

Отчёт по лабораторной работе 5

Архитектура компьютера

Осипов Павел Александрович НБИбд-02-23

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	16

Список иллюстраций

2.1	Создание каталога	6
2.2	Программа 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	Программа lab05-2.asm	10
2.7	Запуск программы lab05-2.asm	10
2.8	Программа lab05-2.asm	11
2.9	Запуск программы lab05-2.asm	12
2.10	Программа lab05-3.asm	13
2.11	Запуск программы lab05-3.asm	14
2.12	Программа lab05-4.asm	15
2.13	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

1. Открыл Midnight Commander и перешел в каталог ~/work/arch-pc. Создал каталог lab05.

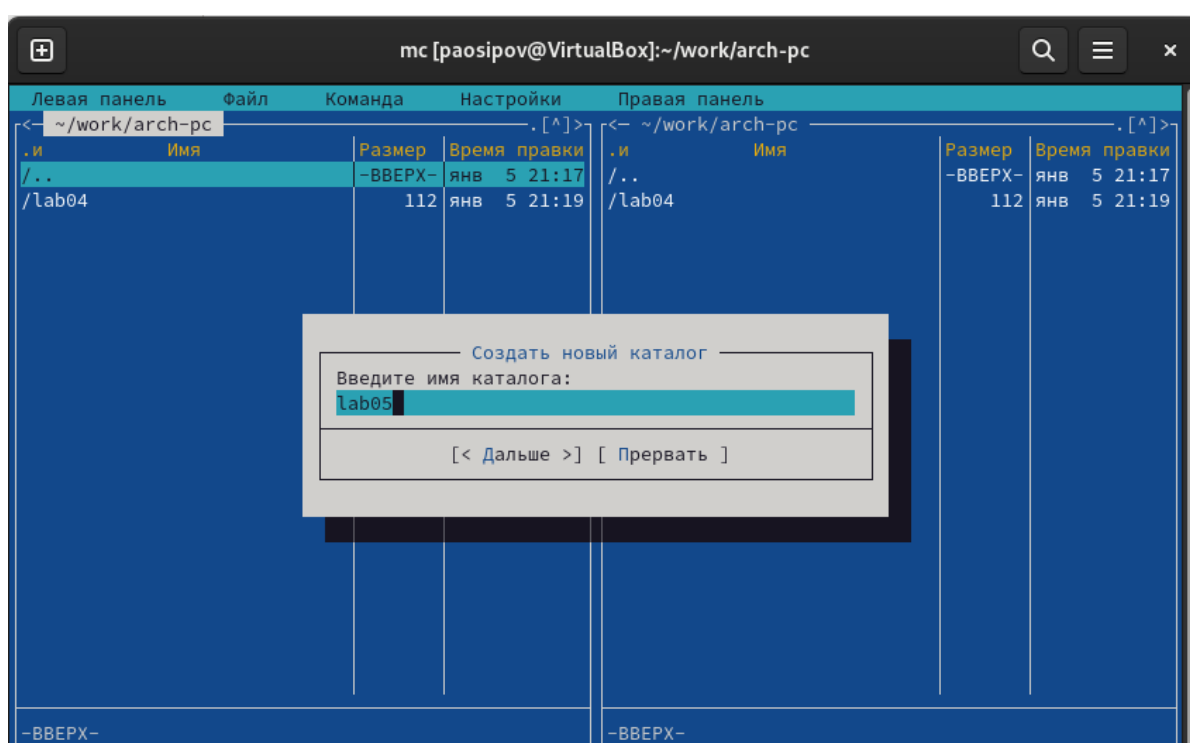


Рис. 2.1: Создание каталога

2. В каталоге lab05 создал файл lab05-1.asm. Открыл файл для редактирования и написал соответствующий код.

