

# **Отчёт по лабораторной работе 5**

**Архитектура компьютеров**

Амрахов Азер Араз оглы

# 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>16</b>

## Список иллюстраций

2.1	Создание каталога . . . . .	6
2.2	Создание файла lab05-1.asm . . . . .	7
2.3	Программа в файле lab05-1.asm . . . . .	8
2.4	Просмотр файла lab05-1.asm . . . . .	9
2.5	Запуск программы lab05-1.asm . . . . .	9
2.6	Копирование файла . . . . .	10
2.7	Программа в файле lab05-2.asm . . . . .	11
2.8	Запуск программы lab05-2.asm . . . . .	11
2.9	Программа в файле lab05-2.asm . . . . .	12
2.10	Запуск программы lab05-2.asm . . . . .	12
2.11	Программа в файле lab05-3.asm . . . . .	13
2.12	Запуск программы lab05-3.asm . . . . .	13
2.13	Программа в файле lab05-4.asm . . . . .	14
2.14	Запуск программы lab05-4.asm . . . . .	15

## **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

Открыл Midnight Commander. Перешел в каталог ~/work/arch-pc. Создал каталог lab05 (рис. 2.1)

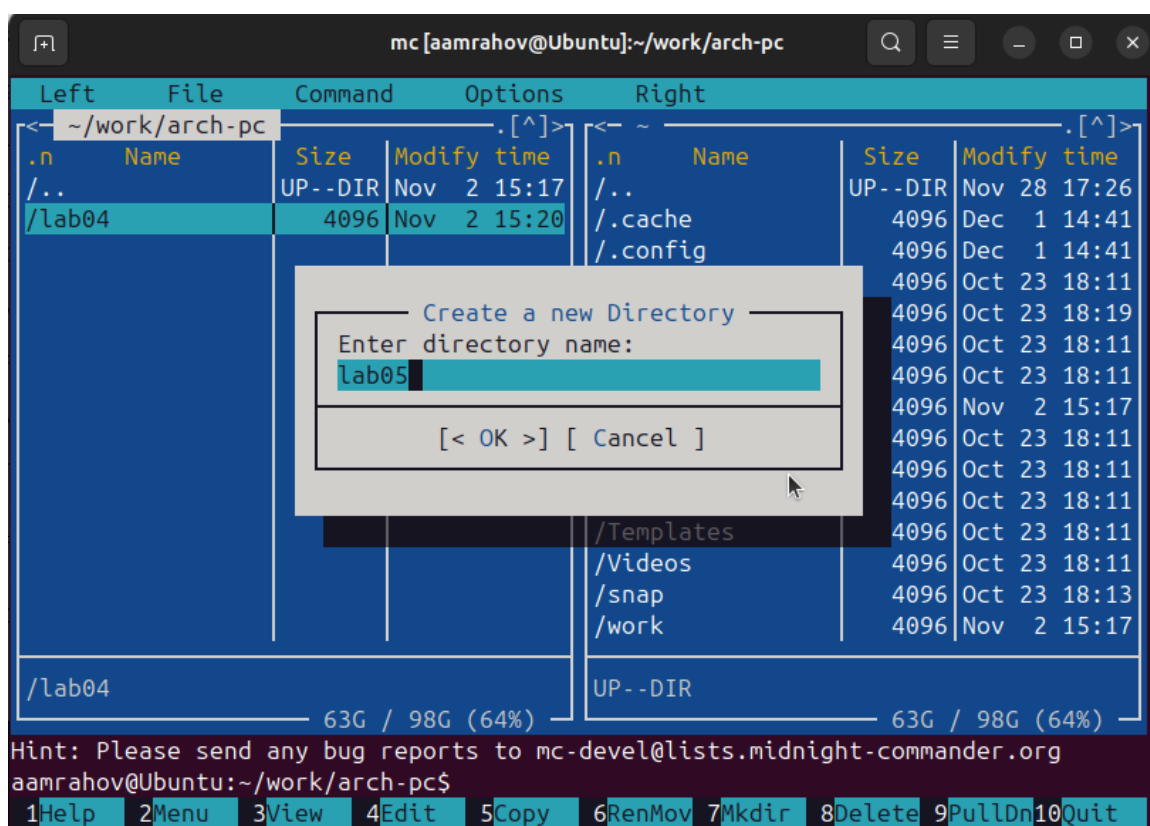


Рисунок 2.1: Создание каталога

Создал файл lab05-1.asm (рис. 2.2)

