

Отчёт по лабораторной работе 5

дисциплина: Архитектура компьютера

Алиев Руслан Нияз оглы

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Знакомство с Midnight Commander	6
2.2	Подключение внешнего файла in_out.asm	11
2.3	Задание для самостоятельной работы	15
3	Выводы	18

Список иллюстраций

2.1	Запуск Midnight Commander	6
2.2	Создание каталога	7
2.3	Создание файла lab05-1.asm	8
2.4	Программа lab05-1.asm	9
2.5	Просмотр файла lab05-1.asm	10
2.6	Запуск программы lab05-1.asm	11
2.7	Копирование файла in_out.asm	11
2.8	Копирование файла lab05-1.asm	12
2.9	Программа lab05-2.asm	13
2.10	Запуск программы lab05-2.asm	13
2.11	Программа в файле lab05-2.asm	14
2.12	Запуск программы lab05-2.asm	14
2.13	Программа lab05-3.asm	15
2.14	Запуск программы lab05-3.asm	16
2.15	Программа lab05-4.asm	17
2.16	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`.

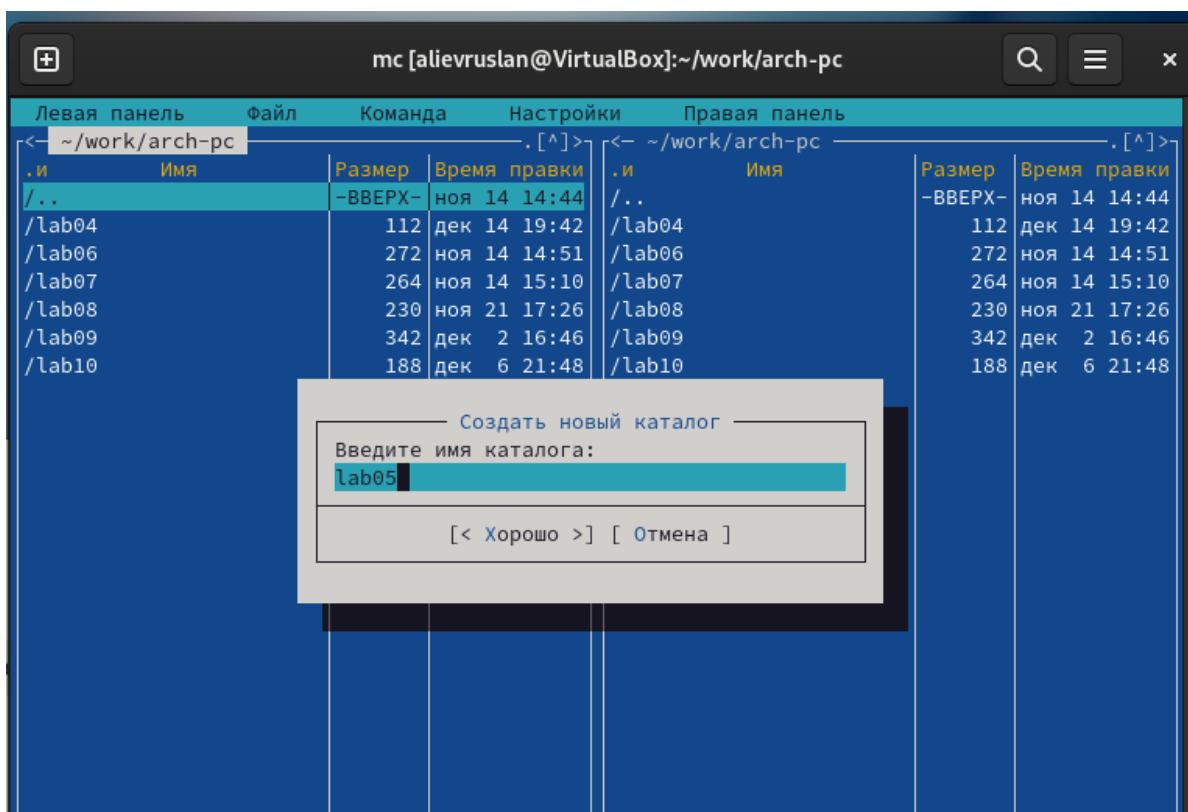


Рис. 2.2: Создание каталога

С помощью команды `touch` создаю файл `lab05-1.asm` (см. рис. 2.3).

