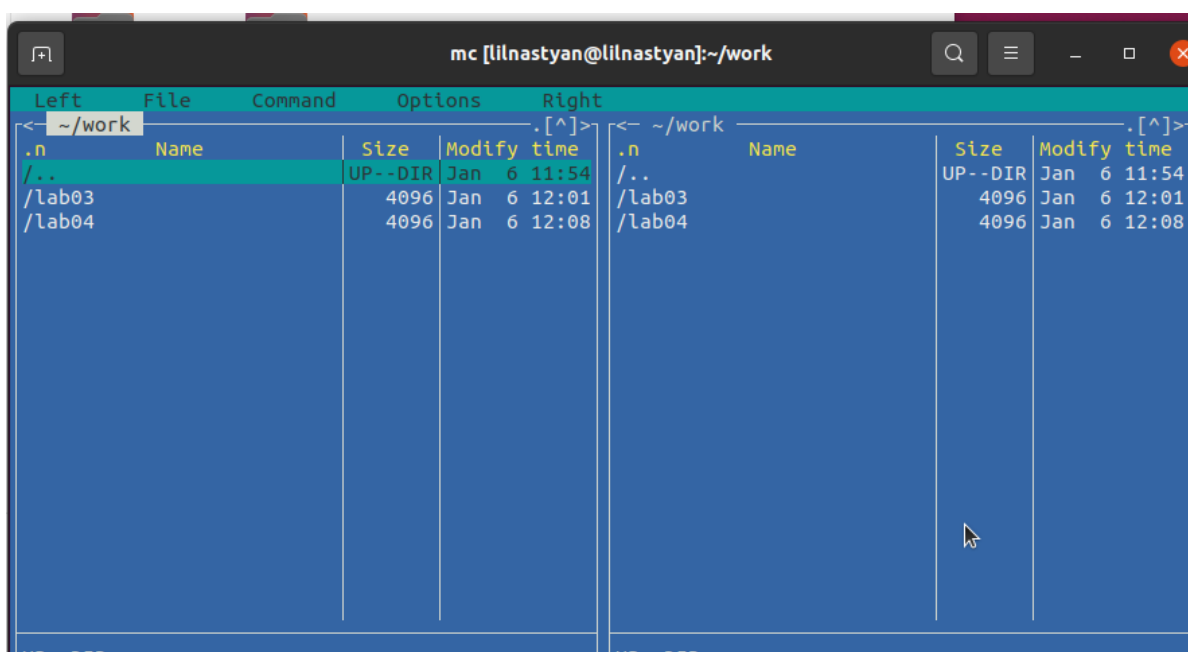


Рис. 4.1: Запуск Midnight Commander

При помощи touch создала файл lab05-1.asm



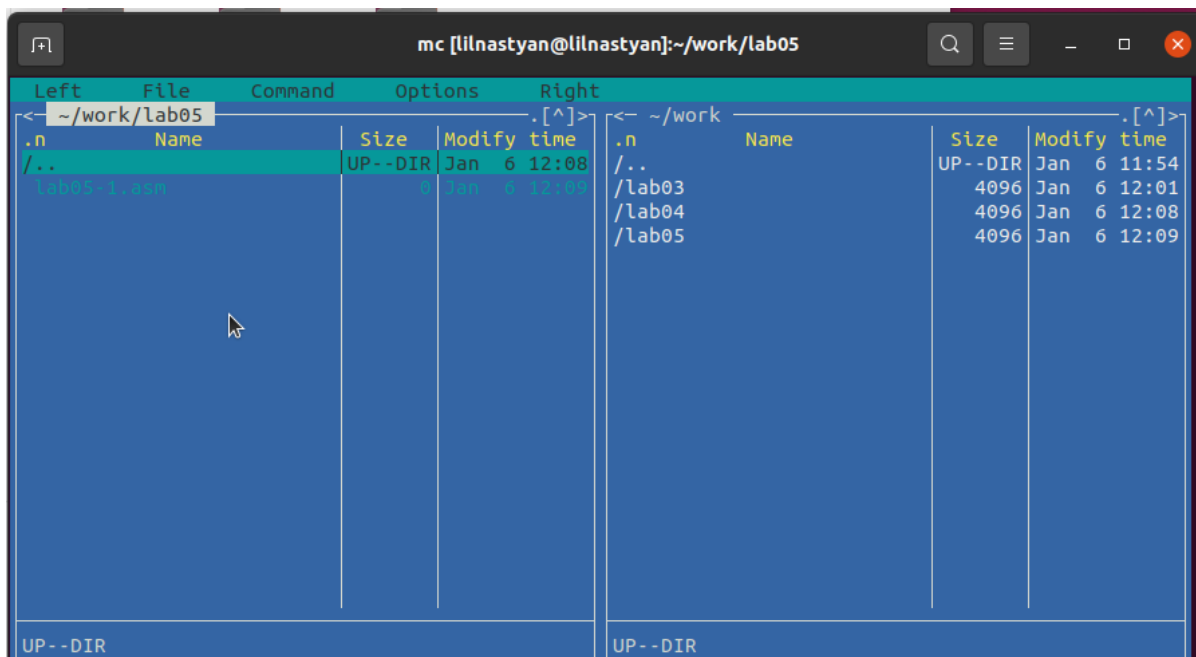
