

# **Отчёт по лабораторной работе 5**

**Архитектура компьютеров**

Довлетов Арслан

# 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)

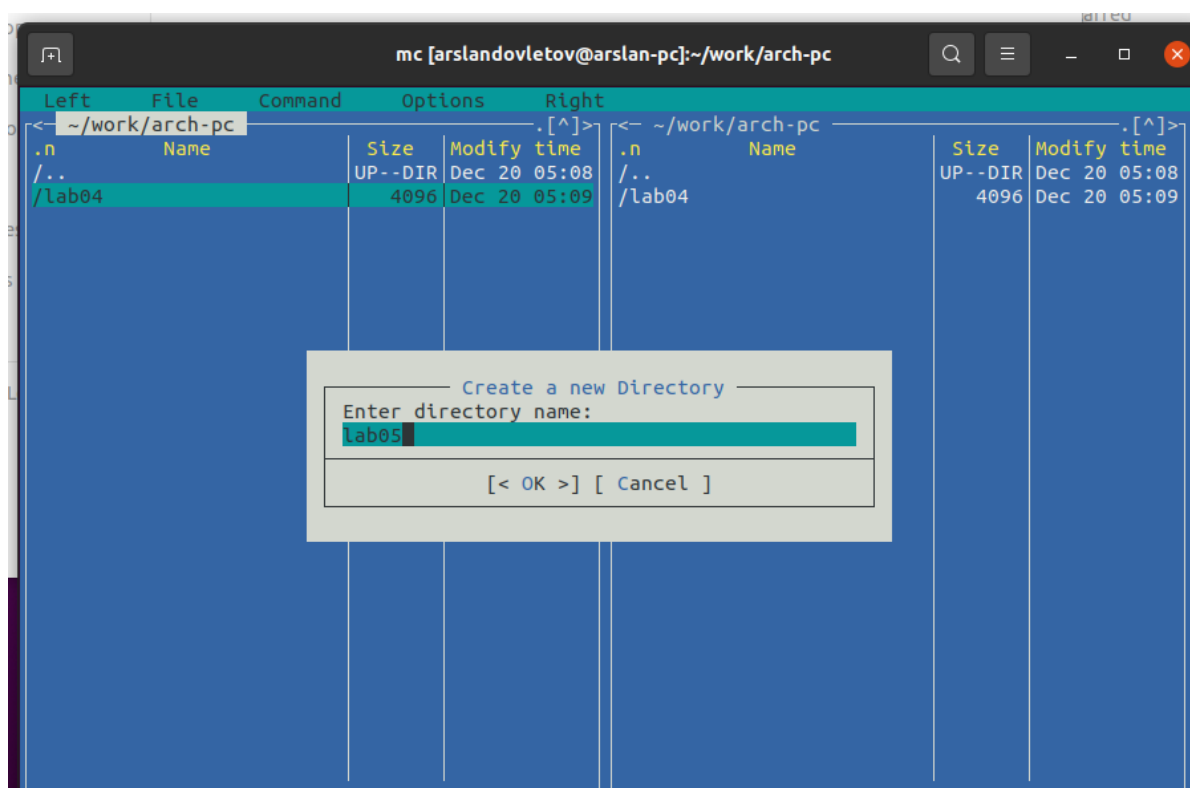


Рис. 2.1: Создание каталога

Создал файл lab05-1.asm (рис. 2.2)