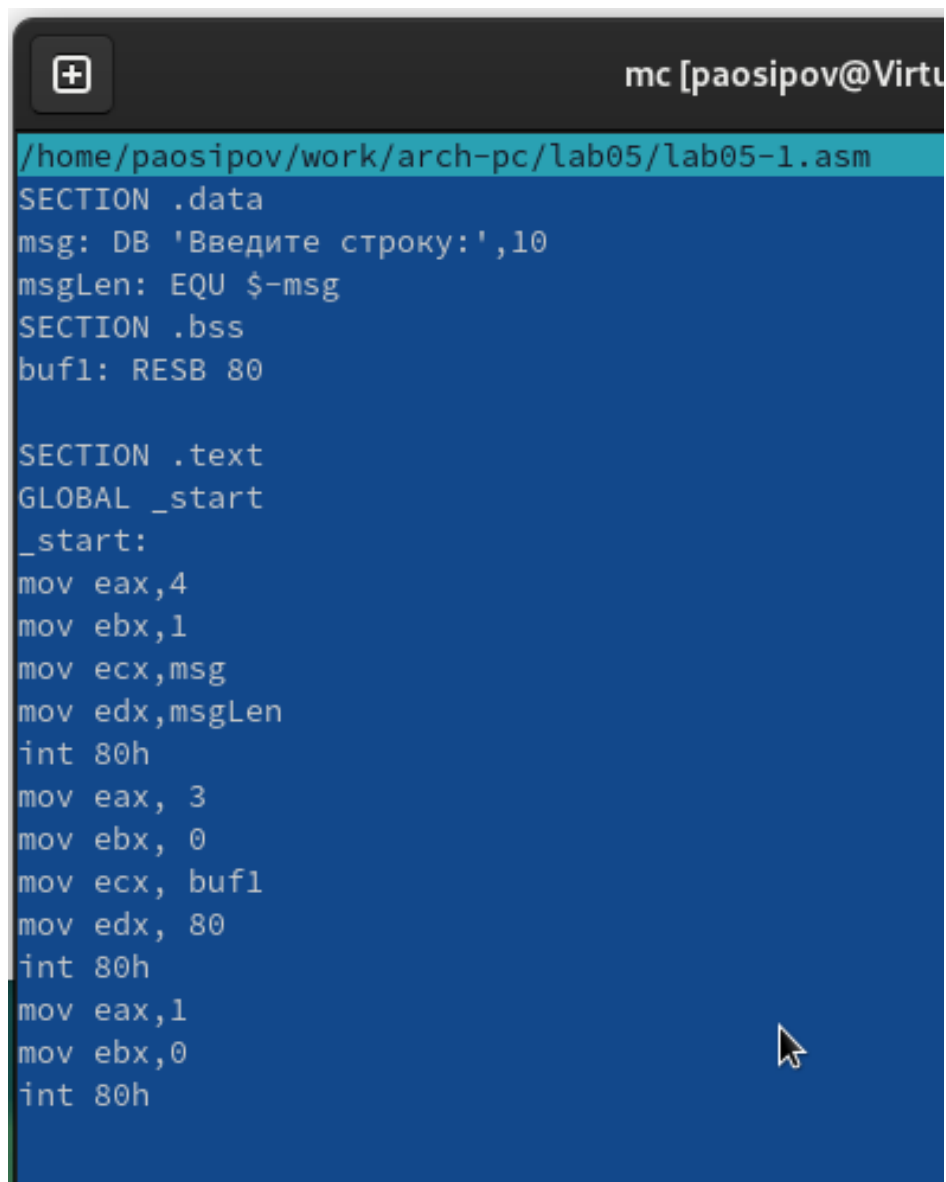


```
mc [paosipov@Virt
lab05-1.asm [----] 14 L: [ 1+12 13/ 23]
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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. 2.2: Программа lab05-1.asm

3. Открыл файл lab05-1.asm для просмотра и проверил написанный код.