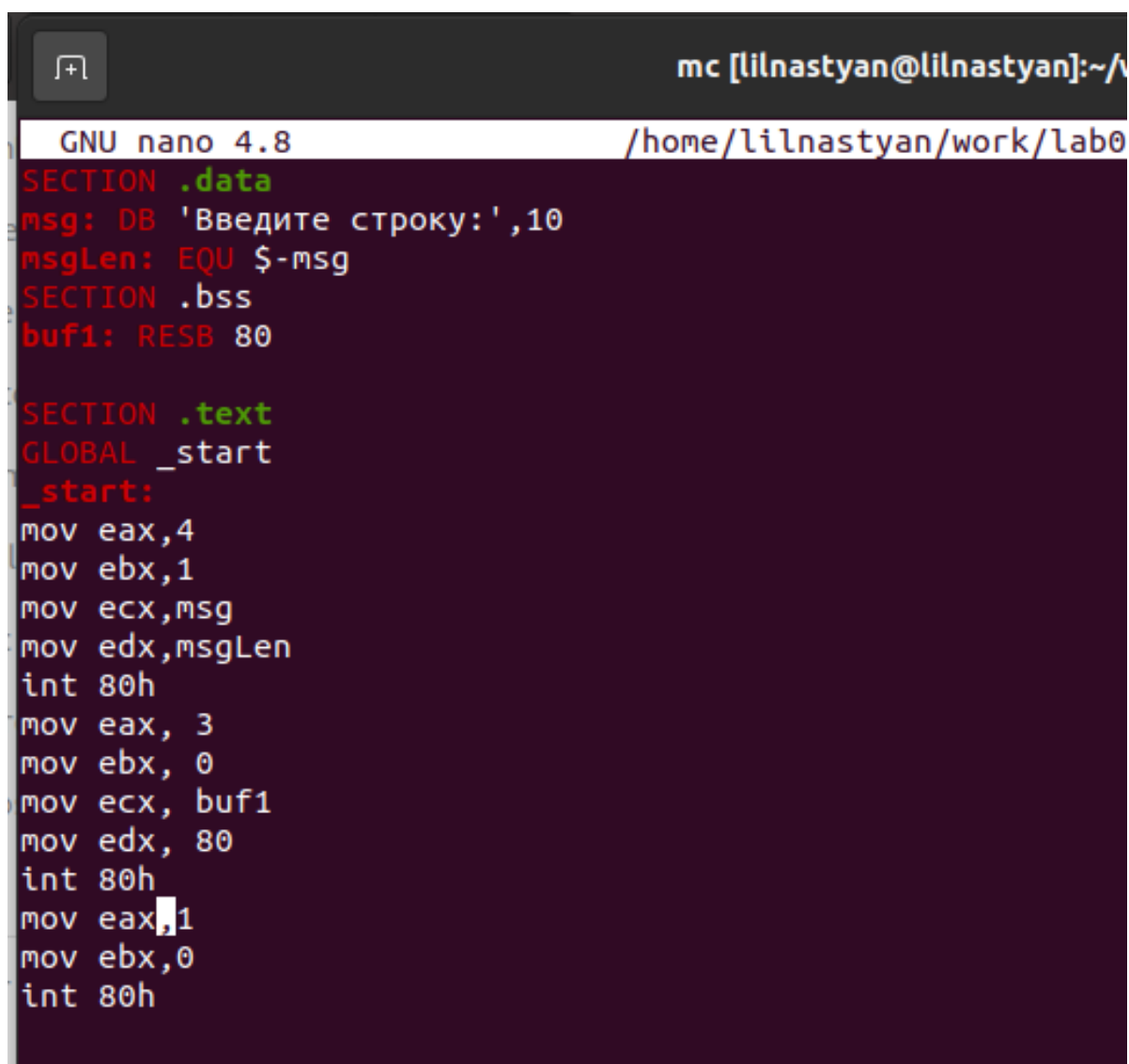


Рис. 4.2: Создание файла lab05-1.asm

Открыла файл на редактирование клавишей F4, выбрала редактор mceditor, написала код программы из задания.