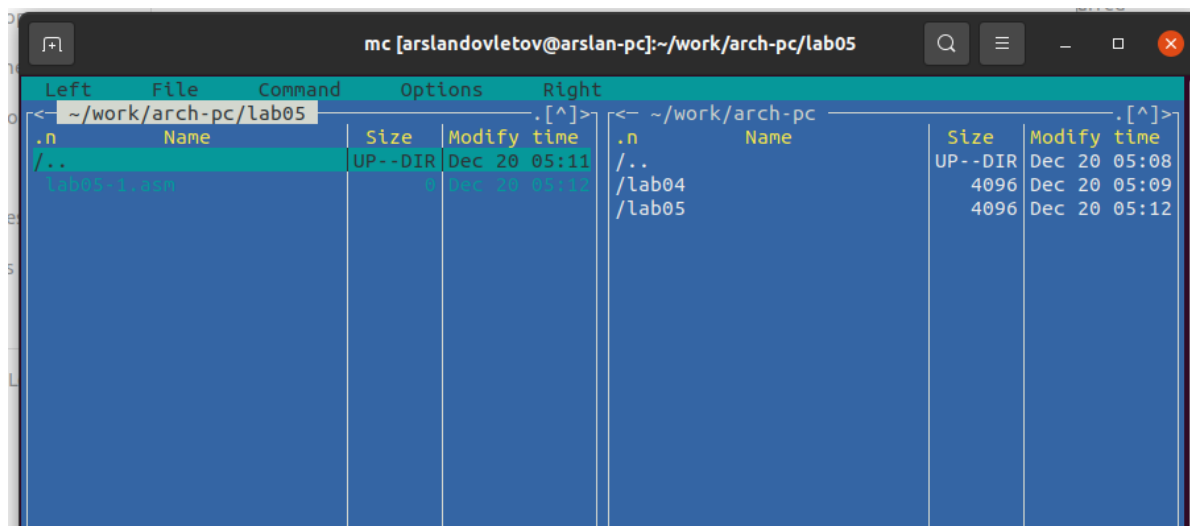
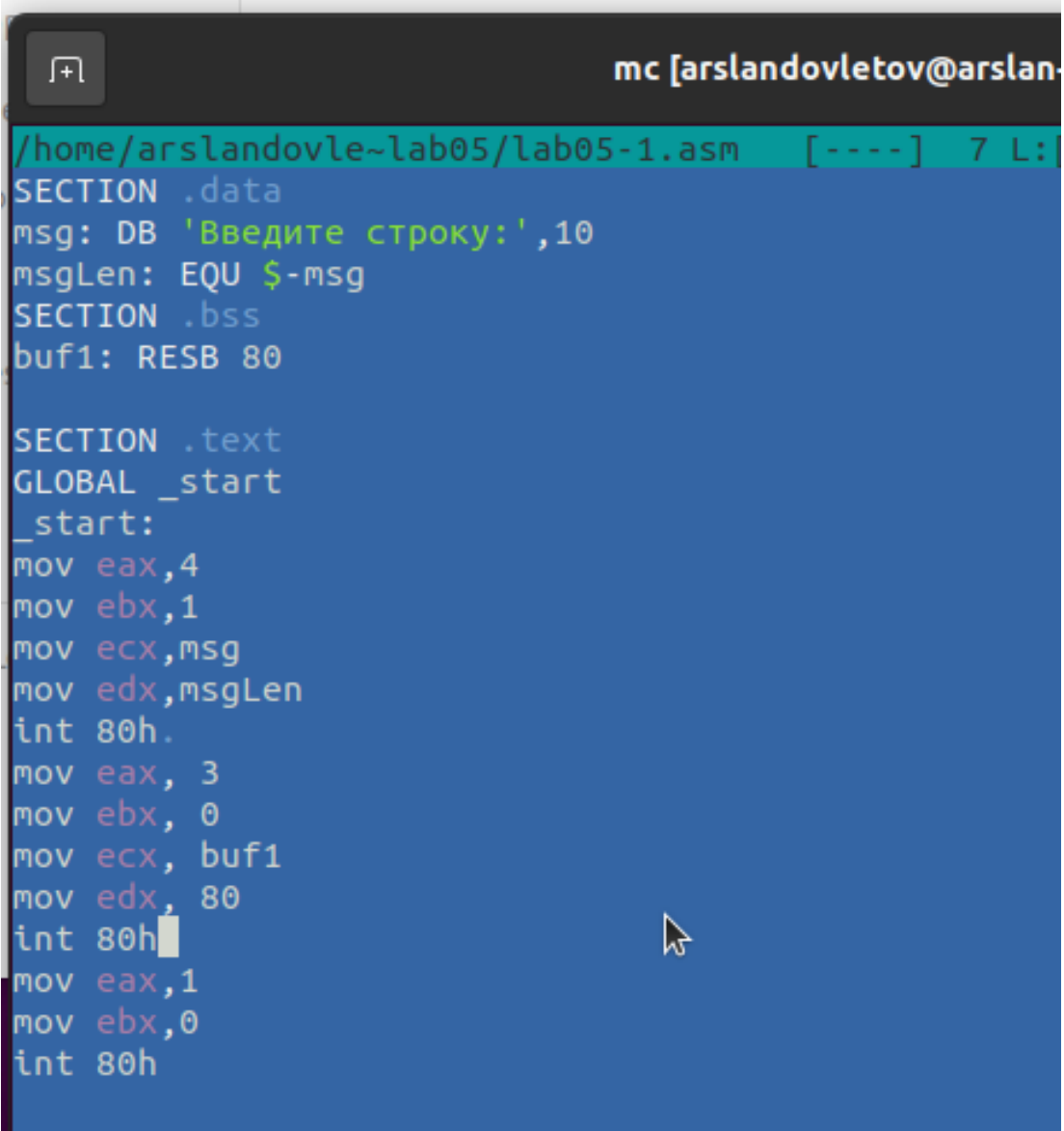