```
mc [paosipov@Virtu
/home/paosipov/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. 2.3: Просмотр файла lab05-1.asm

4. Скомпилировал файл lab05-1.asm и получил исполняемый файл. Проверил, как работает полученный исполняемый файл.


```
[paosipov@VirtualBox lab05]$ nasm -f elf lab05-1.asm
[paosipov@VirtualBox lab05]$ ld -m elf_i386 lab05-1.o -o lab05-1
[paosipov@VirtualBox lab05]$ ./lab05-1
Введите строку:
Pavel
[paosipov@VirtualBox lab05]$
```

Рис. 2.4: Запуск программы lab05-1.asm

5. Скачал файл in_out.asm и добавил его в рабочий каталог. Скопировал содержимое lab05-1.asm в новый файл lab05-2.asm.

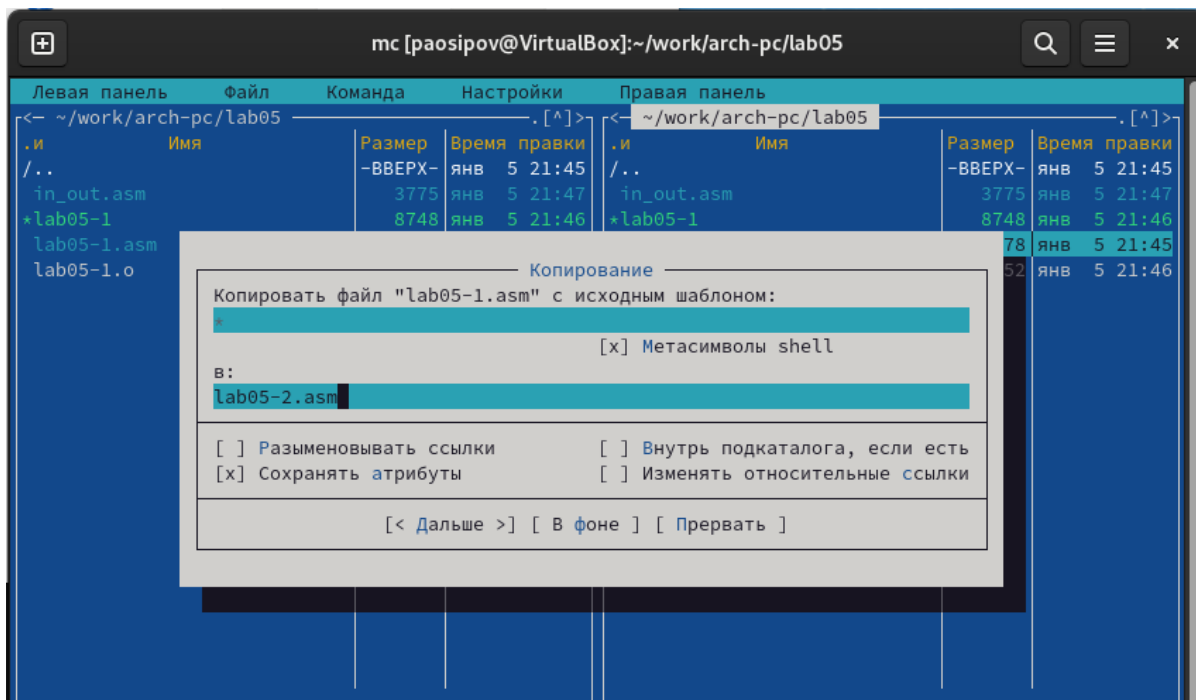
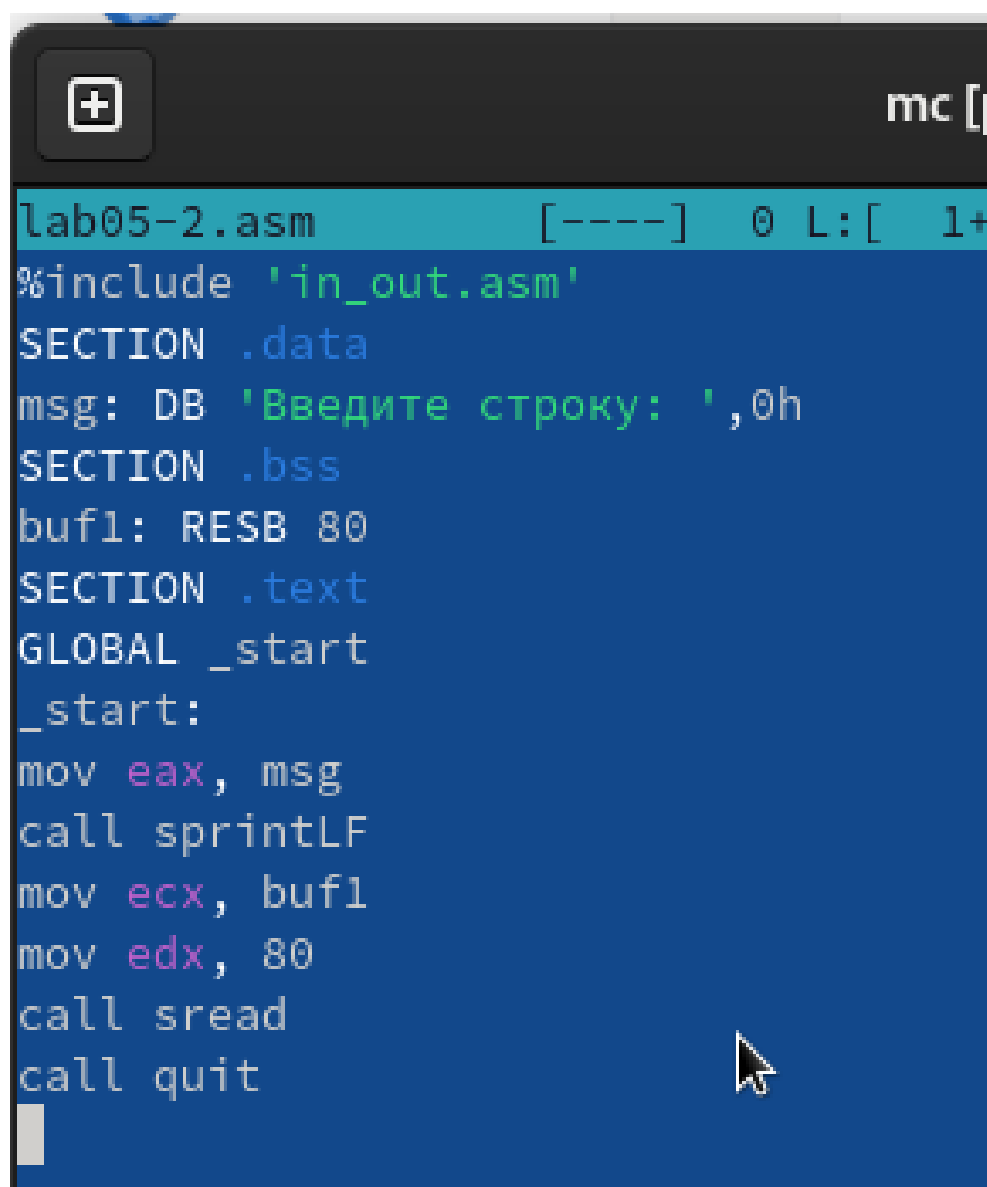