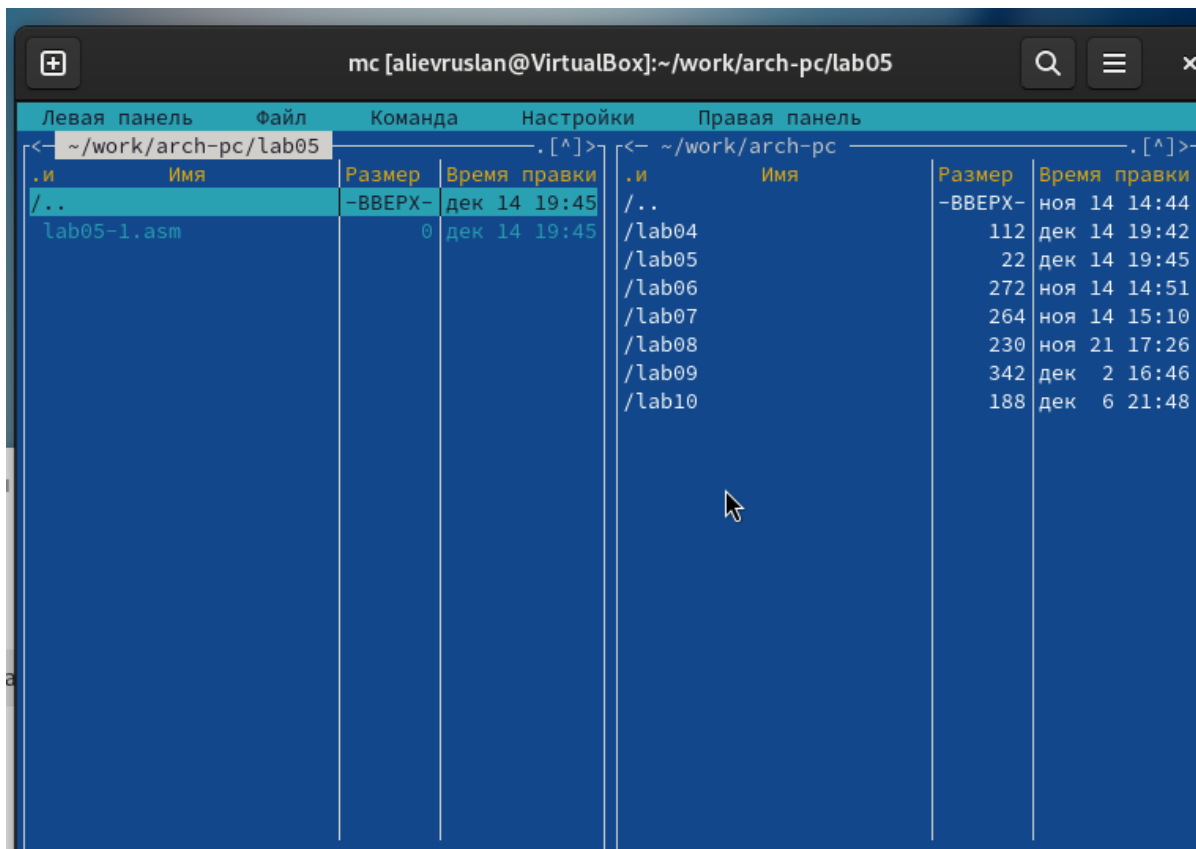


Рис. 2.3: Создание файла lab05-1.asm

Открываю файл для редактирования, нажав F4, и выбираю редактор `mcedit`, после чего пишу код программы согласно заданию (см. рис. 2.4).