Рис. 2.2: Создание файла lab05-1.asm

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

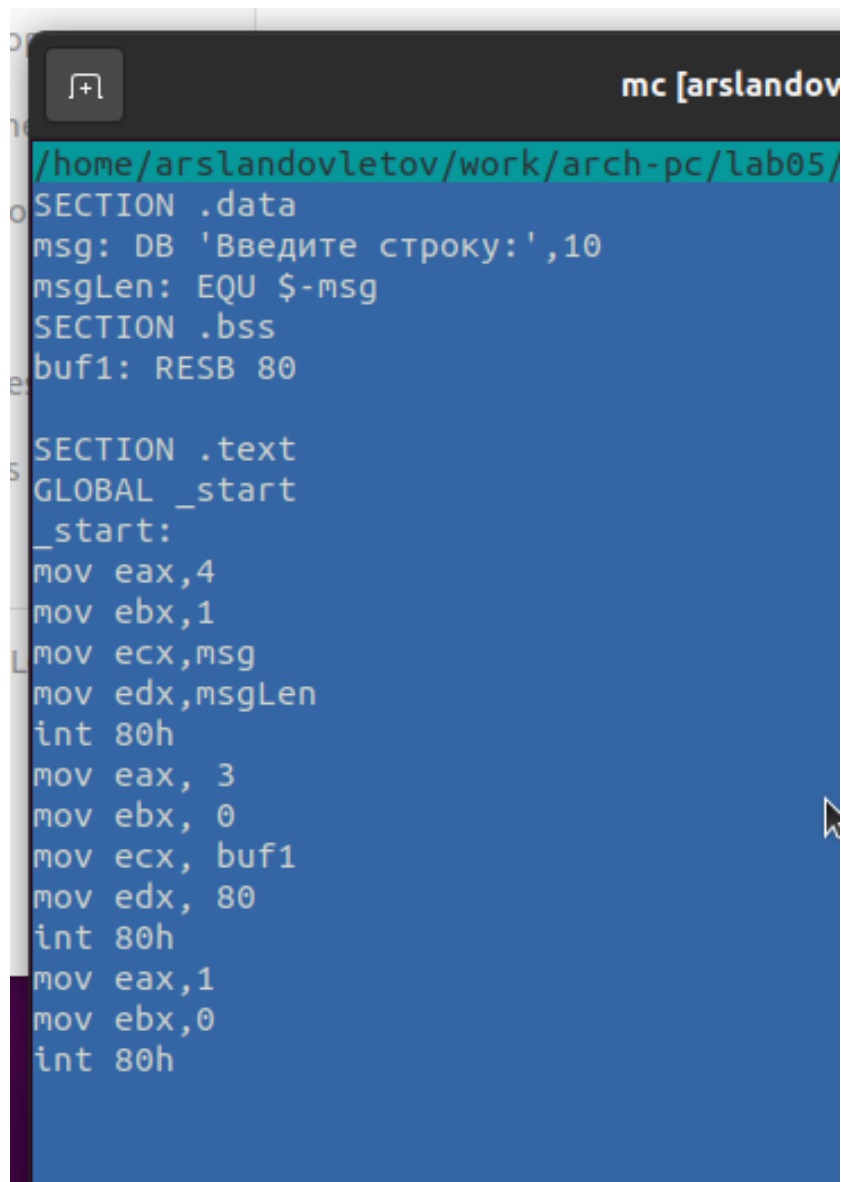


```
mc [arslandovletov@arslan-  
/home/arslandovle~lab05/lab05-1.asm [----] 7 L:  
SECTION .data  
msg: DB 'Введите строку:',10  
msgLen: EQU $-msg  
SECTION .bss  
buf1: RESB 80  
  
SECTION .text  
GLOBAL _start  
_start:  
mov eax,4  
mov ebx,1  
mov ecx,msg  
mov edx,msgLen  
int 80h.  
mov eax, 3  
mov ebx, 0  
mov ecx, buf1  
mov edx, 80  
int 80h  
mov eax,1  
mov ebx,0  
int 80h
```

Рис. 2.3: Программа в файле lab05-1.asm

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

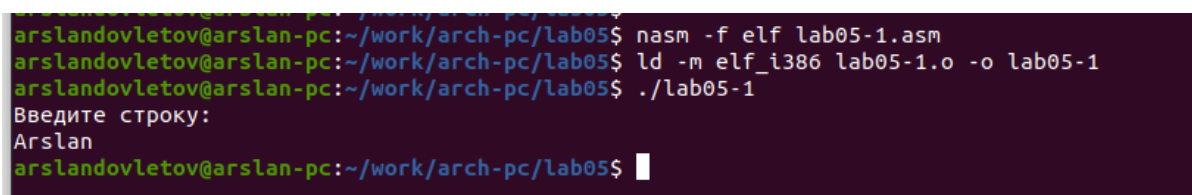




```
mc [arslandov  
/home/arslandovletov/work/arch-pc/lab05/  
SECTION .data  
msg: DB 'Введите строку:',10  
msgLen: EQU $-msg  
SECTION .bss  
buf1: RESB 80  
  
SECTION .text  
GLOBAL _start  
_start:  
mov eax,4  
mov ebx,1  
mov ecx,msg  
mov edx,msgLen  
int 80h  
mov eax, 3  
mov ebx, 0  
mov ecx, buf1  
mov edx, 80  
int 80h  
mov eax,1  
mov ebx,0  
int 80h
```