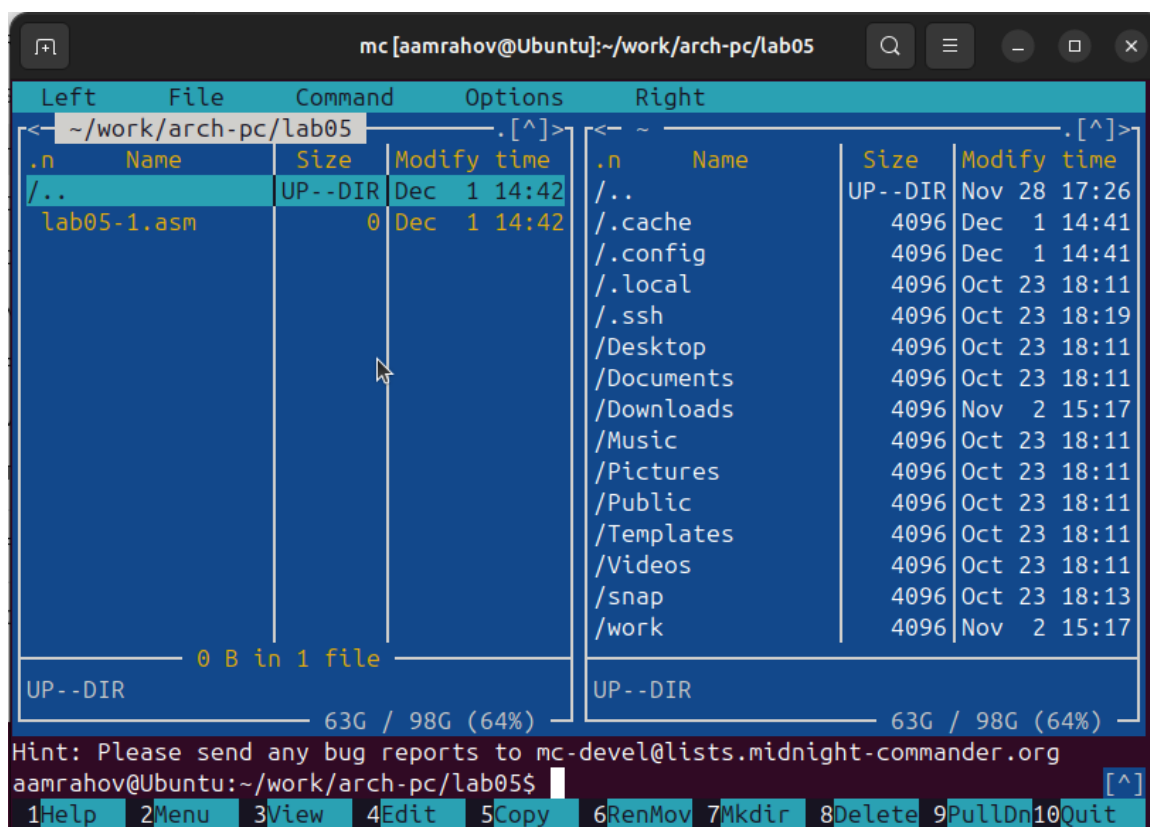