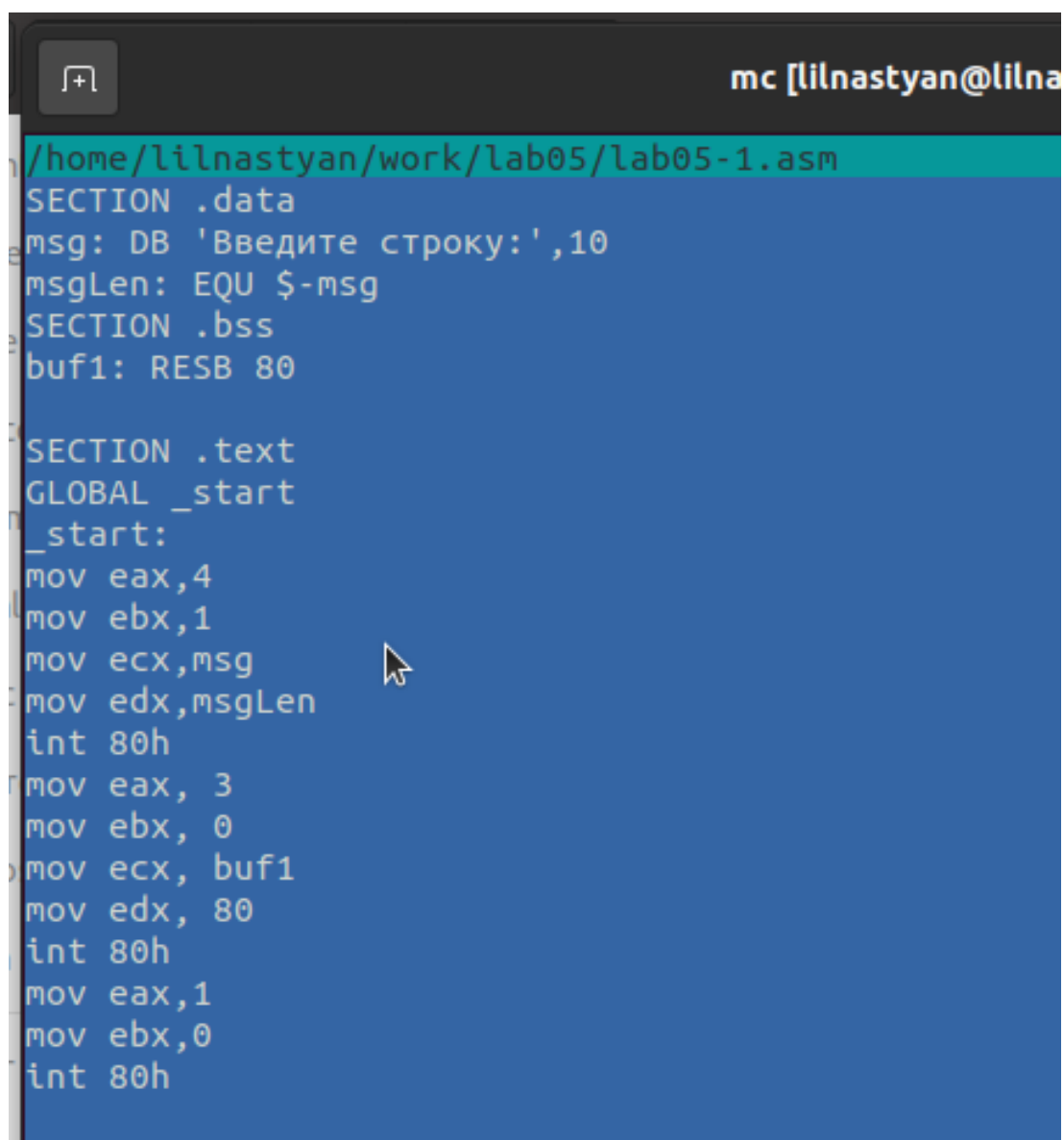
A screenshot of a terminal window with a dark background. At the top, a status bar shows 'mc [lilnastyan@lilnastyan]:~/v'. Below it, the terminal title is 'GNU nano 4.8 /home/lilnastyan/work/lab0'. The main area contains assembly code for a program. The code is organized into sections: .data, .bss, and .text. The .data section defines a message 'Введите строку:' and its length. The .bss section reserves space for a buffer. The .text section contains the main logic, starting with a global \_start label. The code uses x86-64 instructions like mov, int, and equ to set up registers and perform system calls (int 80h) for reading and writing. The code is as follows:

```
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. 4.3: Программа в файле lab05-1.asm

Открыла файл на просмотр клавишей F3 и проверила, что он содержит набранный код.



