

# Отчёт по лабораторной работе 5

Архитектура ЭВМ

Саруханов Артур Евгеньевич

# Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	15

## Список иллюстраций

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

## 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. Я открыл Midnight Commander и перешел в каталог ~/work/arch-pc. Создал новый каталог под названием lab05.

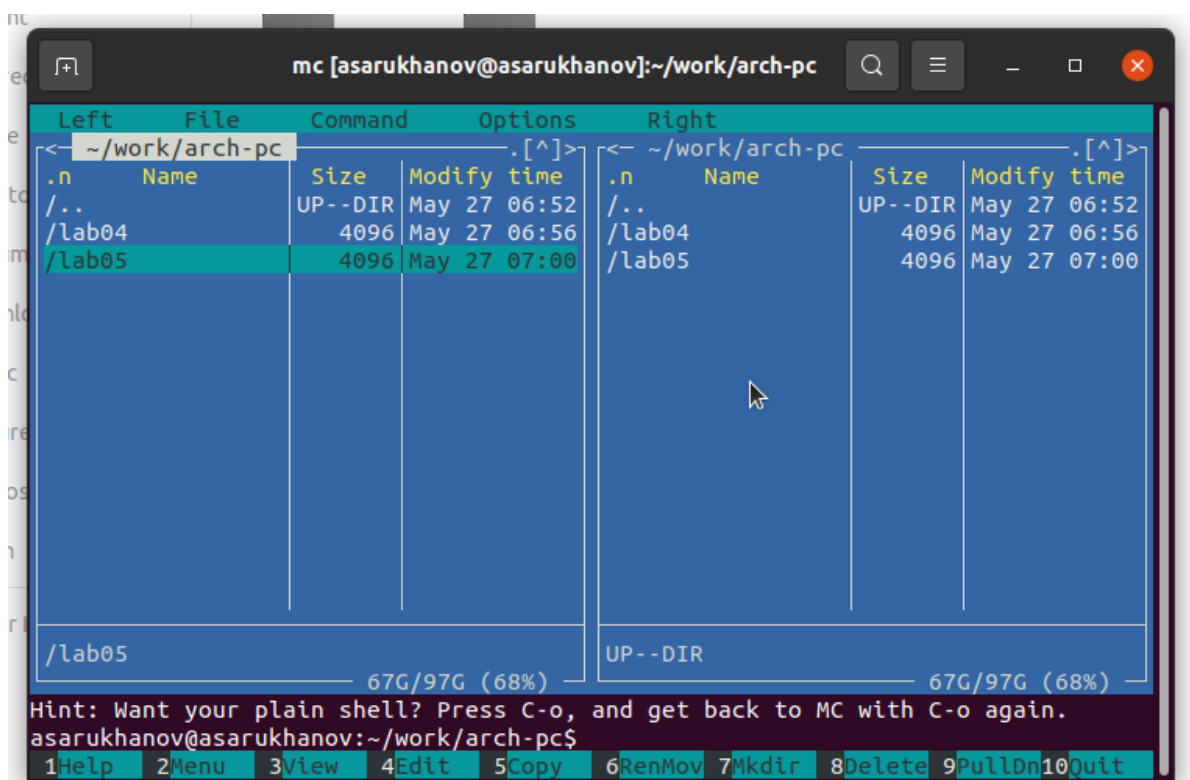


Рис. 2.1: Создание каталога

2. Внутри каталога lab05 я создал файл lab05-1.asm. Затем я открыл этот файл для редактирования и начал писать код.

```
asarukhanov@asarukhanov: ~/work/arch-pc
/home/as~5-1.asm [----] 11 L: [ 1+24 25/ 34] *(2

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: Код lab05-1.asm

3. Я также открыл файл lab05-1.asm для просмотра и проверил, что код был написан правильно.



```
mc [asarukhanov@asarukhanov]:~/work/arch-pc/la...
/home/asarukhanov/w~/lab05/lab05-1.asm 333/333

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. 2.3: Просмотр файла lab05-1.asm

4. После того, как я получил исполняемый файл из кода, я проверил его работу, чтобы убедиться, что все функционирует должным образом.