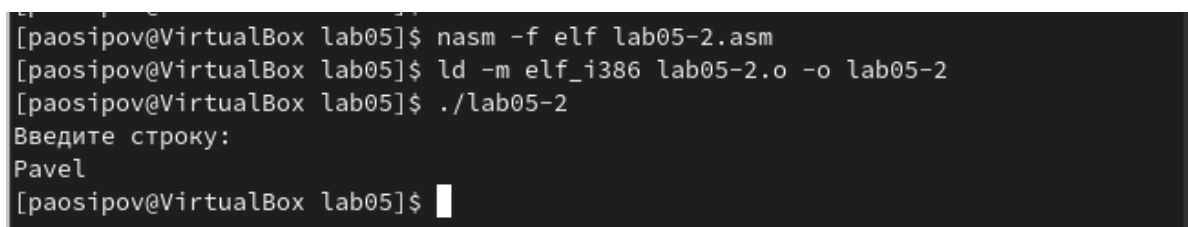
Рис. 2.5: Копирование файла

6. В файле lab05-2.asm написал код программы. Скомпилировал программу и проверил ее запуск.



```
lab05-2.asm [-----] 0 L: [ 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
```

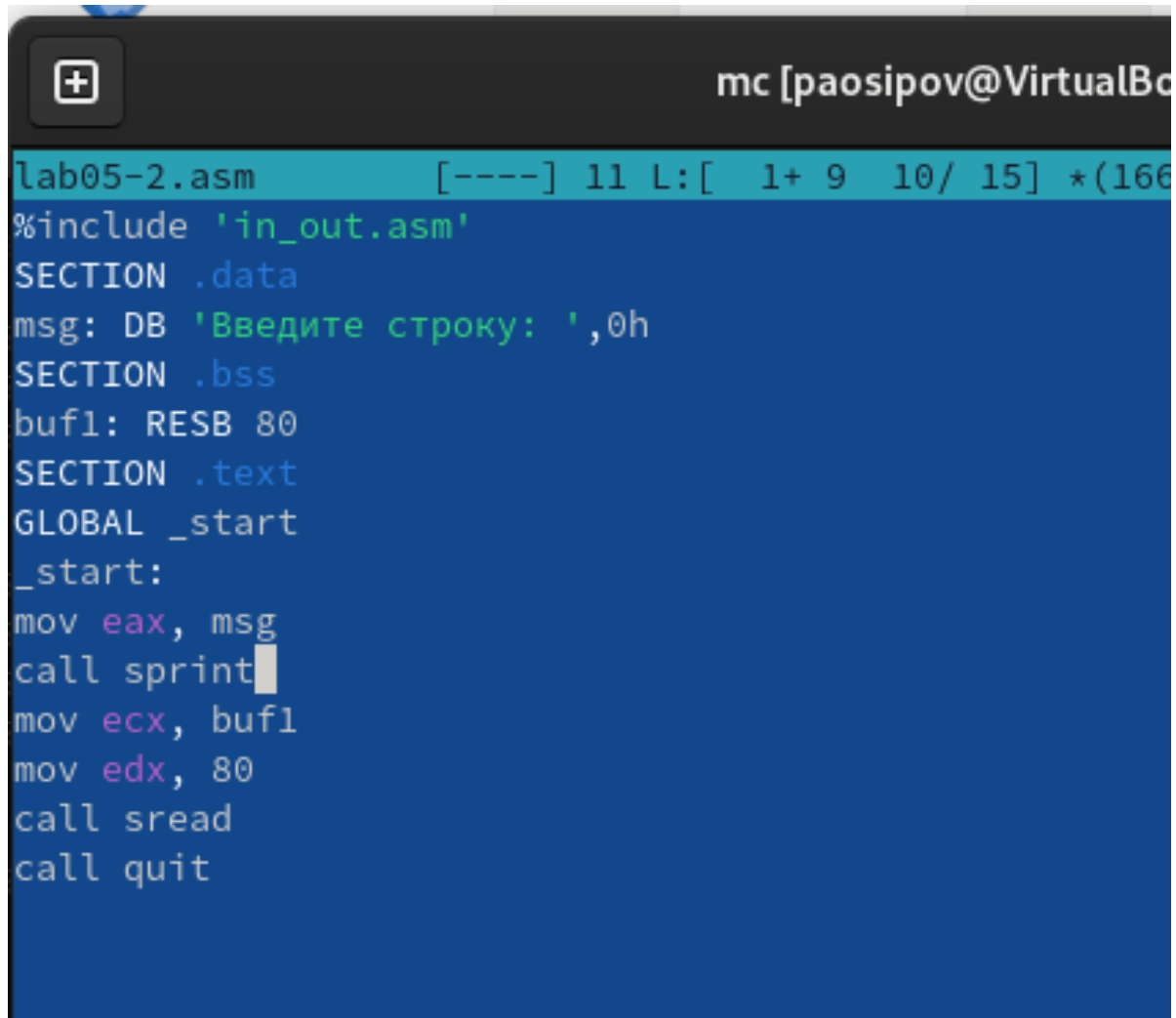
Рис. 2.6: Программа lab05-2.asm



```
[paosipov@VirtualBox lab05]$ nasm -f elf lab05-2.asm
[paosipov@VirtualBox lab05]$ ld -m elf_i386 lab05-2.o -o lab05-2
[paosipov@VirtualBox lab05]$ ./lab05-2
Введите строку:
Pavel
[paosipov@VirtualBox lab05]$
```