```
mc [lilnastyan@lilna
/home/lilnastyan/work/lab05/lab05-1.asm
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. 4.4: Просмотр файла lab05-1.asm

Транслировала файл программы в объектный файл, выполнила компоновку объектного файла, получила исполняемый файл программы и проверила ее работу.

```
lilnastyan@lilnastyan:~/work/lab05$ nasm -f elf lab05-1.asm
lilnastyan@lilnastyan:~/work/lab05$ ld -m elf_i386 lab05-1.o -o lab05-1
lilnastyan@lilnastyan:~/work/lab05$ ./lab05-1
Введите строку:
Anastasiya
lilnastyan@lilnastyan:~/work/lab05$
```

Рис. 4.5: Запуск программы lab05-1.asm

Для упрощения написания программ часто встречающиеся одинаковые участки кода (такие как, например, вывод строки на экран или выход из программы) можно оформить в виде подпрограмм и сохранить в отдельные файлы, а во всех нужных местах поставить вызов нужной подпрограммы. Это позволяет сделать основную программу более удобной для написания и чтения.

Скачала файл in\_out.asm и разместила его в рабочем каталоге. Для копирования используется клавиша F5. Для перемещения используется клавиша F6.

Скопировала lab05-1.asm в lab05-2.asm.

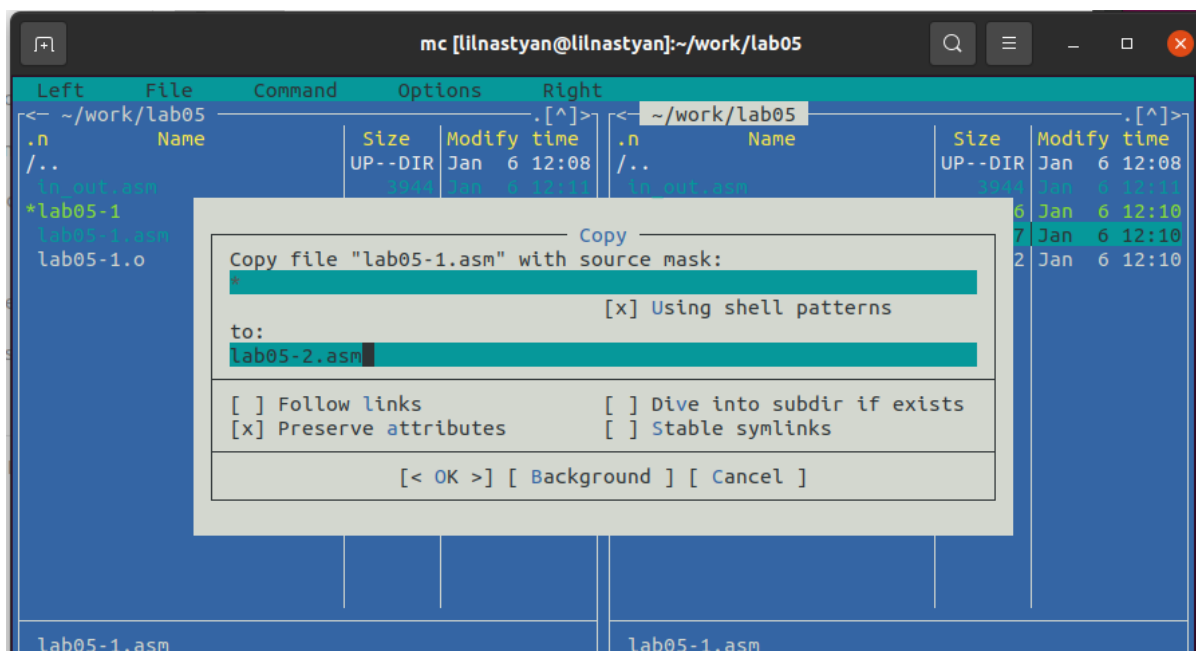
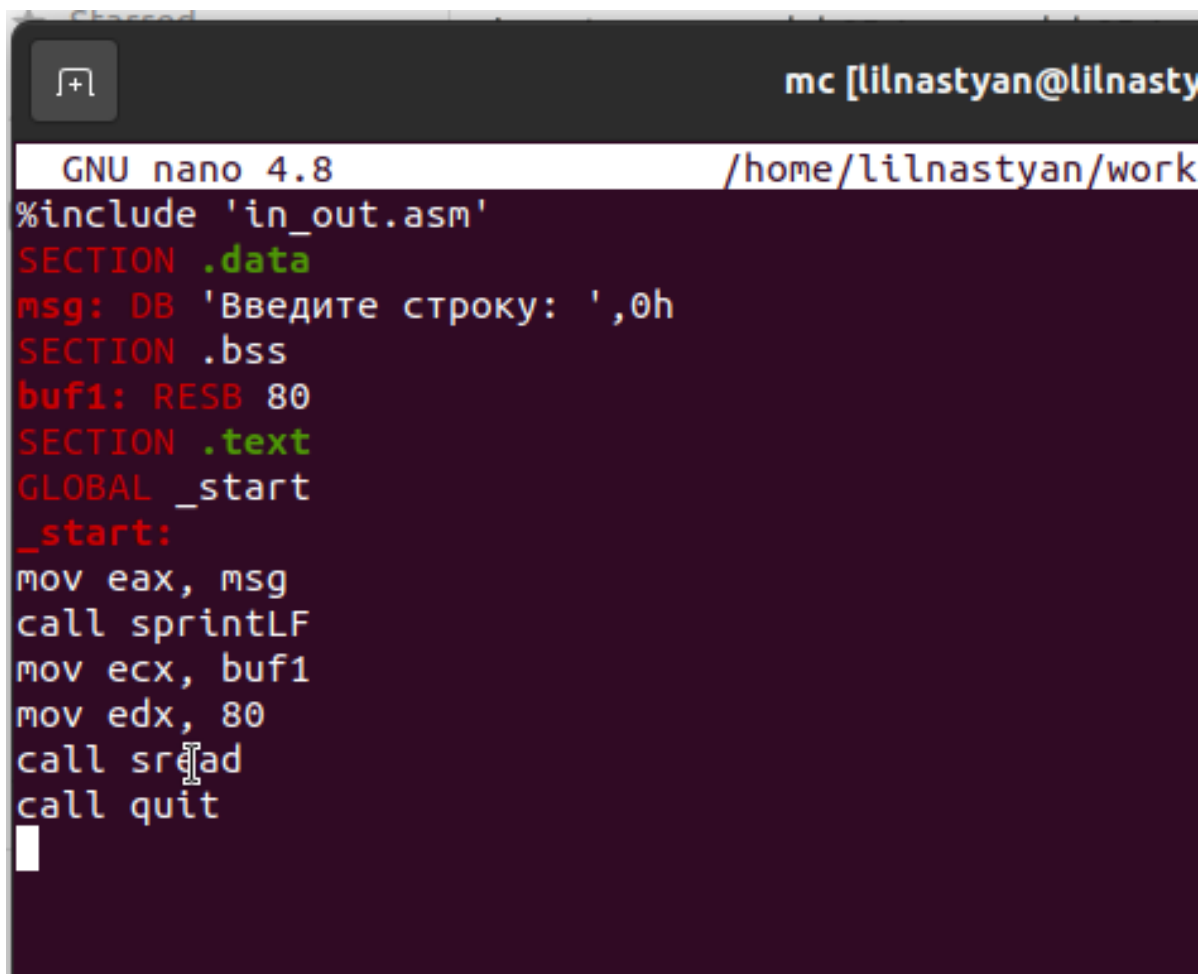


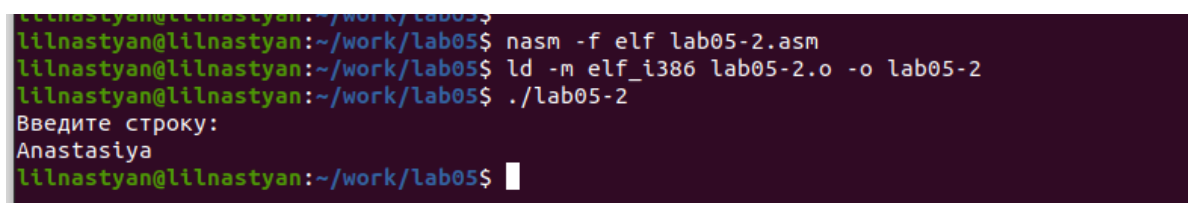
Рис. 4.6: Копирование файла lab05-1.asm

Написала код программы lab05-2.asm с использованием подпрограмм из внешнего файла in\_out.asm.