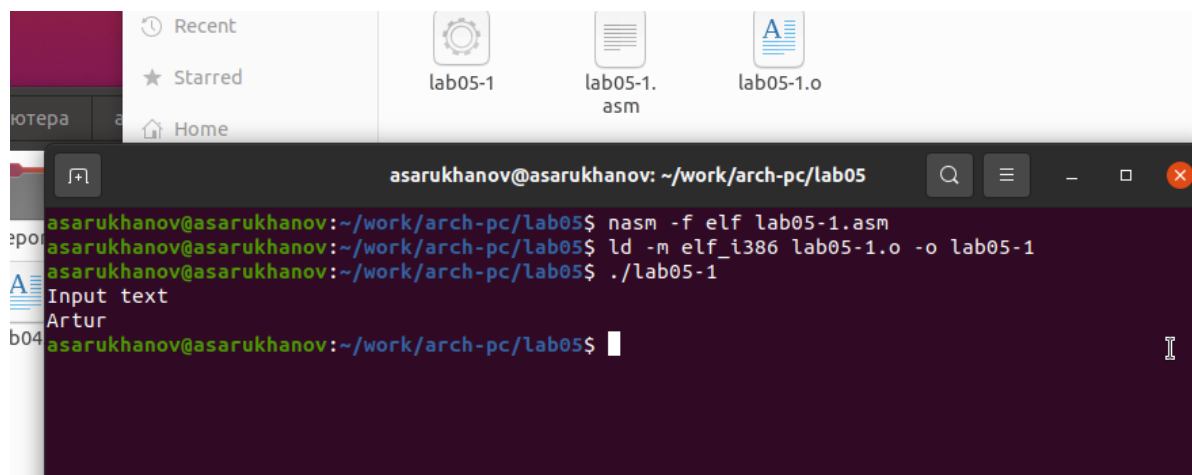


Рис. 2.4: Запуск программы lab05-1.asm

5. Я скачал файл in\_out.asm и добавил его в рабочий каталог. Затем я скопировал содержимое файла lab05-1.asm в новый файл под названием lab05-2.asm.