Рис. 2.4: Просмотр файла lab05-1.asm

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



```
arslandovletov@arslan-pc:~/work/arch-pc/lab05$ nasm -f elf lab05-1.asm  
arslandovletov@arslan-pc:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-1.o -o lab05-1  
arslandovletov@arslan-pc:~/work/arch-pc/lab05$ ./lab05-1  
Введите строку:  
Arslan  
arslandovletov@arslan-pc:~/work/arch-pc/lab05$
```

Рис. 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)

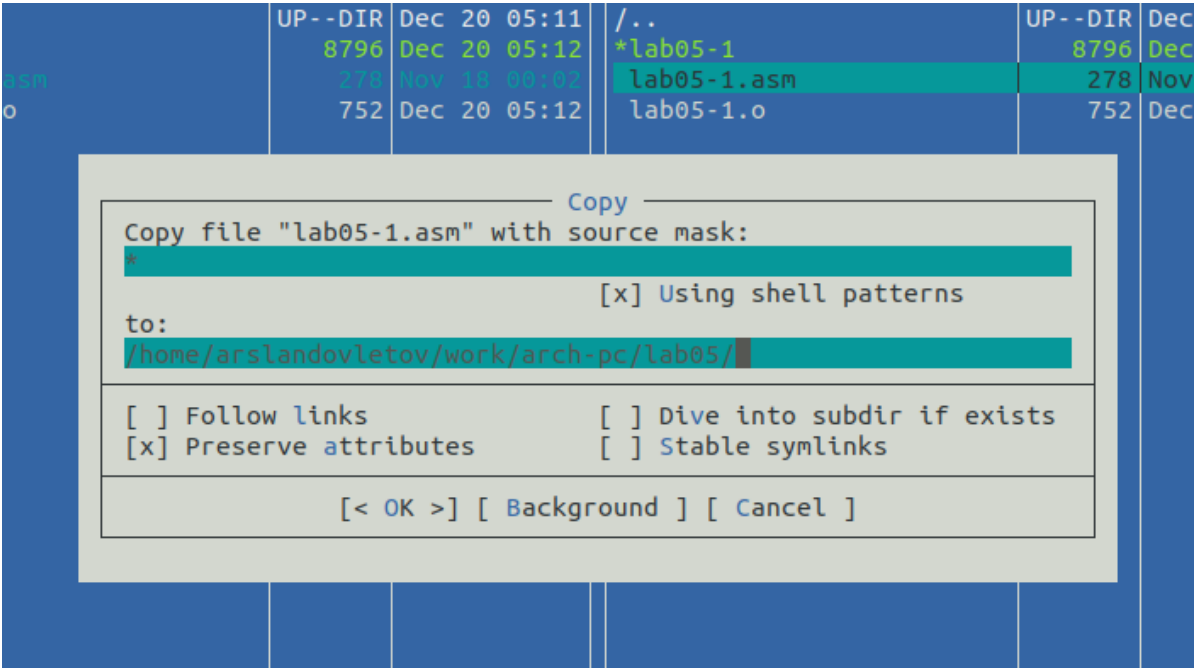
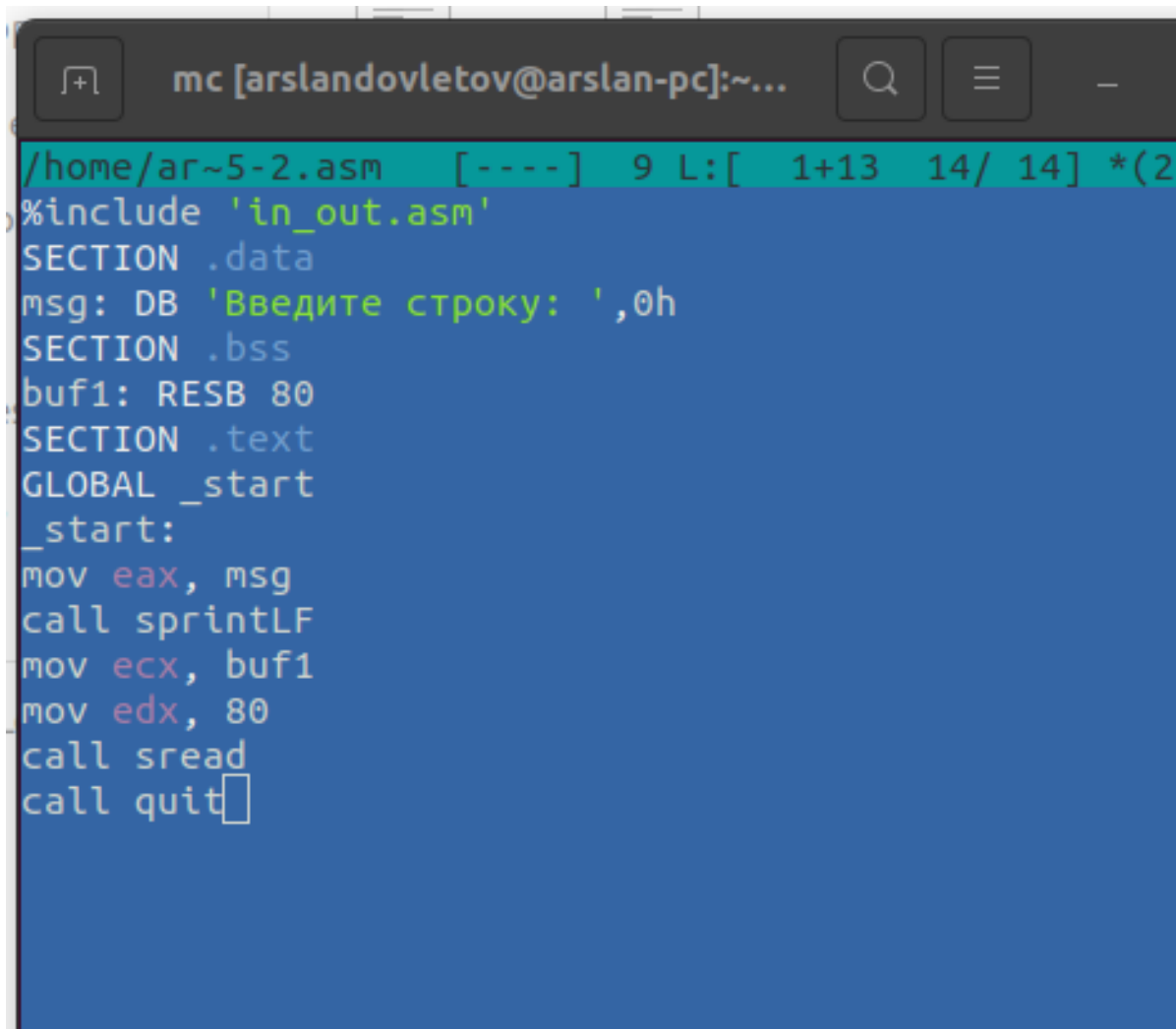