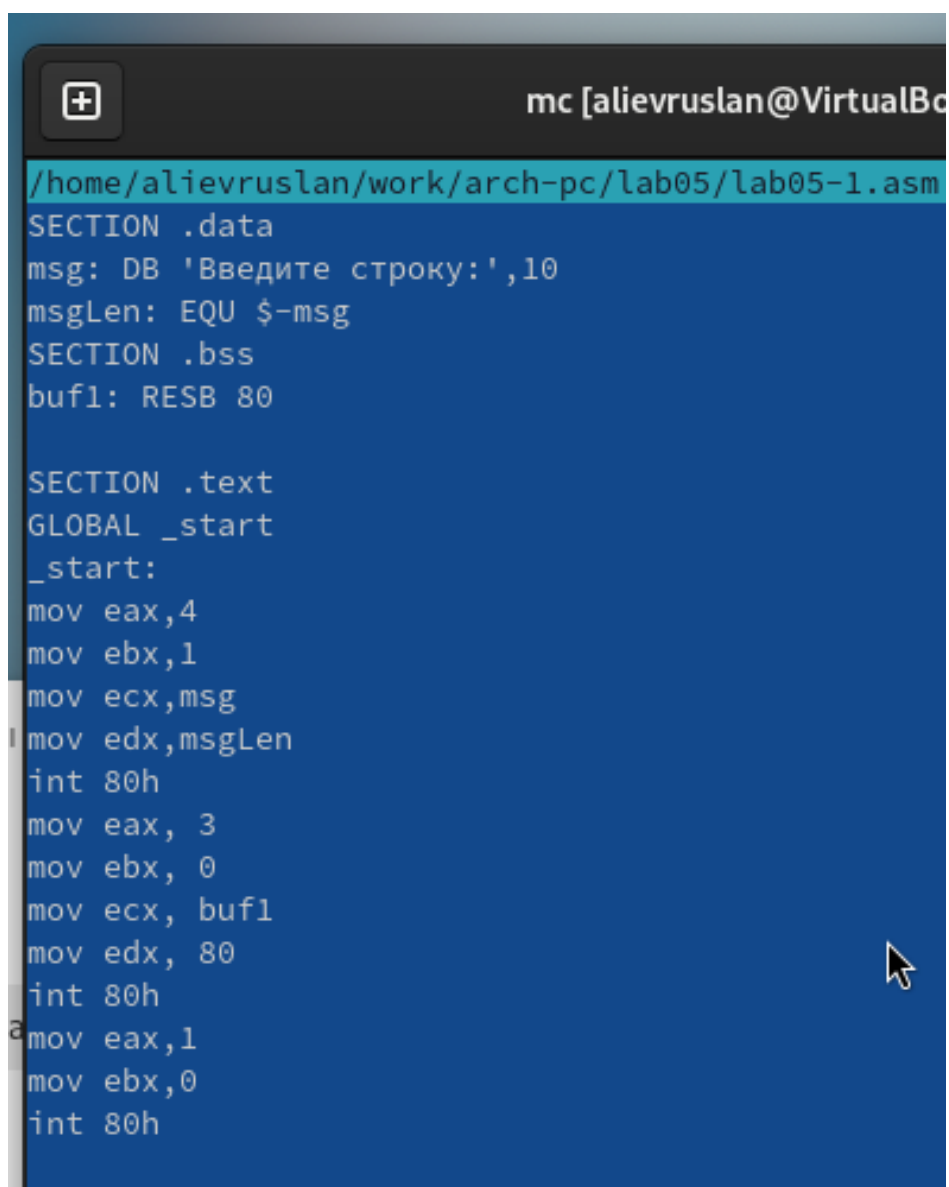


```
lab05-1.asm [----] 10 L: [ 1+14 15/ 23
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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: Программа lab05-1.asm

Для проверки содержимого файла открываю его на просмотр, нажав F3, и убеждаюсь, что код написан верно (см. рис. 2.5).



```
mc [alievruslan@VirtualBo...]  
/home/alievruslan/work/arch-pc/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. 2.5: Просмотр файла lab05-1.asm

Транслирую файл программы в объектный файл, а затем выполняю компоновку, в результате чего получаю исполняемый файл программы (см. рис. 2.6).

```

alievruslan@VirtualBox:~/work/arch-pc/lab05$ nasm -f elf lab05-1.asm
alievruslan@VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-1.o -o lab05-1
alievruslan@VirtualBox:~/work/arch-pc/lab05$ ./lab05-1
Введите строку:
Ruslan
alievruslan@VirtualBox:~/work/arch-pc/lab05$

```

Рис. 2.6: Запуск программы lab05-1.asm

2.2 Подключение внешнего файла in_out.asm

Скачиваю файл `in_out.asm` и размещаю его в рабочем каталоге (см. рис. 2.7). Для копирования файла использую клавишу F5, а для перемещения — клавишу F6.