Рис. 2.7: Запуск программы lab05-2.asm

7. В файле lab05-2.asm заменил вызов подпрограммы sprintLF на sprint, чтобы вывод строки происходил без перехода на следующую строку. Пересобрал исполняемый файл.



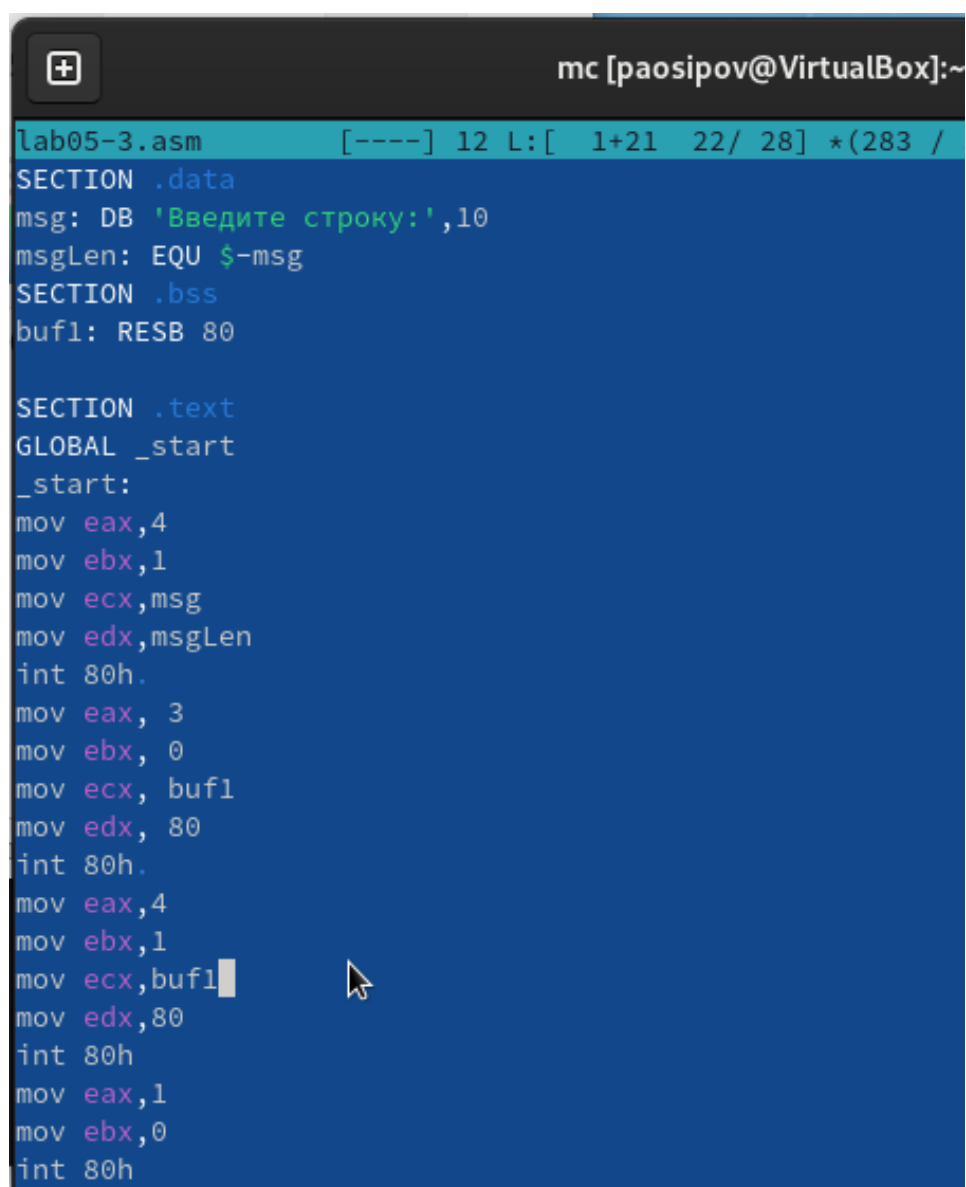
```
lab05-2.asm [----] 11 L: [ 1+ 9 10/ 15] *(166
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку: ',0h
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprint
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рис. 2.8: Программа lab05-2.asm