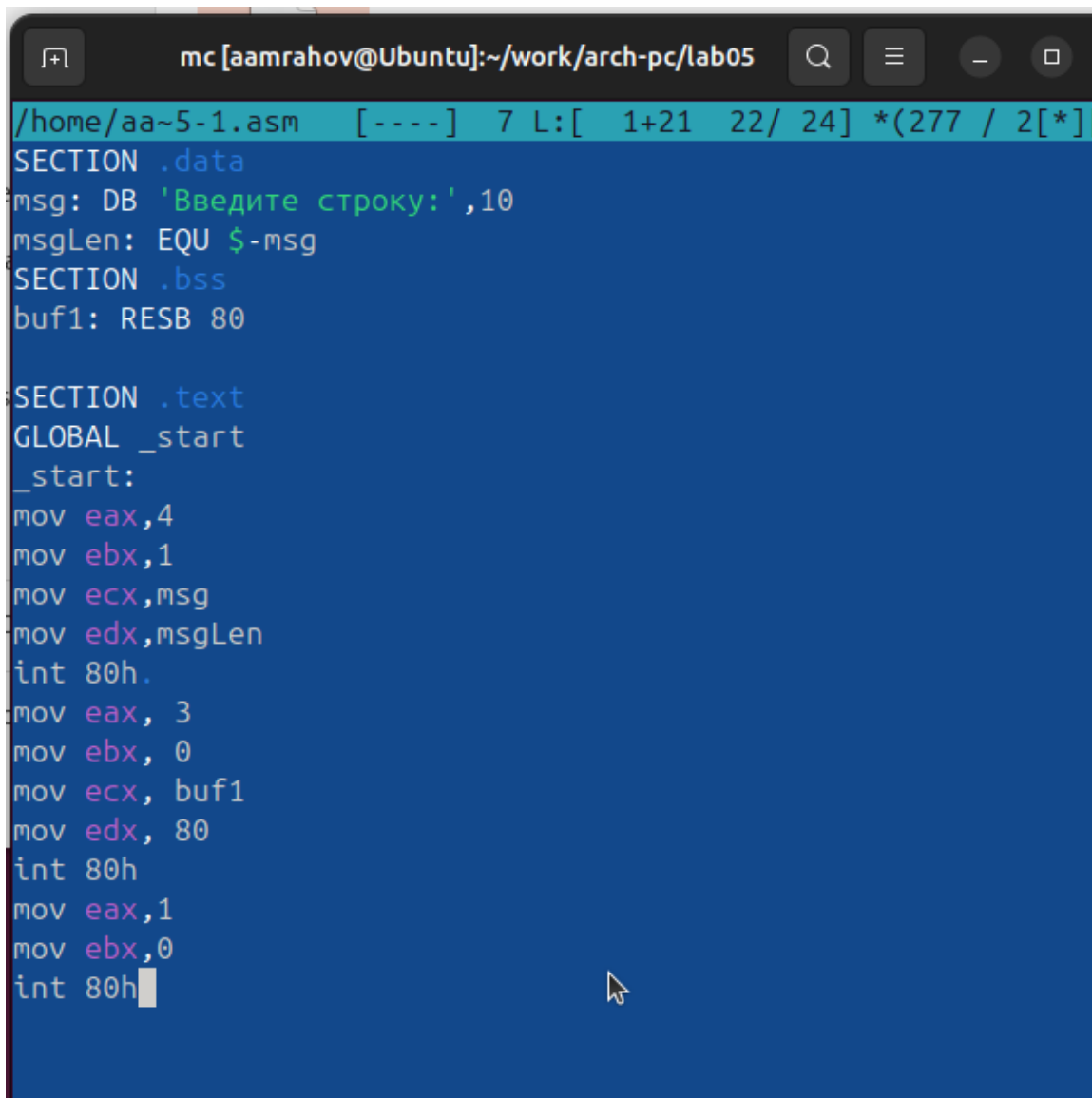