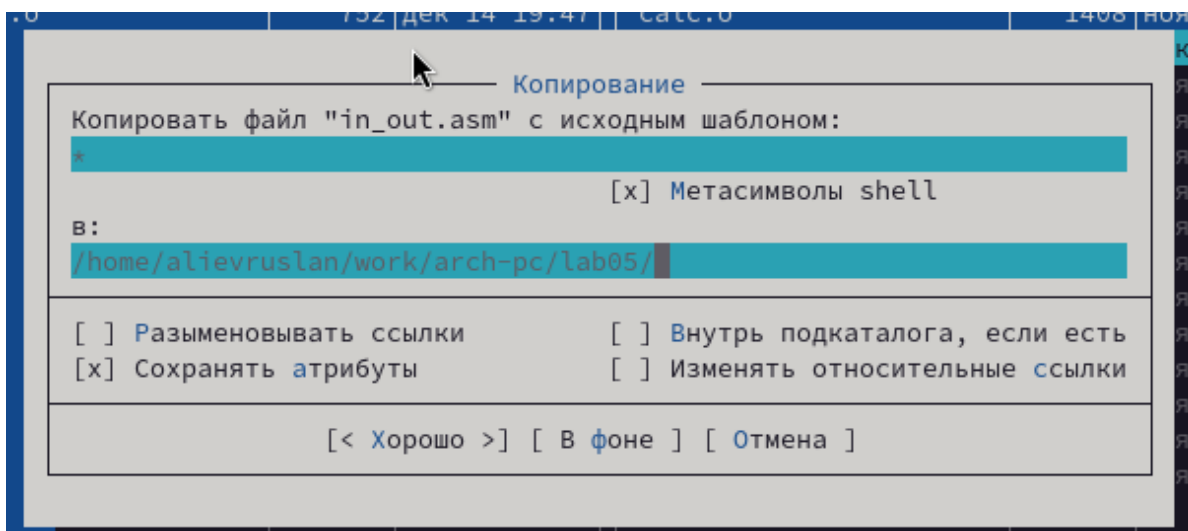


Рис. 2.7: Копирование файла in_out.asm

Копирую файл `lab05-1.asm`, создавая его копию под именем `lab05-2.asm` (см. рис. 2.8).