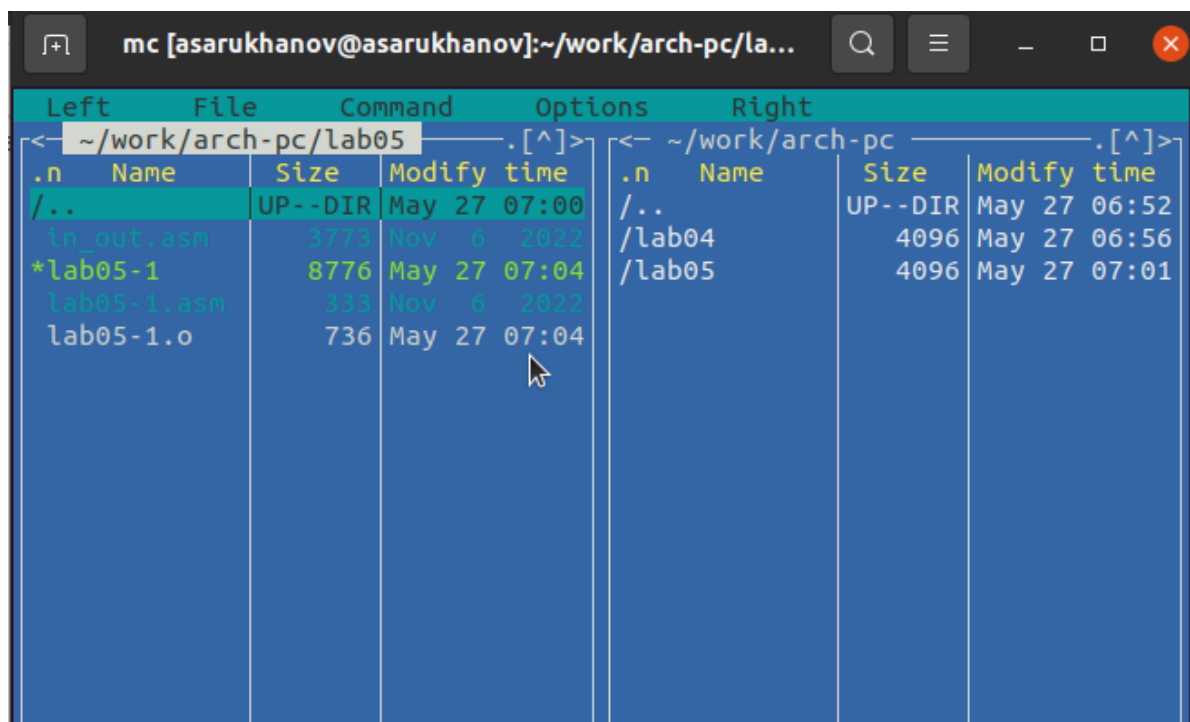
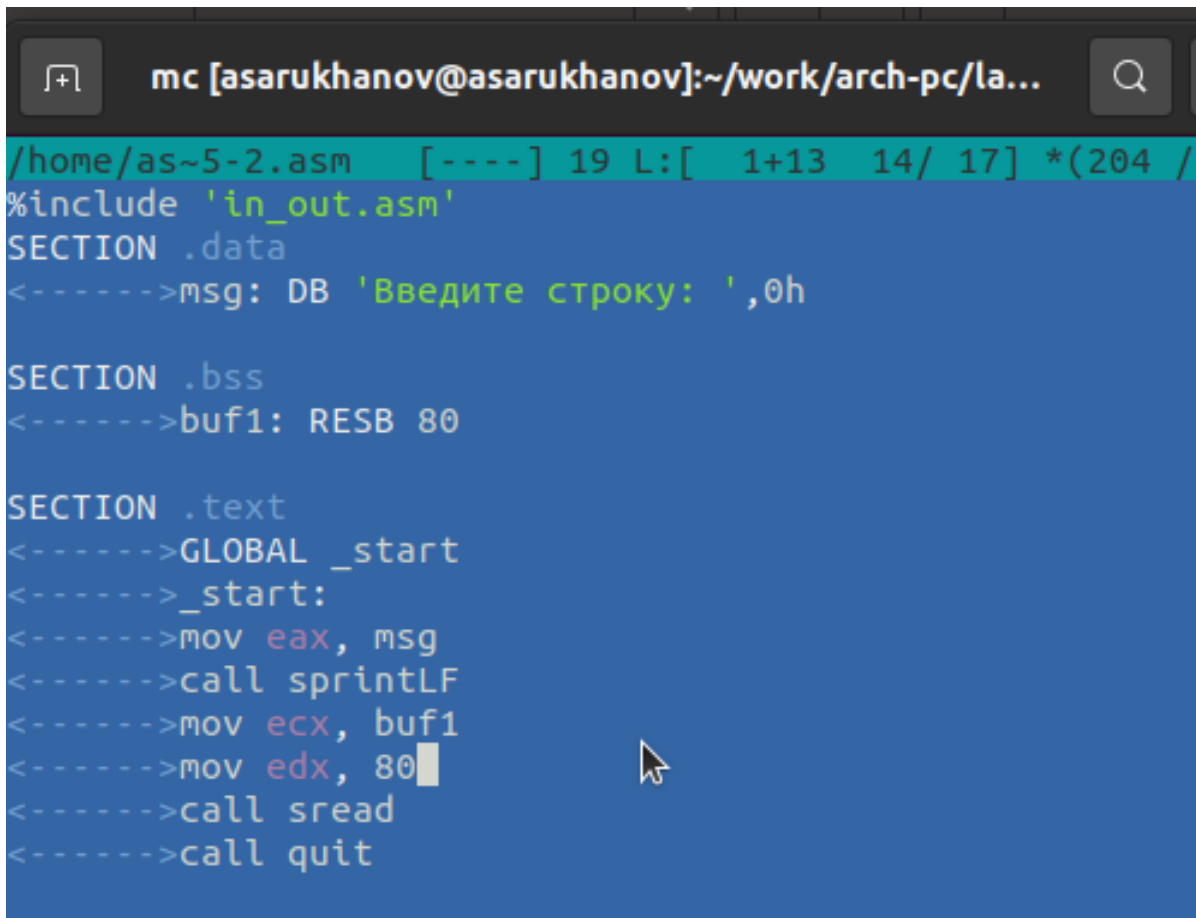


Рис. 2.5: Копирование файла

6. В файле lab05-2.asm я написал код программы. После этого я скомпилиро-

вал программу и проверил ее запуск.