Рисунок 2.2: Создание файла lab05-1.asm

Открыл файл на редактирование. Написал код. (рис. 2.3)



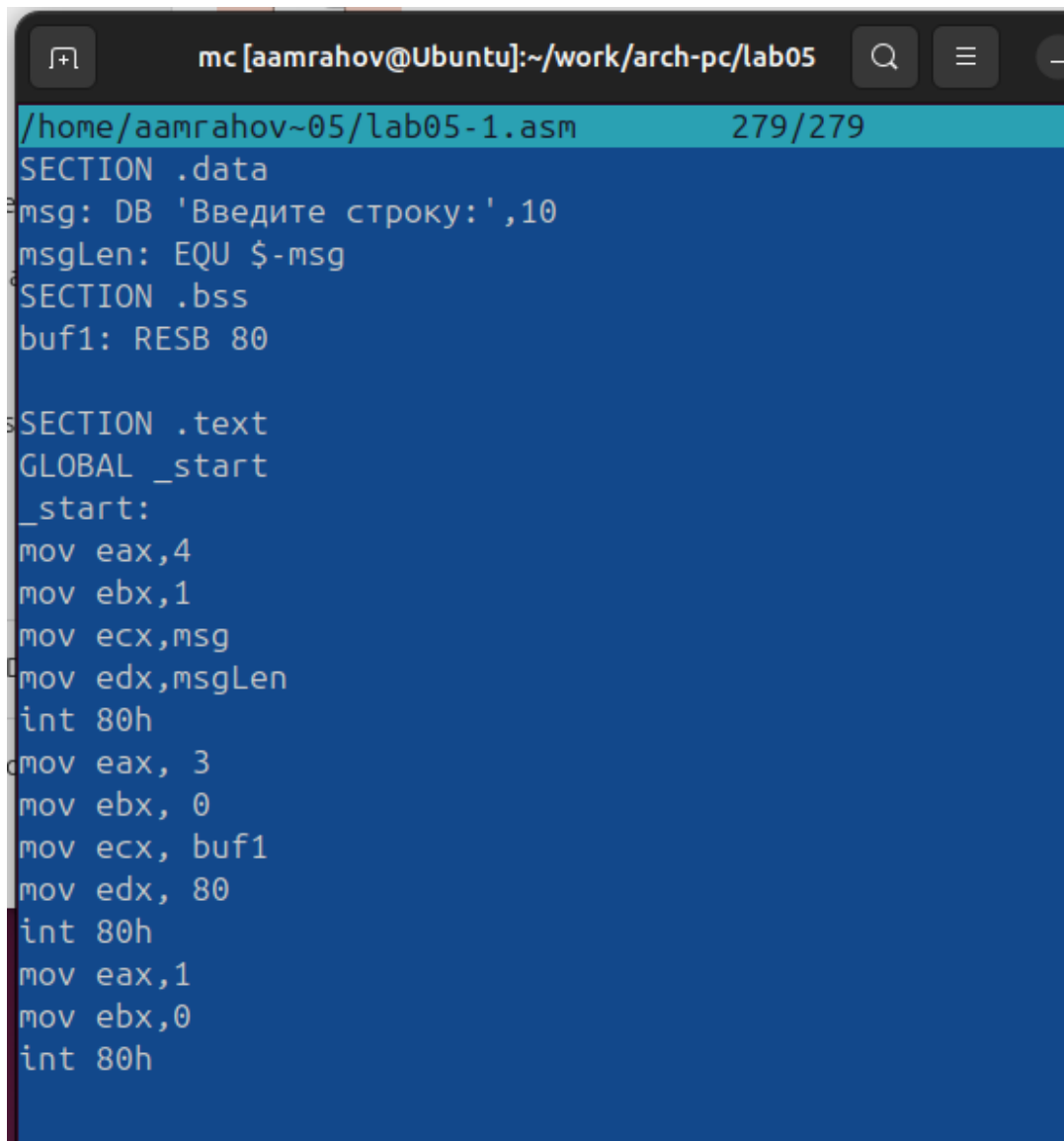
```
mc [aamrahov@Ubuntu]:~/work/arch-pc/lab05
/home/aa~5-1.asm  [----]  7 L:[ 1+21 22/ 24] *(277 / 2[*])
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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
```

Рисунок 2.3: Программа в файле lab05-1.asm

Открыл файл для просмотра и убедился, что он содержит написанный код.  
(рис. 2.4)



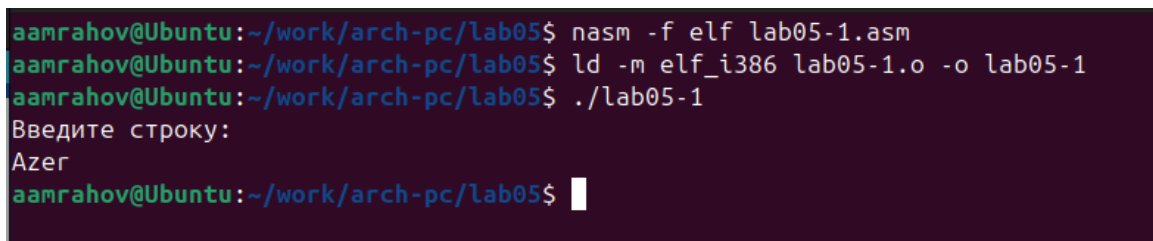


```
mc [aamrahov@Ubuntu]:~/work/arch-pc/lab05 279/279
/home/aamrahov~05/lab05-1.asm
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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
```

Рисунок 2.4: Просмотр файла lab05-1.asm

Получил исполняемый файл программы и проверил его работу.(рис. 2.5)



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

Рисунок 2.5: Запуск программы lab05-1.asm

Скачал файл in\_out.asm. Добавил файл in\_out.asm в рабочий каталог. Скопировал lab05-1.asm в lab05-2.asm. (рис. 2.6)