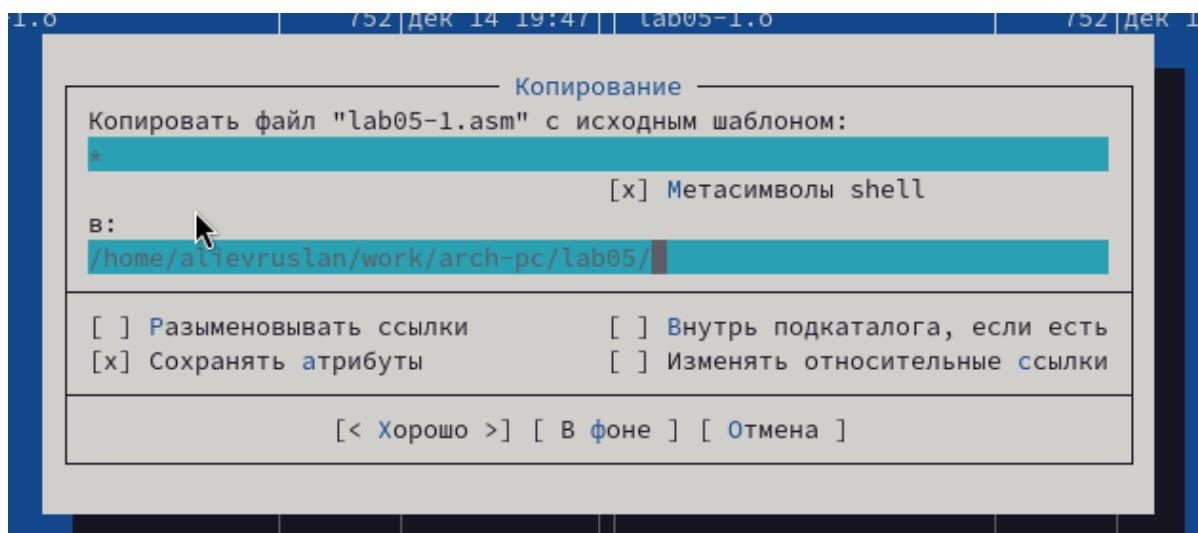
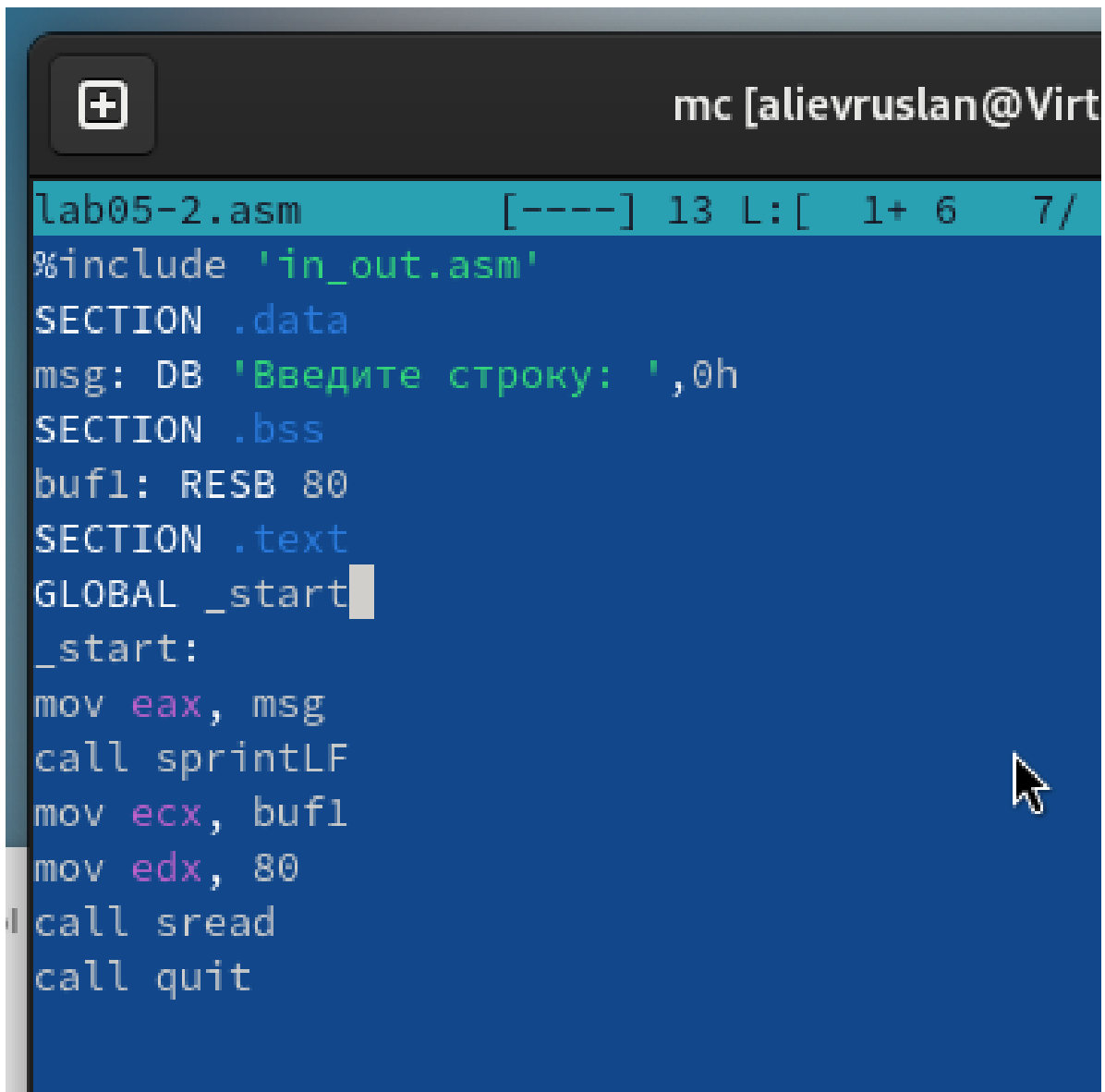