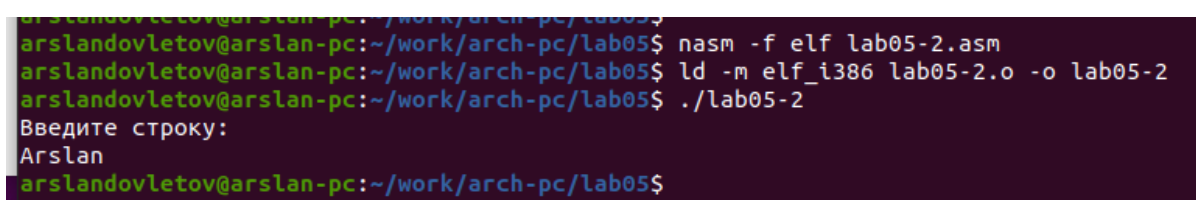
Рис. 2.6: Копирование файла

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 [arslandovletov@arslan-pc]:~...'. The editor shows assembly code for a file named 'lab05-2.asm'. The code includes a header line with file path and line numbers, followed by an include directive for 'in\_out.asm'. It then defines sections for data, bss, and text. In the data section, a message string is defined. In the bss section, a buffer is reserved. In the text section, the entry point is defined, and assembly instructions are used to move the message pointer to EAX, call the 'sprintLF' function, move the buffer address to ECX, set EDI to 80, call 'sread', and finally call 'quit'.

```
/home/ar~5-2.asm [----] 9 L:[ 1+13 14/ 14] *(2
%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
```