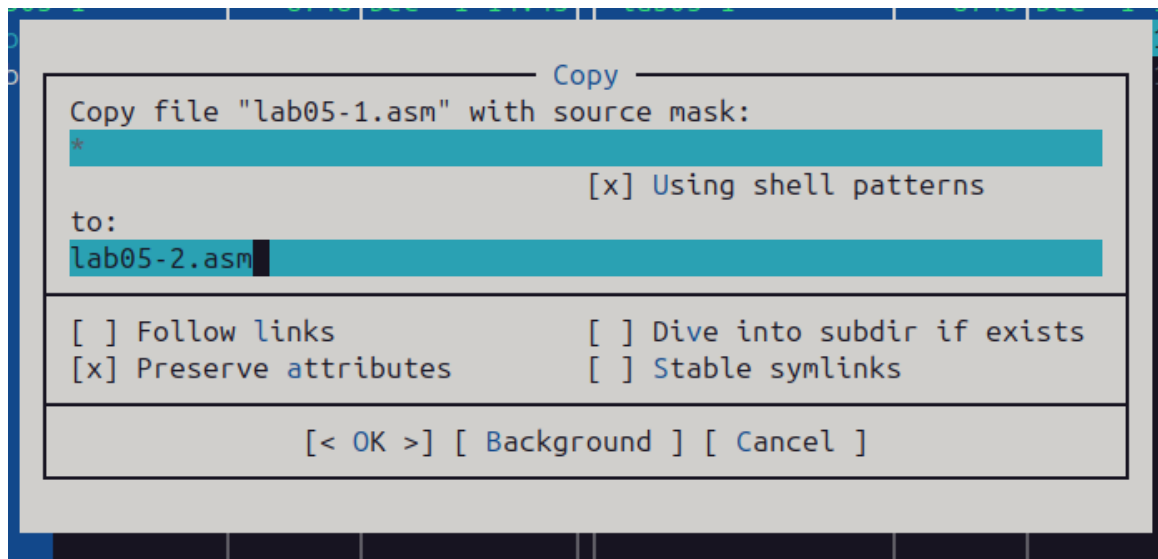
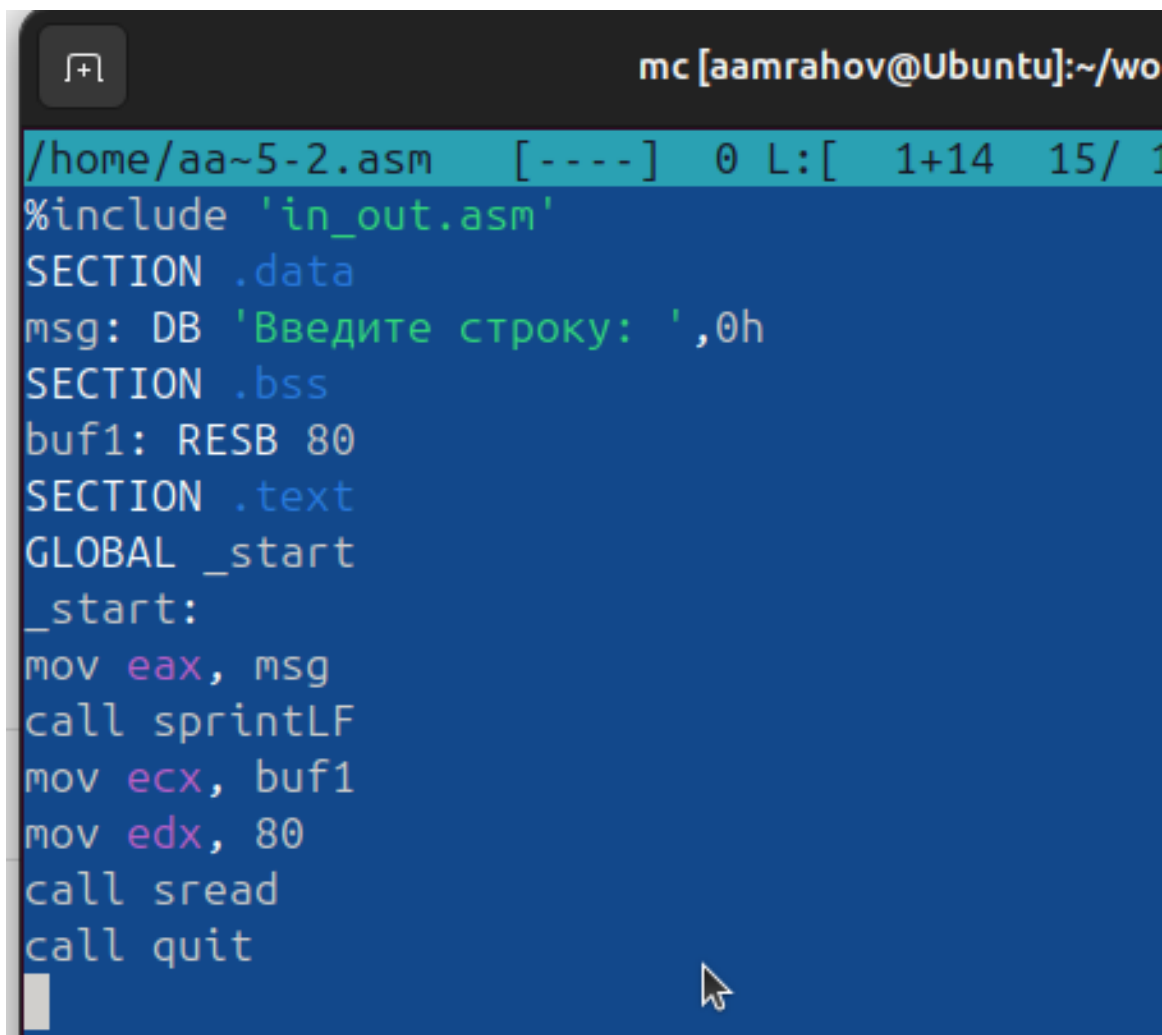