Рис. 2.8: Копирование файла lab05-1.asm

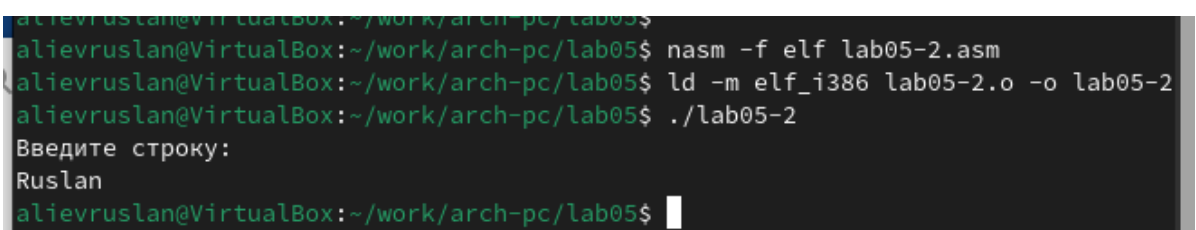
Пишу код для программы lab05-2.asm, используя подпрограммы из внешнего файла in_out.asm (см. рис. 2.9).



```
lab05-2.asm [-----] 13 L: [ 1+ 6 7/
#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. 2.9: Программа lab05-2.asm

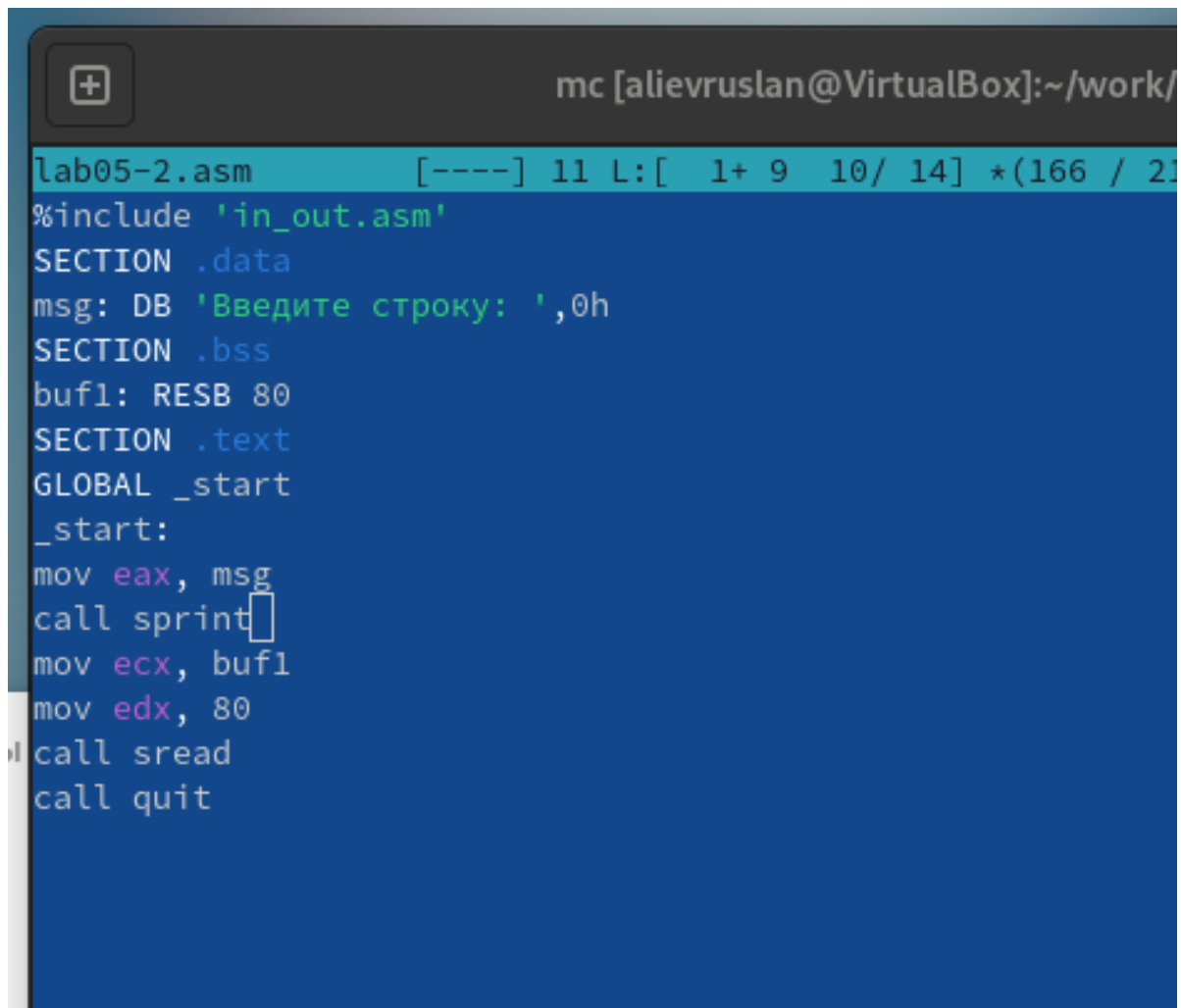
Компилирую программу и проверяю её запуск (см. рис. 2.10).



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

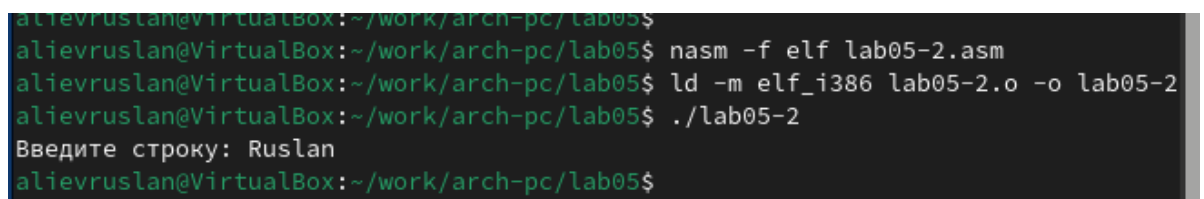
Рис. 2.10: Запуск программы lab05-2.asm

В файле lab05-2.asm заменяю подпрограмму sprintLF на sprint. После этого заново собираю исполняемый файл (см. рис. 2.11 и 2.12).