A screenshot of a terminal window with a dark background. The title bar shows 'mc [lilnastyan@lilnasty...'. The terminal content shows the GNU nano 4.8 editor at /home/lilnastyan/work. The code includes 'in\_out.asm', defines a data section with a message, a bss section with a buffer, and a text section with a \_start label. The \_start routine calls sprintf to format the message into the buffer, then calls read to get input, and finally calls quit.

```
GNU nano 4.8 /home/lilnastyan/work
%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 read
call quit
```

Рис. 4.7: Программа в файле lab05-2.asm

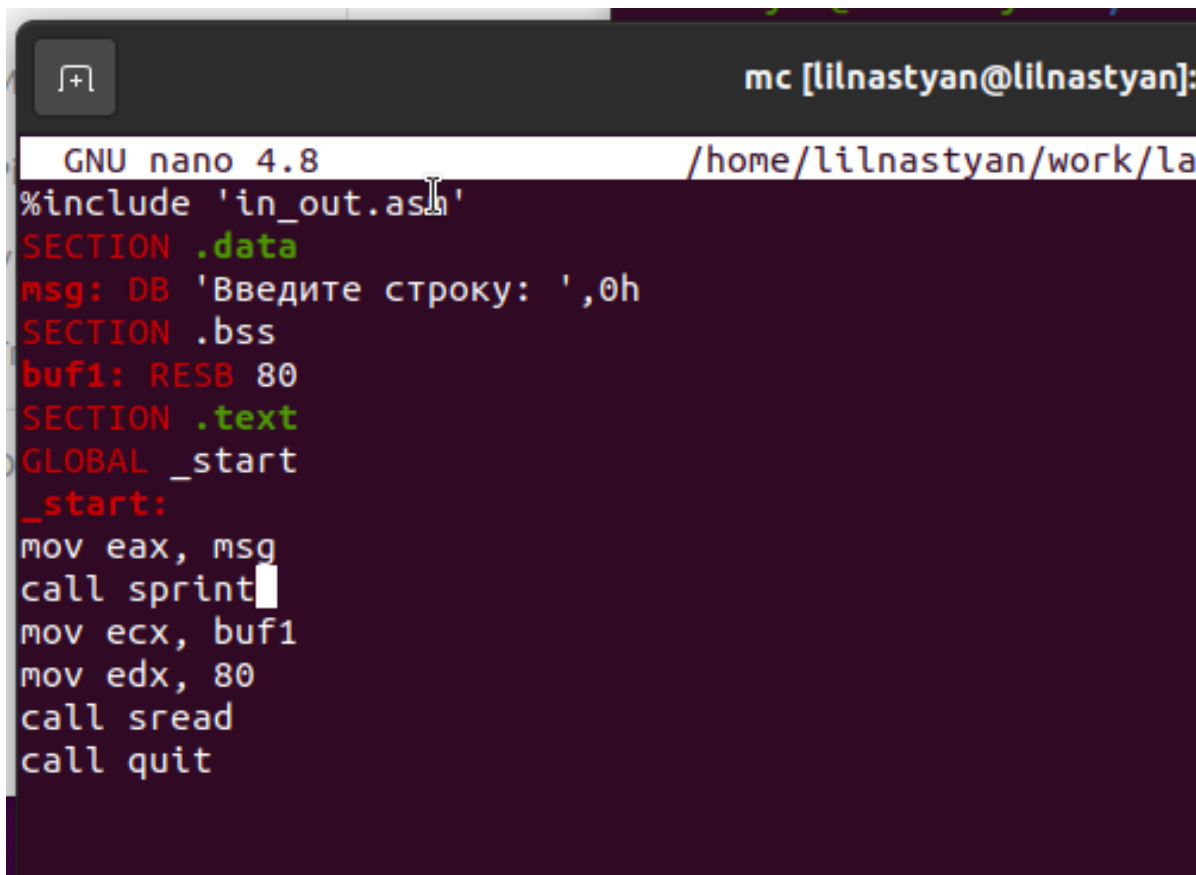
A screenshot of a terminal window showing the compilation and execution of the assembly program. The user runs nasm to compile lab05-2.asm into lab05-2.o, then ld to link it into an executable lab05-2. The program is then run, prompting for a string, which is entered as 'Anastasiya'.

```
lilnastyan@lilnastyan:~/work/lab05$ nasm -f elf lab05-2.asm
lilnastyan@lilnastyan:~/work/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
lilnastyan@lilnastyan:~/work/lab05$ ./lab05-2
Введите строку:
Anastasiya
lilnastyan@lilnastyan:~/work/lab05$
```