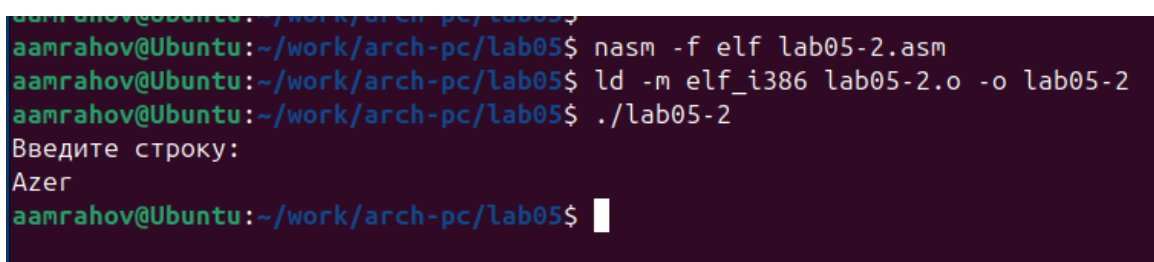
Рисунок 2.6: Копирование файла

Написал код программы lab05-2.asm. (рис. 2.7) Скомпилировал программу и проверил запуск. (рис. 2.8)

A screenshot of a code editor window titled 'mc [aamrahov@Ubuntu]:~/wo'. The editor shows the contents of a file named '/home/aa~5-2.asm'. The code is written in assembly language and includes sections for data, bss, and text. It defines a message 'Введите строку: ',0h, reserves space for a buffer, and contains a main routine that prints the message and reads input from the user. The code is as follows:

```
/home/aa~5-2.asm [----] 0 L:[ 1+14 15/ 1
%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
```