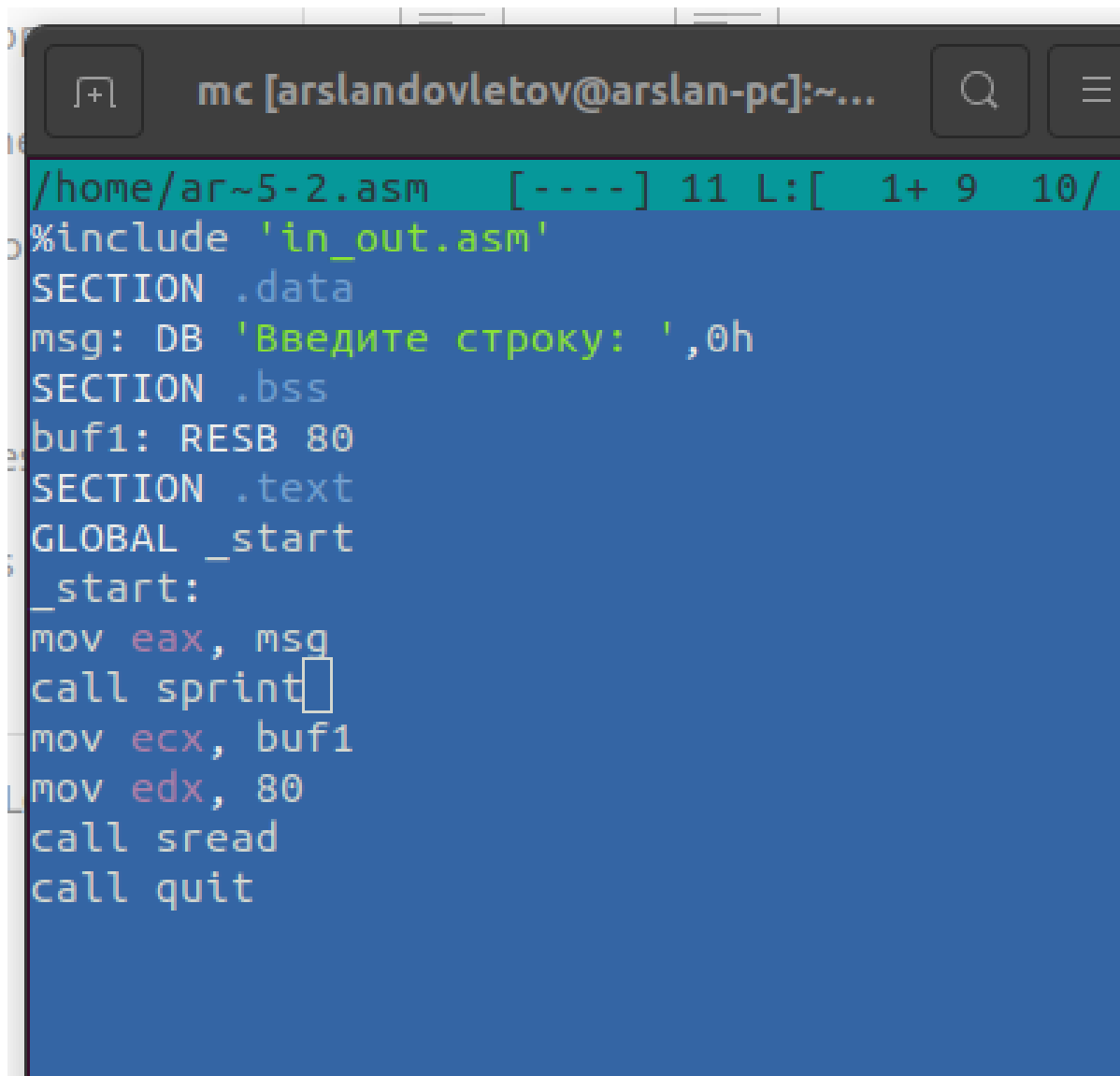
Рис. 2.7: Программа в файле lab05-2.asm

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 assemble the code into an object file, then 'ld -m elf\_i386 lab05-2.o -o lab05-2' to link it into an executable. Finally, they run './lab05-2', which prompts 'Введите строку:' (Enter a line:). The user enters 'Arslan', and the program returns to the shell prompt.

```
arslandovletov@arslan-pc:~/work/arch-pc/lab05$
arslandovletov@arslan-pc:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
arslandovletov@arslan-pc:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
arslandovletov@arslan-pc:~/work/arch-pc/lab05$ ./lab05-2
Введите строку:
Arslan
arslandovletov@arslan-pc:~/work/arch-pc/lab05$
```

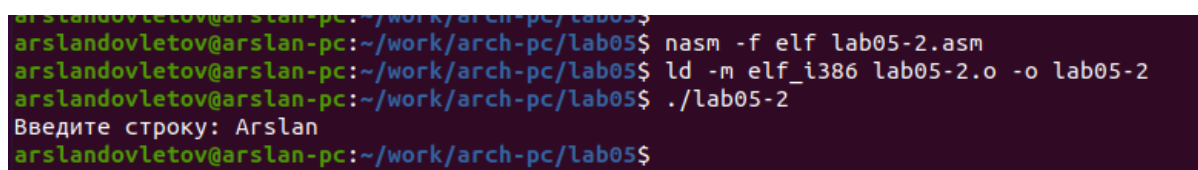
Рис. 2.8: Запуск программы lab05-2.asm

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 [arslandovletov@arslan-pc]:~...  
/home/ar~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. 2.9: Программа в файле lab05-2.asm