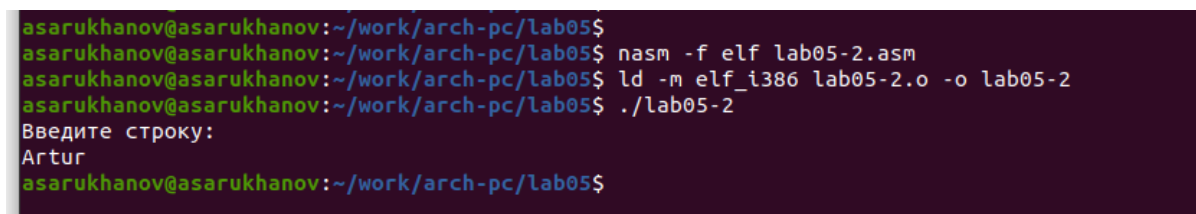


```
mc [asarukhanov@asarukhanov]:~/work/arch-pc/la...
/home/as~5-2.asm [----] 19 L:[ 1+13 14/ 17] *(204 /
#include 'in_out.asm'
SECTION .data
<----->msg: DB 'Введите строку: ',0h

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
```

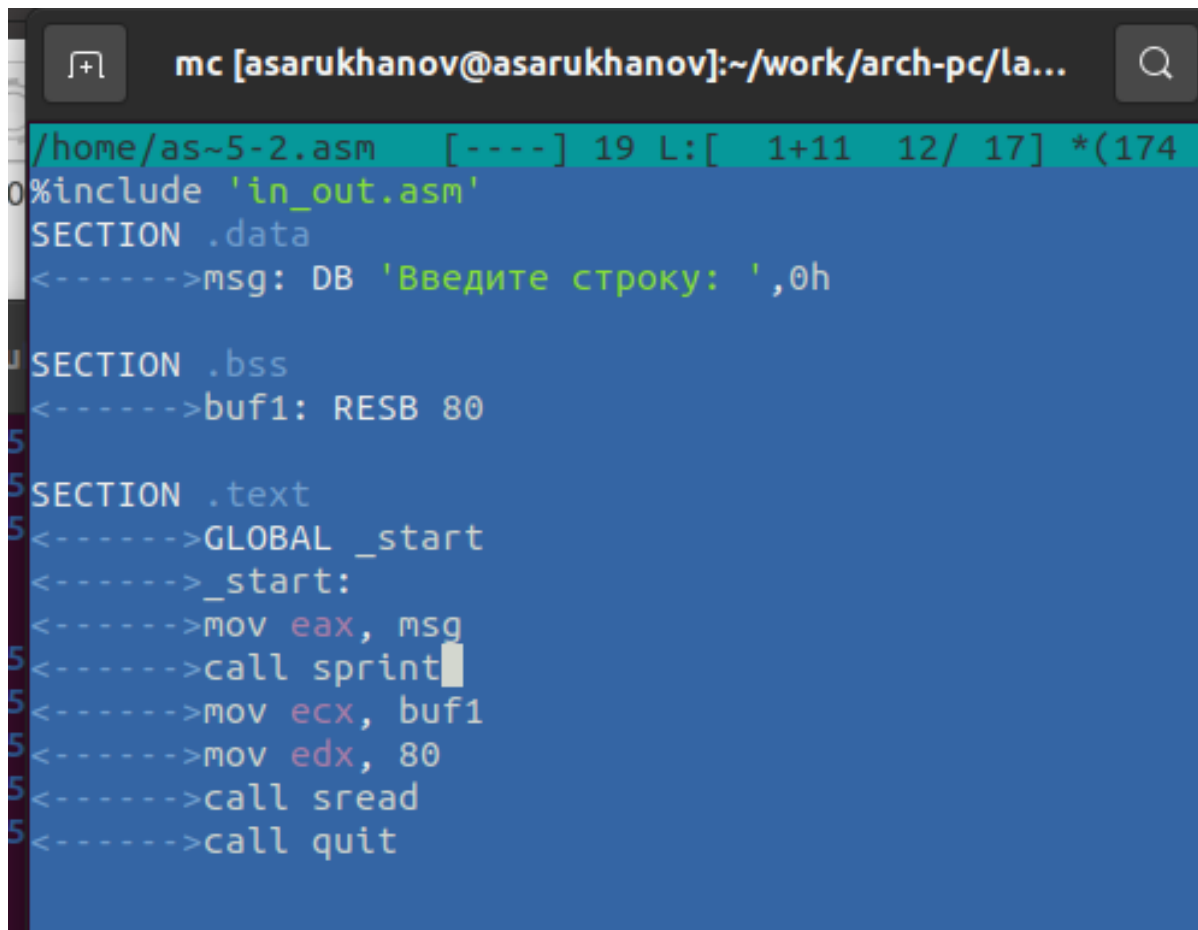
Рис. 2.6: Код lab05-2.asm



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

Рис. 2.7: Запуск программы lab05-2.asm

7. Внутри файла lab05-2.asm я заменил подпрограмму sprintLF на sprint. Затем я пересобрал исполняемый файл. Теперь при выводе строки нет перехода на следующую строку.

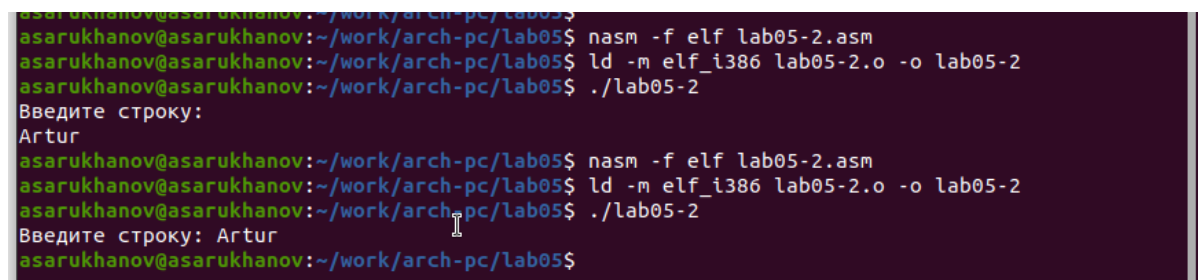


```
mc [asarukhanov@asarukhanov]:~/work/arch-pc/la...
/home/as~5-2.asm [----] 19 L:[ 1+11 12/ 17] *(174
0%include 'in_out.asm'
SECTION .data
<----->msg: DB 'Введите строку: ',0h

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
```