```
[paosipov@VirtualBox lab05]$ nasm -f elf lab05-2.asm
[paosipov@VirtualBox lab05]$ ld -m elf_i386 lab05-2.o -o lab05-2
[paosipov@VirtualBox lab05]$ ./lab05-2
Введите строку:
Pavel
[paosipov@VirtualBox lab05]$ nasm -f elf lab05-2.asm
[paosipov@VirtualBox lab05]$ ld -m elf_i386 lab05-2.o -o lab05-2
[paosipov@VirtualBox lab05]$ ./lab05-2
Введите строку: Pavel
[paosipov@VirtualBox lab05]$
```

Рис. 2.9: Запуск программы lab05-2.asm

8. 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.



```
mc [paosipov@VirtualBox]:~
lab05-3.asm [----] 12 L:[ 1+21 22/ 28] *(283 /
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

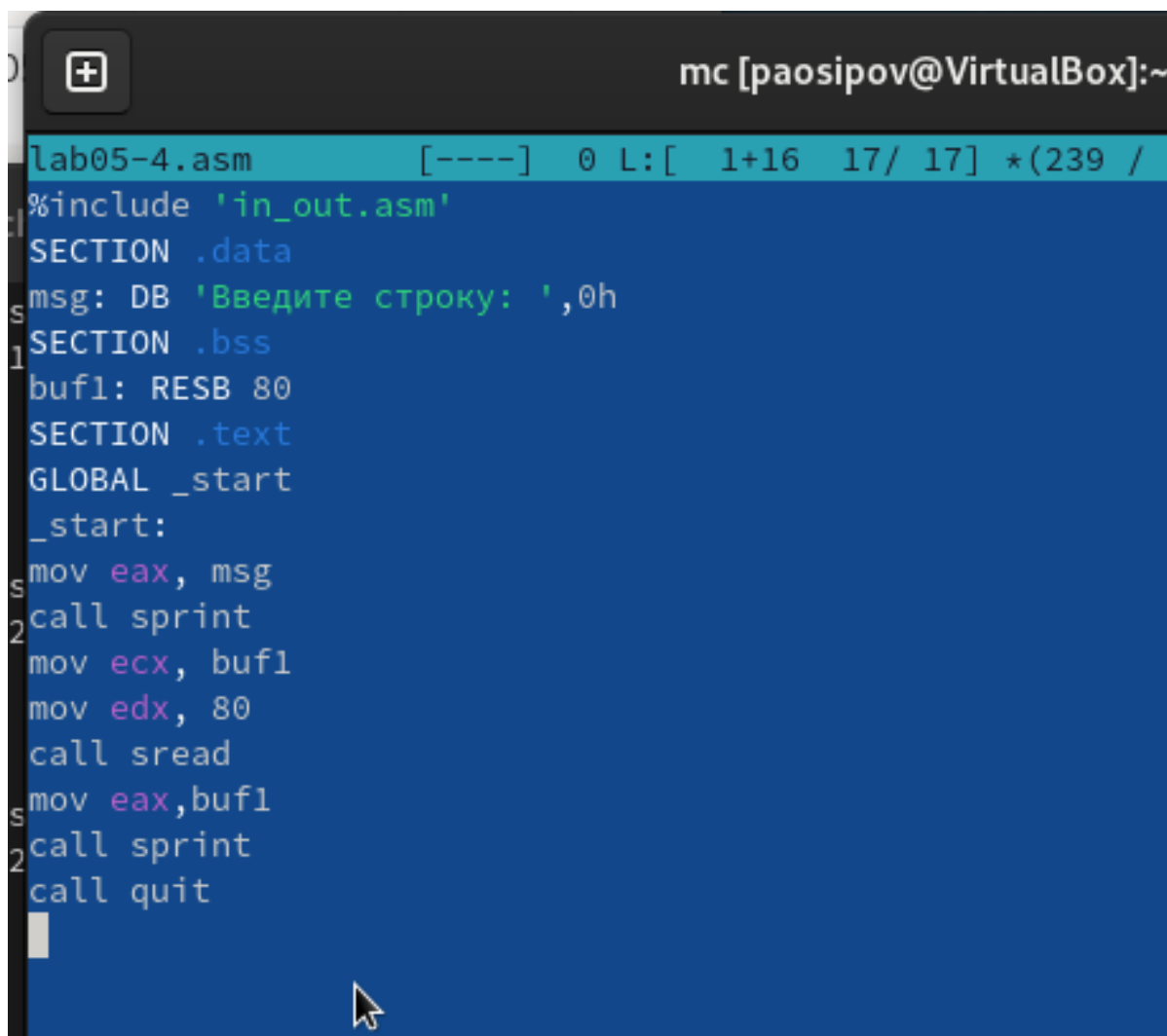
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h.
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h.
mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,80
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рис. 2.10: Программа lab05-3.asm

```
[paosipov@VirtualBox lab05]$  
[paosipov@VirtualBox lab05]$ nasm -f elf lab05-3.asm  
[paosipov@VirtualBox lab05]$ ld -m elf_i386 lab05-3.o -o lab05-3  
[paosipov@VirtualBox lab05]$ ./lab05-3  
Введите строку:  
Pavel  
Pavel  
[paosipov@VirtualBox lab05]$
```