```
mc [alievruslan@VirtualBox]:~/work/
lab05-2.asm      [----] 11 L:[ 1+ 9 10/ 14] *(166 / 21
#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. 2.11: Программа в файле lab05-2.asm




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

Рис. 2.12: Запуск программы lab05-2.asm

Теперь программа выводит строку без перехода на новую строку в конце.

2.3 Задание для самостоятельной работы

Копирую программу lab05-1.asm и модифицирую код, чтобы она работала по следующему алгоритму (см. рис. 2.13 и 2.14): - выводит приглашение “Введите строку:”; - принимает строку с клавиатуры; - отображает введенную строку на экране.



```
mc [alievruslan@VirtualBox]:~/V
lab05-3.asm [----] 10 L: [ 1+15 16/ 28] *(215
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

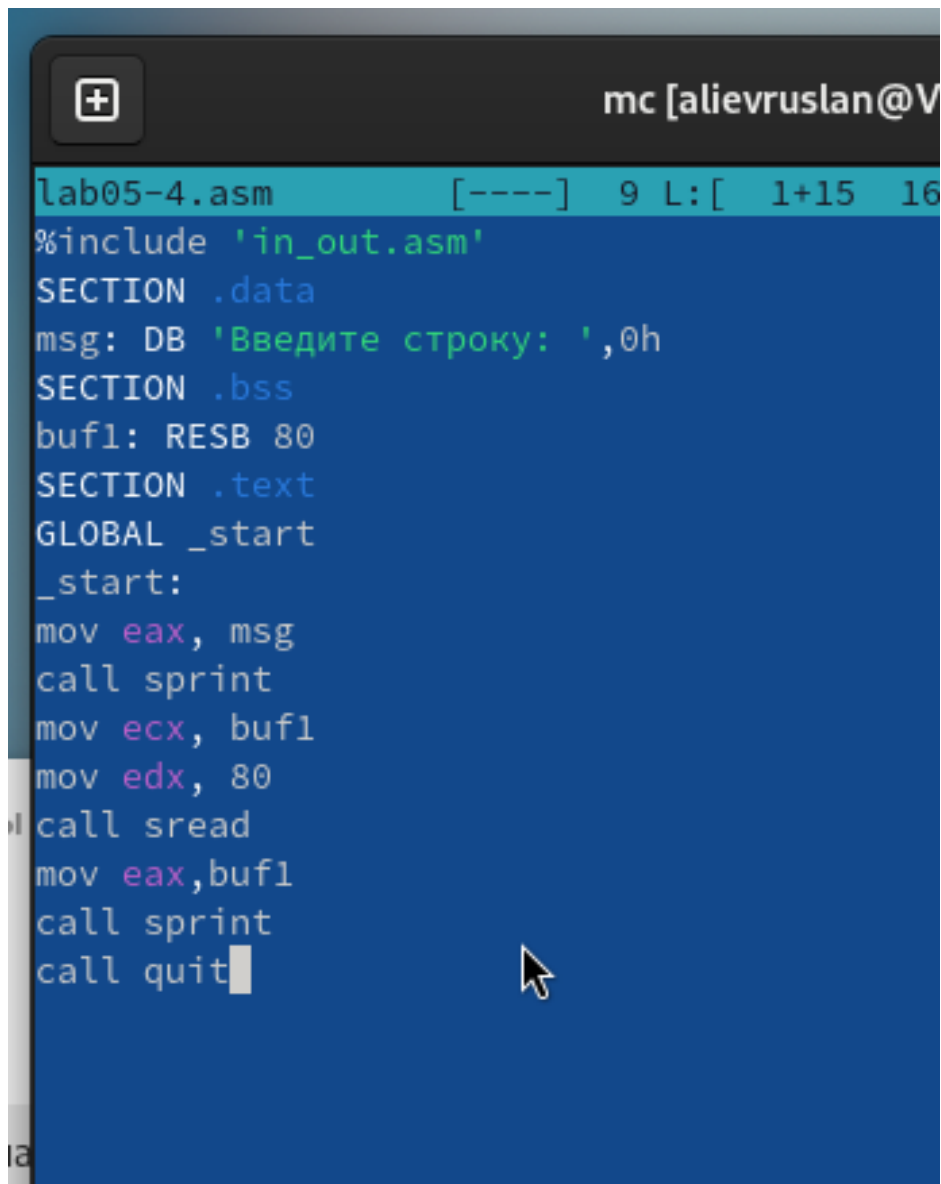
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h.
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h.
mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,80