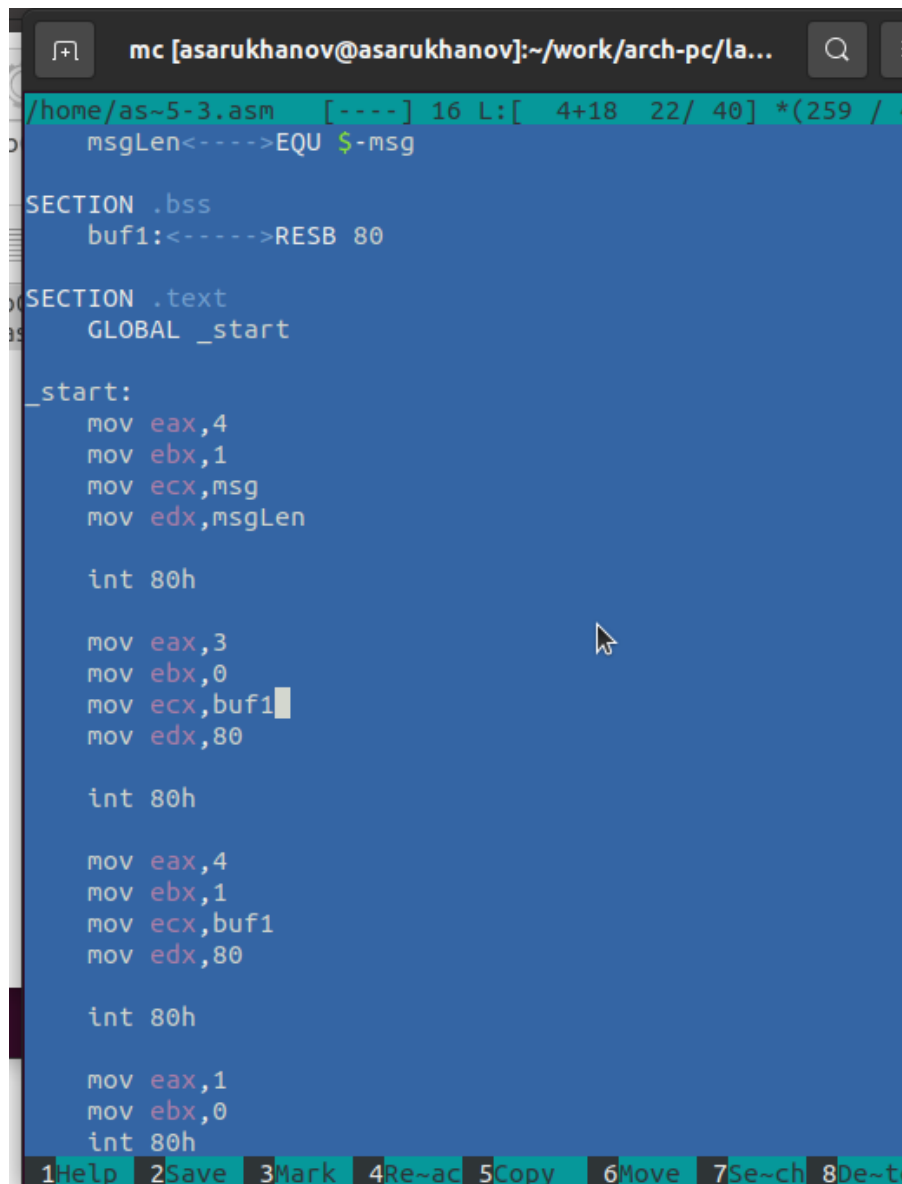
Рис. 2.8: Код lab05-2.asm



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

Рис. 2.9: Запуск программы lab05-2.asm

8. Я скопировал программу lab05-1.asm и изменил код так, чтобы сначала выводилось приглашение “Введите строку:”, затем пользователь вводил строку с клавиатуры, и введенная строка выводилась на экран.



```
mc [asarukhanov@asarukhanov]:~/work/arch-pc/la...
/home/as~5-3.asm [----] 16 L: [ 4+18 22/ 40] *(259 / 4
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

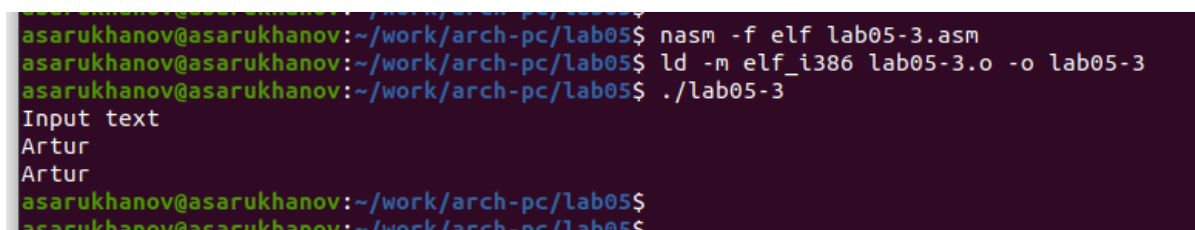
    int 80h

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

    int 80h

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

Рис. 2.10: Код lab05-3.asm

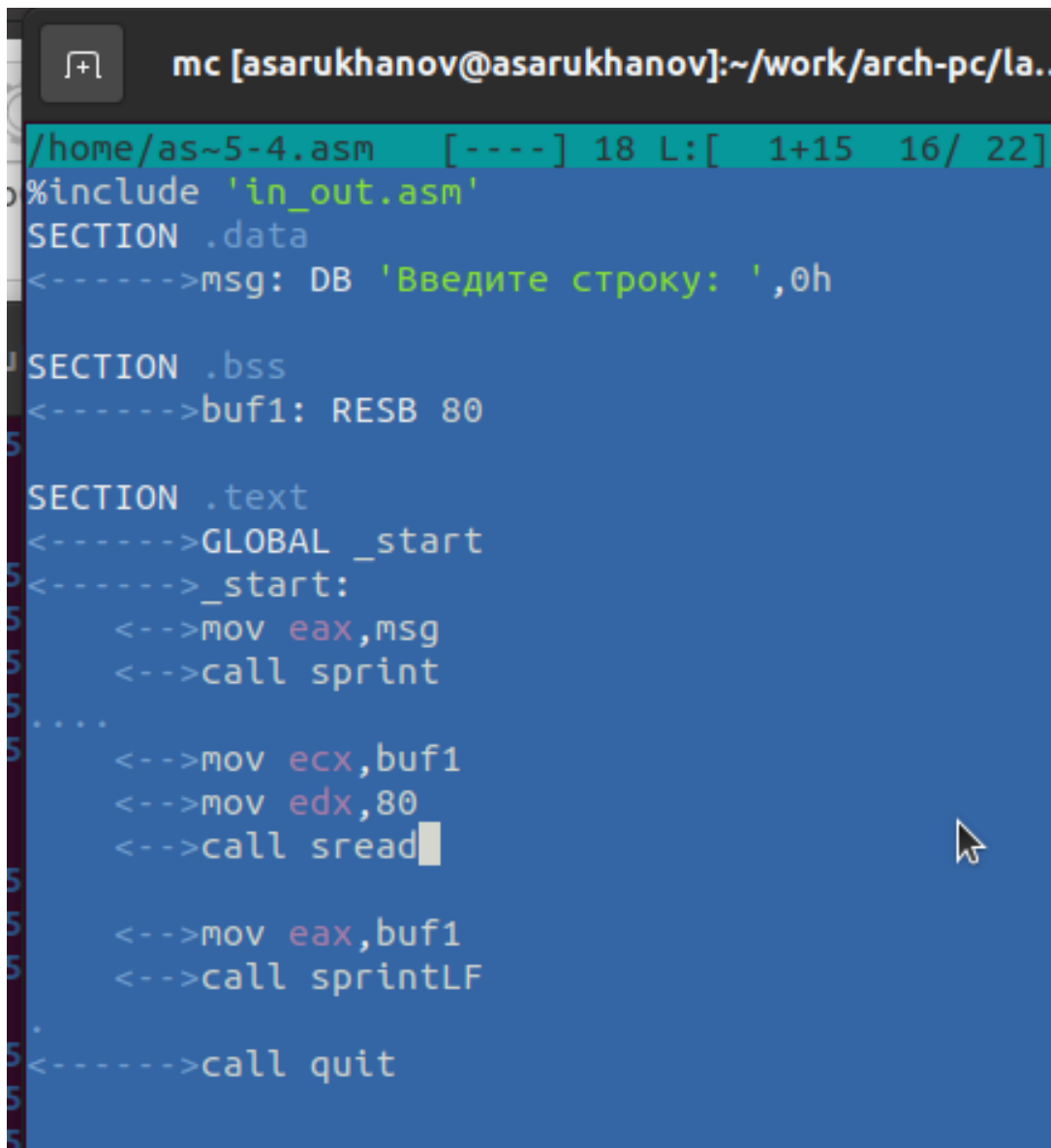


```
asarukhanov@asarukhanov:~/work/arch-pc/lab05$ nasm -f elf lab05-3.asm
asarukhanov@asarukhanov:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3
asarukhanov@asarukhanov:~/work/arch-pc/lab05$ ./lab05-3
Input text
Artur
Artur
asarukhanov@asarukhanov:~/work/arch-pc/lab05$
asarukhanov@asarukhanov:~/work/arch-pc/lab05$
```

Рис. 2.11: Запуск программы lab05-3.asm

9. Затем я скопировал программу lab05-2.asm и выполнил аналогичные

действия, описанные выше, но теперь использовал возможности из файла in\_out.asm.



```
mc [asarukhanov@asarukhanov]:~/work/arch-pc/la..  
/home/as~5-4.asm [----] 18 L:[ 1+15 16/ 22]  
%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. 2.12: Код lab05-4.asm

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

Рис. 2.13: Запуск программы lab05-4.asm

## 3 Выводы

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