Рис. 4.8: Запуск программы lab05-2.asm

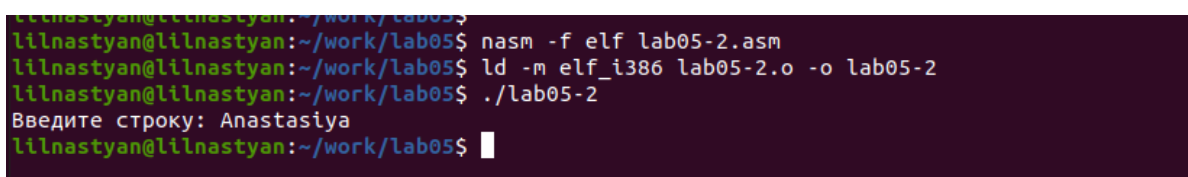
В файле lab5-2.asm заменила подпрограмму sprintf на sprintf. Заново собрала

исполняемый файл. Теперь после вывода строки она не завершается символом перехода на новую строку.



```
mc [lilnastyan@lilnastyan]:  
GNU nano 4.8 /home/lilnastyan/work/la  
%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. 4.9: Программа в файле lab05-2.asm



```
lilnastyan@lilnastyan:~/work/lab05$  
lilnastyan@lilnastyan:~/work/lab05$ nasm -f elf lab05-2.asm  
lilnastyan@lilnastyan:~/work/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2  
lilnastyan@lilnastyan:~/work/lab05$ ./lab05-2  
Введите строку: Anastasiya  
lilnastyan@lilnastyan:~/work/lab05$
```

Рис. 4.10: Запуск программы lab05-2.asm

Скопировала программу lab05-1.asm и изменила код, так чтобы она работала по следующему алгоритму:

- вывести приглашение типа “Введите строку:”;

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