Рисунок 2.7: Программа в файле lab05-2.asm

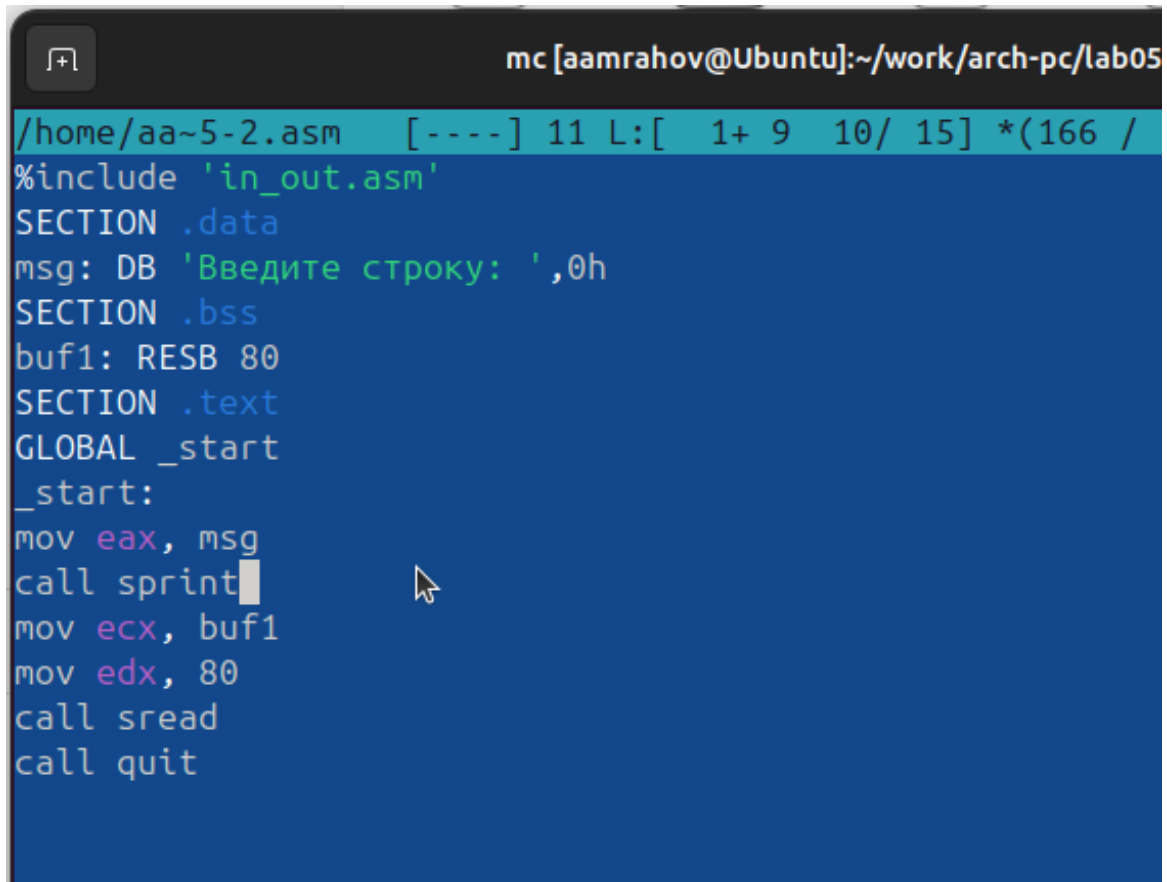
A screenshot of a terminal window showing the compilation and execution of the assembly program. The user runs 'nasm -f elf lab05-2.asm' to compile the assembly file into an object file 'lab05-2.o'. Then, they run 'ld -m elf\_i386 lab05-2.o -o lab05-2' to link the object file into an executable file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

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

Рисунок 2.8: Запуск программы lab05-2.asm

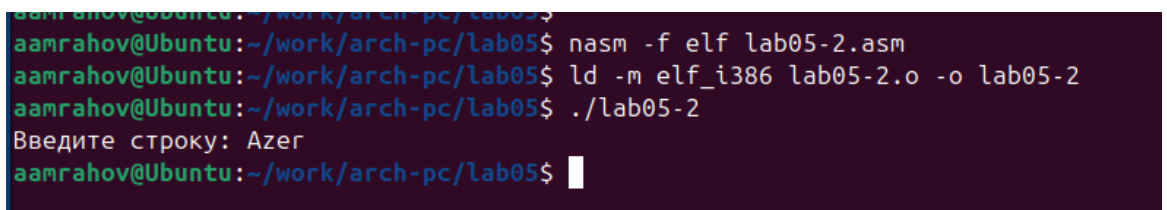
В файле lab5-2.asm я заменил подпрограмму sprintLF на sprint (рис. 2.9). Затем я снова собрал исполняемый файл (рис. 2.10). Теперь после вывода строки

она не завершается символом перехода на новую строку.



```
mc [aamrahov@Ubuntu]:~/work/arch-pc/lab05
/home/aa~5-2.asm [----] 11 L:[ 1+ 9 10/ 15] *(166 /
#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
```