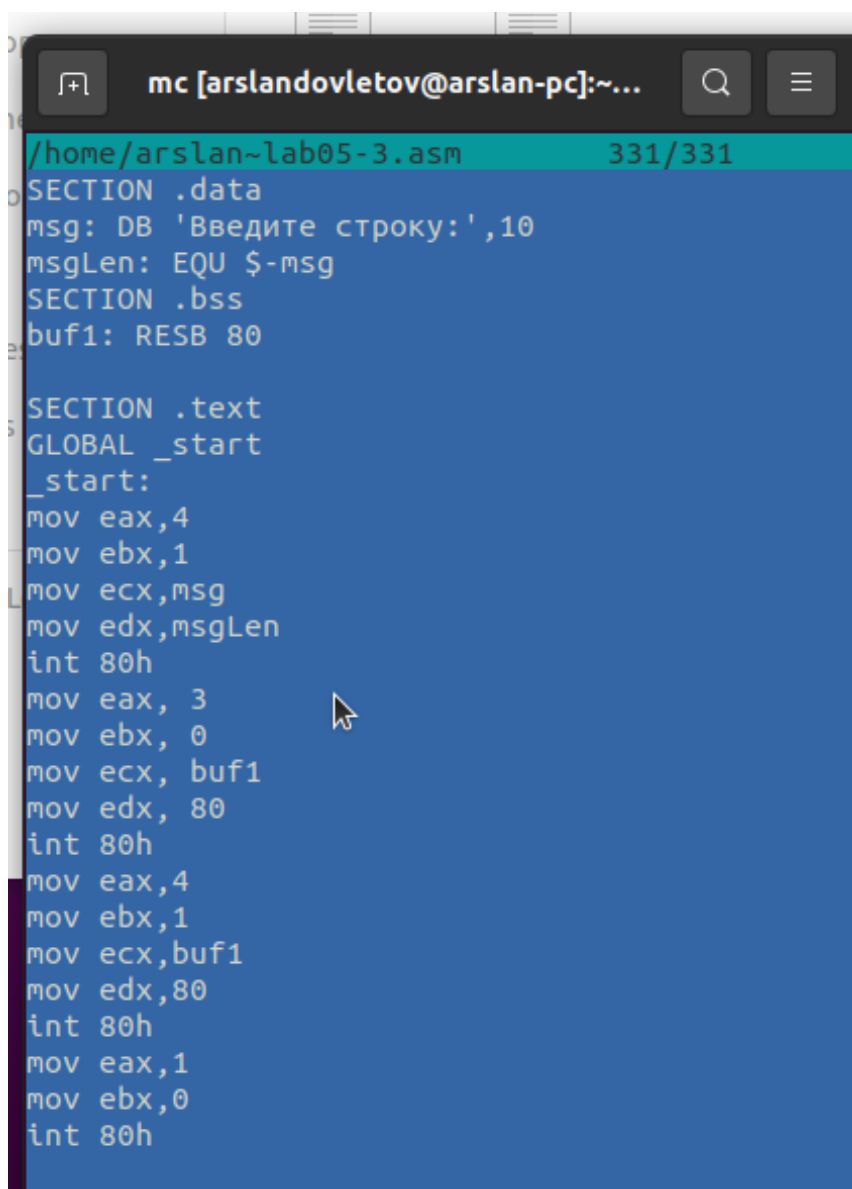


```
arslandovletov@arslan-pc:~/work/arch-pc/lab05$  
arslandovletov@arslan-pc:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm  
arslandovletov@arslan-pc:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2  
arslandovletov@arslan-pc:~/work/arch-pc/lab05$ ./lab05-2  
Введите строку: Arslan  
arslandovletov@arslan-pc:~/work/arch-pc/lab05$
```

Рис. 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

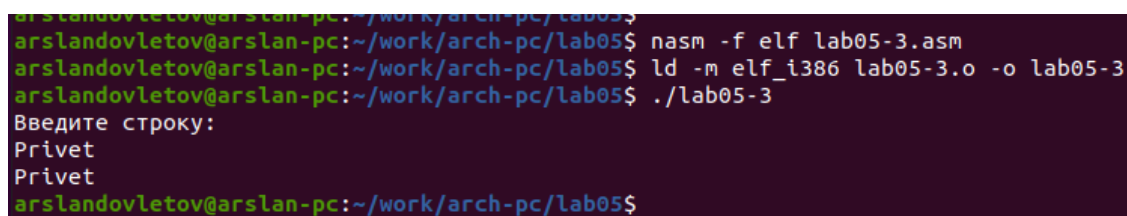
выводила введенную строку на экран. (рис. 2.11, рис. 2.12)