```
mc [lilnastya@
GNU nano 4.8 /home/lilnastya
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
```

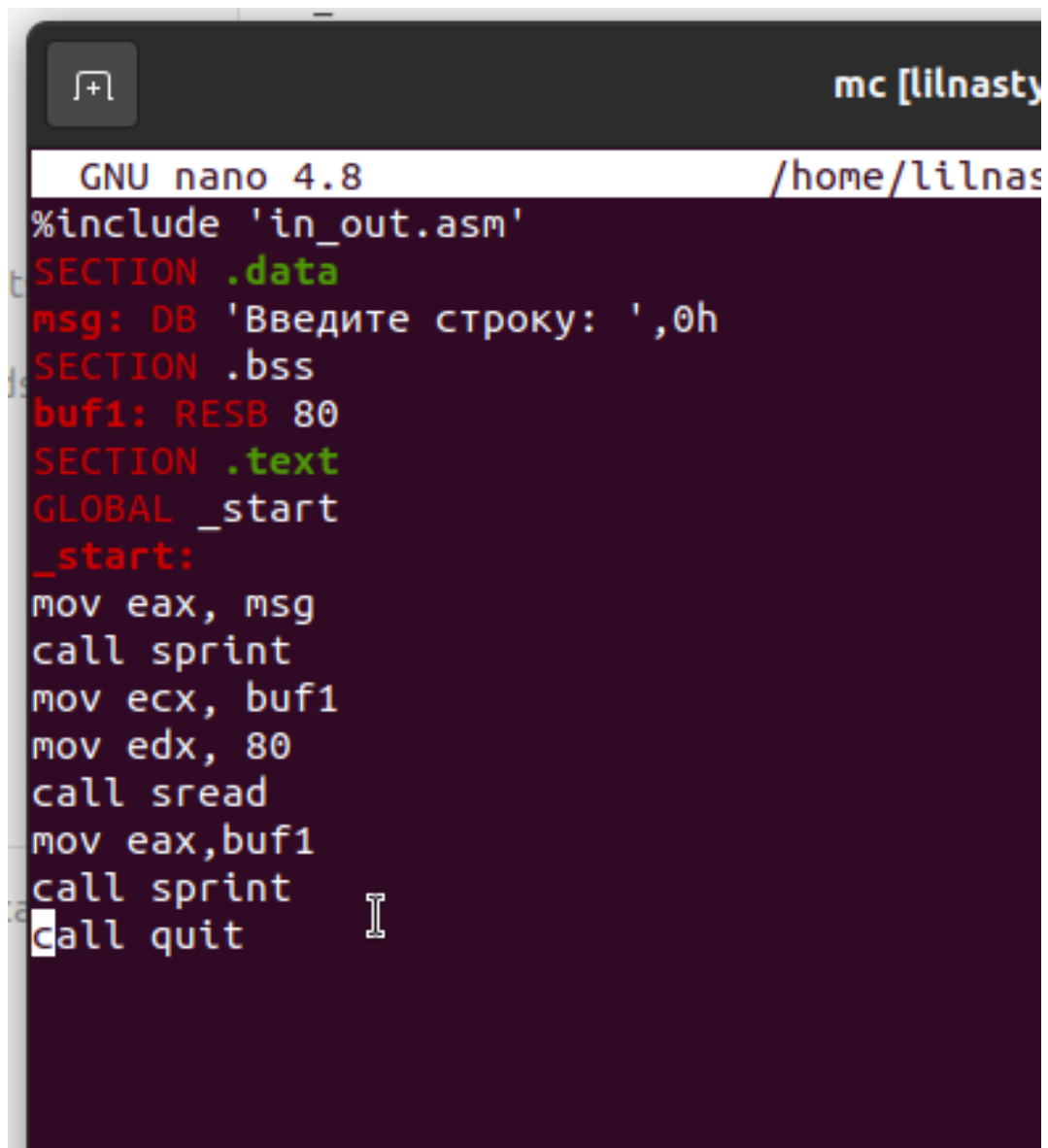
Рис. 4.11: Программа в файле lab05-3.asm

```
lilnastyan@lilnastyan:~/work/lab05$ nasm -f elf lab05-3.asm
lilnastyan@lilnastyan:~/work/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3
lilnastyan@lilnastyan:~/work/lab05$ ./lab05-3
Введите строку:
Anastasiya
Anastasiya
lilnastyan@lilnastyan:~/work/lab05$
```

Рис. 4.12: Запуск программы lab05-3.asm

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

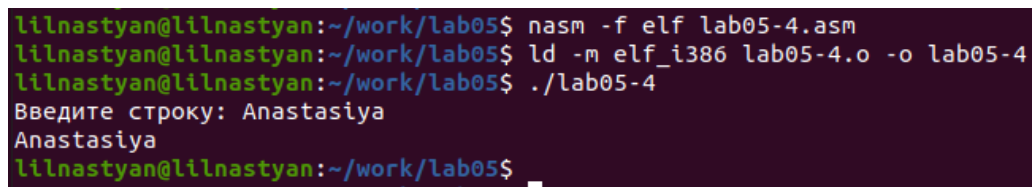




The screenshot shows a terminal window with a dark background. At the top, there's a title bar with a window icon and the text "mc [lilnasty". Below it, the terminal shows the GNU nano 4.8 editor interface. The file path is "/home/lilnasty". The code being edited is assembly code for a program that prompts the user to enter a string. The code includes a data section with a message, a bss section with a buffer, and a text section with the main logic. The cursor is at the end of the "call quit" instruction.

```
GNU nano 4.8 /home/lilnasty
%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. 4.13: Программа в файле lab05-4.asm



The screenshot shows a terminal window with a dark background. The user is in the directory ~/work/lab05. They run the following commands: "nasm -f elf lab05-4.asm", "ld -m elf\_i386 lab05-4.o -o lab05-4", and "./lab05-4". The program prompts the user to enter a string, and the user enters "Anastasiya".

```
lilnastyan@lilnastyan:~/work/lab05$ nasm -f elf lab05-4.asm
lilnastyan@lilnastyan:~/work/lab05$ ld -m elf_i386 lab05-4.o -o lab05-4
lilnastyan@lilnastyan:~/work/lab05$ ./lab05-4
Введите строку: Anastasiya
Anastasiya
lilnastyan@lilnastyan:~/work/lab05$
```

Рис. 4.14: Запуск программы lab05-4.asm

## 5 Выводы

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