Рисунок 2.9: Программа в файле lab05-2.asm



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

Рисунок 2.10: Запуск программы lab05-2.asm

Скопировал программу lab05-1.asm и изменил код, чтобы программа выводила приглашение типа «Введите строку», затем считывала строку с клавиатуры и выводила введенную строку на экран. (рис. 2.11, рис. 2.12)

```
mc [aamrahov@Ubuntu]:~/work/arch
/home/aa~5-3.asm [----] 9 L:[ 1+19 20/ 28] *
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

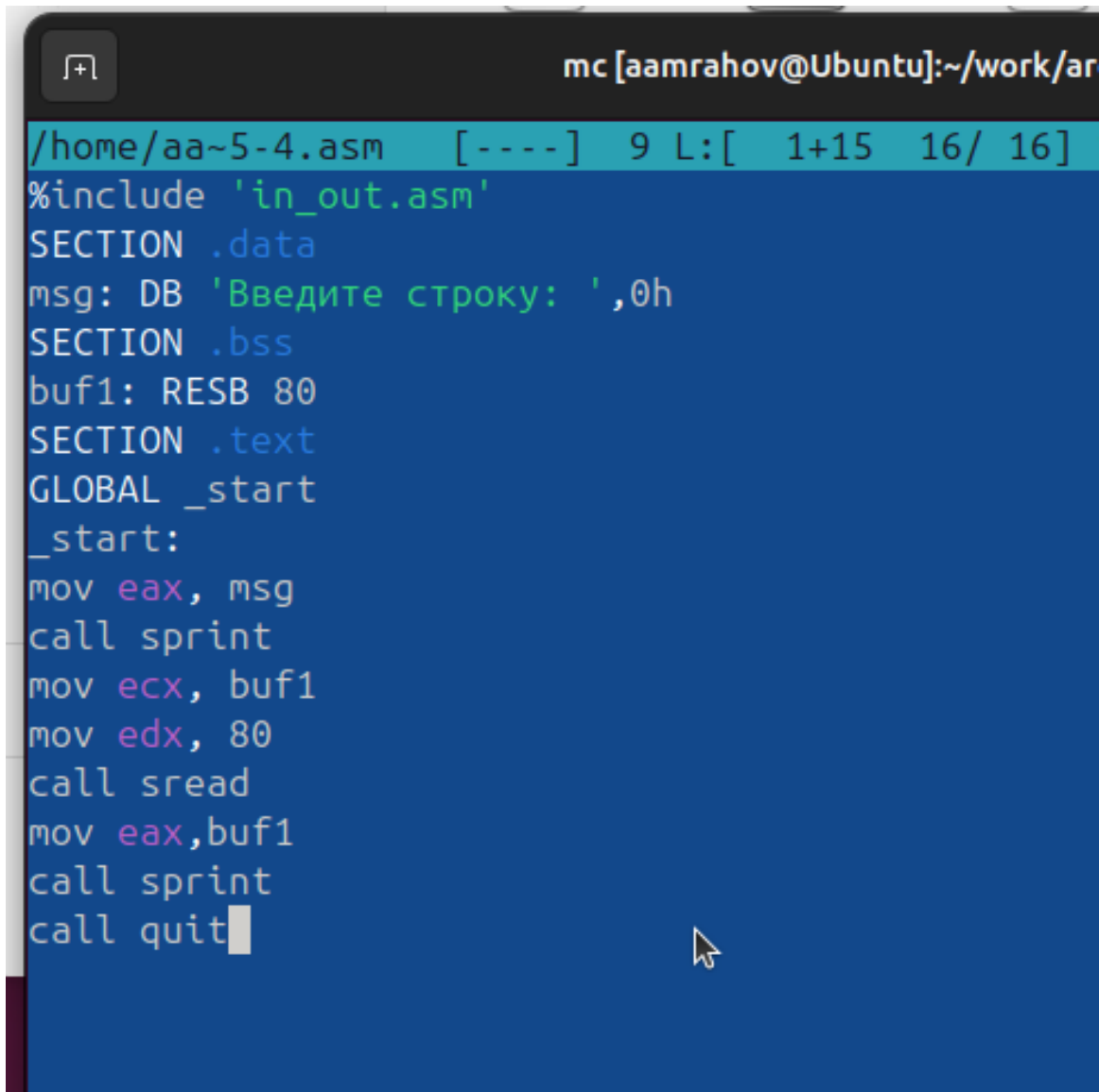
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h.
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h.
mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,80
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рисунок 2.11: Программа в файле lab05-3.asm

```
aamrahov@Ubuntu:~/work/arch-pc/lab05$
aamrahov@Ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab05-3.asm
aamrahov@Ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3
aamrahov@Ubuntu:~/work/arch-pc/lab05$ ./lab05-3
Введите строку:
Azer
Azer
aamrahov@Ubuntu:~/work/arch-pc/lab05$
```

Рисунок 2.12: Запуск программы lab05-3.asm

Также я скопировал программу lab05-2.asm и внес соответствующие изменения в код, чтобы программа выводила приглашение типа «Введите строку:», затем считывала строку с клавиатуры и выводила введенную строку на экран.(рис. 2.13, рис. 2.14)



```
mc [aamrahov@Ubuntu]:~/work/ar
/home/aa~5-4.asm  [ - - - - ]  9  L:[  1+15  16/ 16]
%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
```

Рисунок 2.13: Программа в файле lab05-4.asm

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

Рисунок 2.14: Запуск программы lab05-4.asm

Отличие этих двух реализаций заключается в том, что файл `in_out.asm` содержит уже готовые подпрограммы для обеспечения ввода/вывода. Таким образом, нам остается только разместить данные в нужных регистрах и вызвать желаемую подпрограмму с помощью инструкции `call`.

## 3 Выводы

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