A screenshot of a code editor window titled 'mc [arslandovletov@arslan-pc]:~...'. The editor shows the assembly file '/home/arslan~lab05-3.asm' at line 331/331. 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,4
mov ebx,1
mov ecx,buf1
mov edx,80
int 80h
mov eax,1
mov ebx,0
int 80h
```

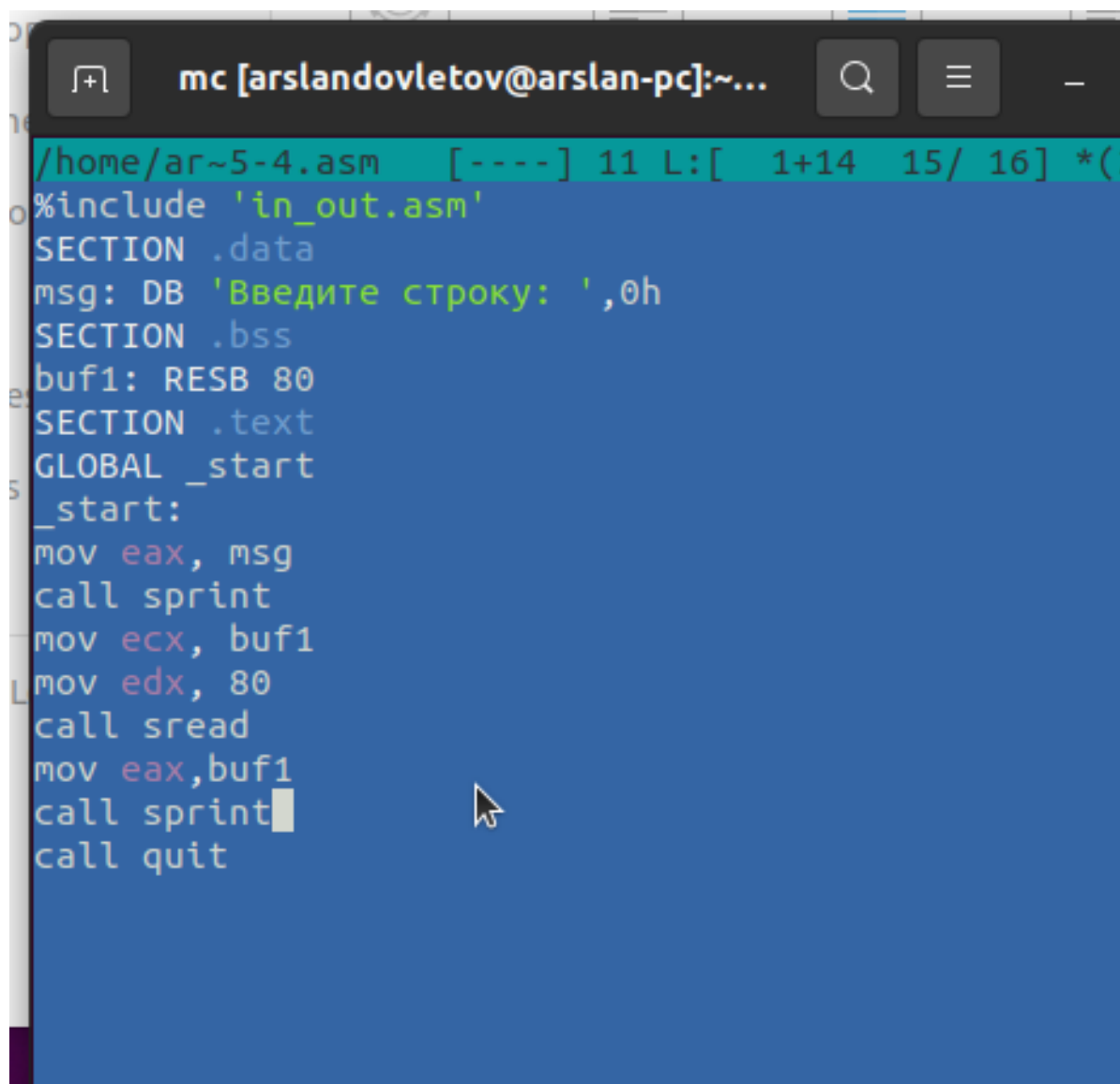
Рис. 2.11: Программа в файле lab05-3.asm

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

```
arslandovletov@arslan-pc:~/work/arch-pc/lab05$
arslandovletov@arslan-pc:~/work/arch-pc/lab05$ nasm -f elf lab05-3.asm
arslandovletov@arslan-pc:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3
arslandovletov@arslan-pc:~/work/arch-pc/lab05$ ./lab05-3
Введите строку:
Privet
Privet
arslandovletov@arslan-pc:~/work/arch-pc/lab05$
```

Рис. 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)

A screenshot of a terminal window with a dark background. The title bar shows the user 'mc [arslandovletov@arслан-pc]:~...'. The terminal displays assembly code for a program. The code includes a directive to include 'in\_out.asm', defines a data section with a message 'Введите строку:', defines a bss section with a buffer 'buf1' of size 80, and a text section with the main logic. The logic involves printing the message, reading input into the buffer, and printing the buffer's contents before calling 'quit'.

```
mc [arslandovletov@arслан-pc]:~...  
/home/ar~5-4.asm [----] 11 L:[ 1+14 15/ 16] *(2  
%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. 2.13: Программа в файле lab05-4.asm

```
arslandovletov@arslan-pc:~/work/arch-pc/lab05$  
arslandovletov@arslan-pc:~/work/arch-pc/lab05$ nasm -f elf lab05-4.asm  
arslandovletov@arslan-pc:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-4.o -o lab05-4  
arslandovletov@arslan-pc:~/work/arch-pc/lab05$ ./lab05-4  
Введите строку: Privet  
Privet  
arslandovletov@arslan-pc:~/work/arch-pc/lab05$
```

Рис. 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`.