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рис. 2.13: Программа lab05-3.asm

```
alievruslan@VirtualBox:~/work/arch-pc/lab05$  
alievruslan@VirtualBox:~/work/arch-pc/lab05$ nasm -f elf lab05-3.asm  
alievruslan@VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3  
alievruslan@VirtualBox:~/work/arch-pc/lab05$ ./lab05-3  
Введите строку:  
Ruslan  
Ruslan  
alievruslan@VirtualBox:~/work/arch-pc/lab05$
```

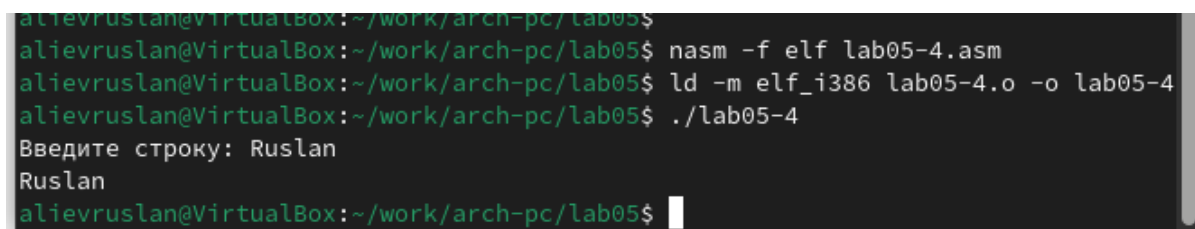
Рис. 2.14: Запуск программы lab05-3.asm

Аналогично, копирую программу lab05-2.asm и изменяю код, теперь используя подпрограммы из файла in_out.asm (см. рис. 2.15 и 2.16).



```
lab05-4.asm      [----]  9 L:[  1+15  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
```

Рис. 2.15: Программа lab05-4.asm



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

Рис. 2.16: Запуск программы lab05-4.asm

3 Выводы

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