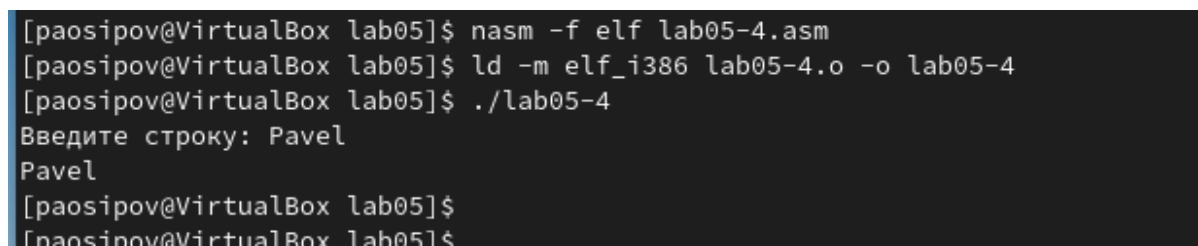
Рис. 2.11: Запуск программы lab05-3.asm

9. Скопировал программу lab05-2.asm и сделал аналогичные изменения, но теперь использовал возможности из файла in_out.asm для ввода-вывода строк.



```
lab05-4.asm [-----] 0 L:[ 1+16 17/ 17] *(239 /
#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. 2.12: Программа lab05-4.asm



```
[paosipov@VirtualBox lab05]$ nasm -f elf lab05-4.asm
[paosipov@VirtualBox lab05]$ ld -m elf_i386 lab05-4.o -o lab05-4
[paosipov@VirtualBox lab05]$ ./lab05-4
Введите строку: Pavel
Pavel
[paosipov@VirtualBox lab05]$
[paosipov@VirtualBox lab05]$
```

Рис. 2.13: Запуск программы lab05-4.asm

3 Выводы

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