

Отчёт по лабораторной работе 5

Архитектура компьютеров и операционные системы

Ханеков Максат НКА-06-23

Содержание

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

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	10
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])

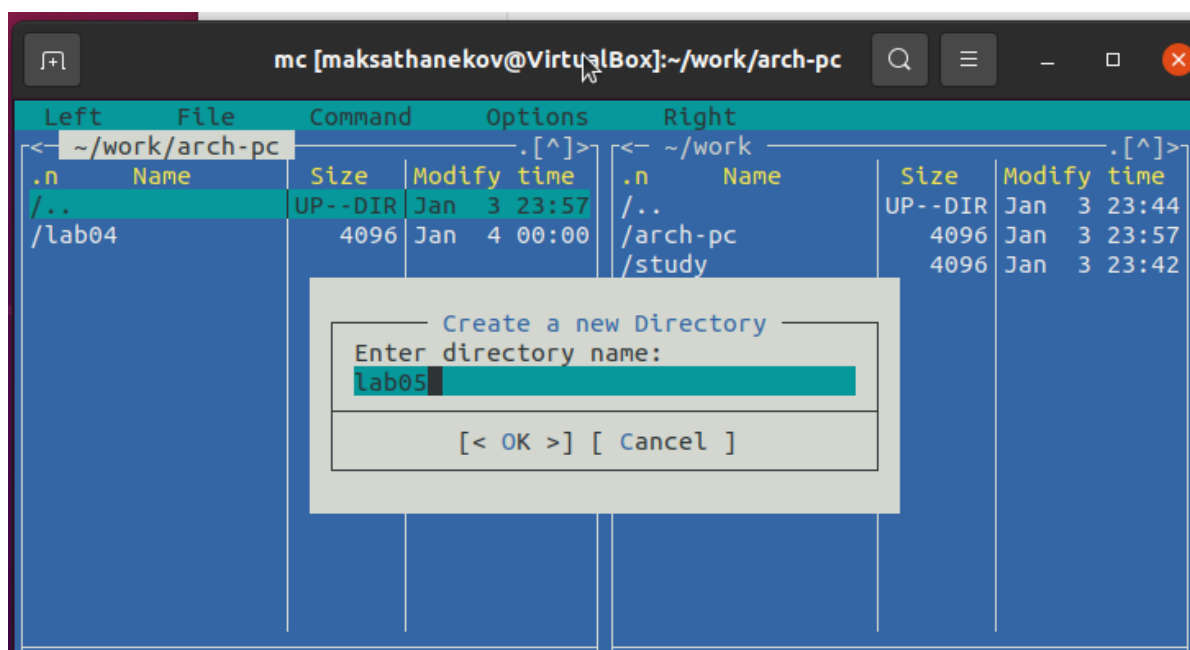


Рис. 2.1: Создание каталога

Создал файл lab05-1.asm (рис. [2.2])

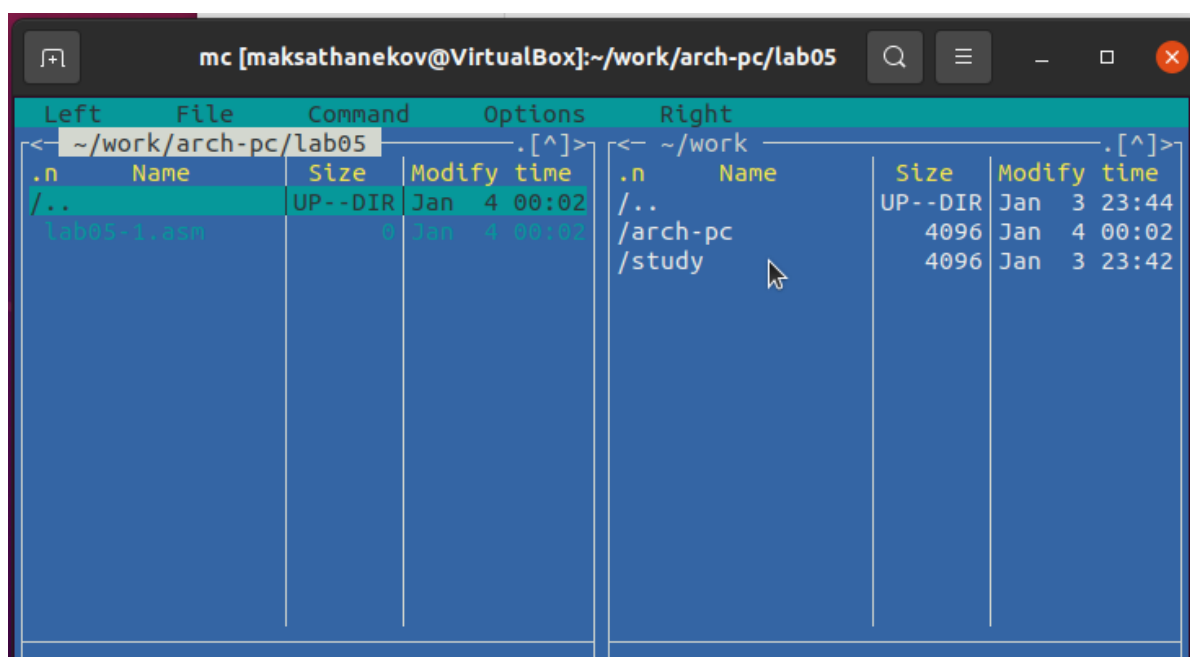
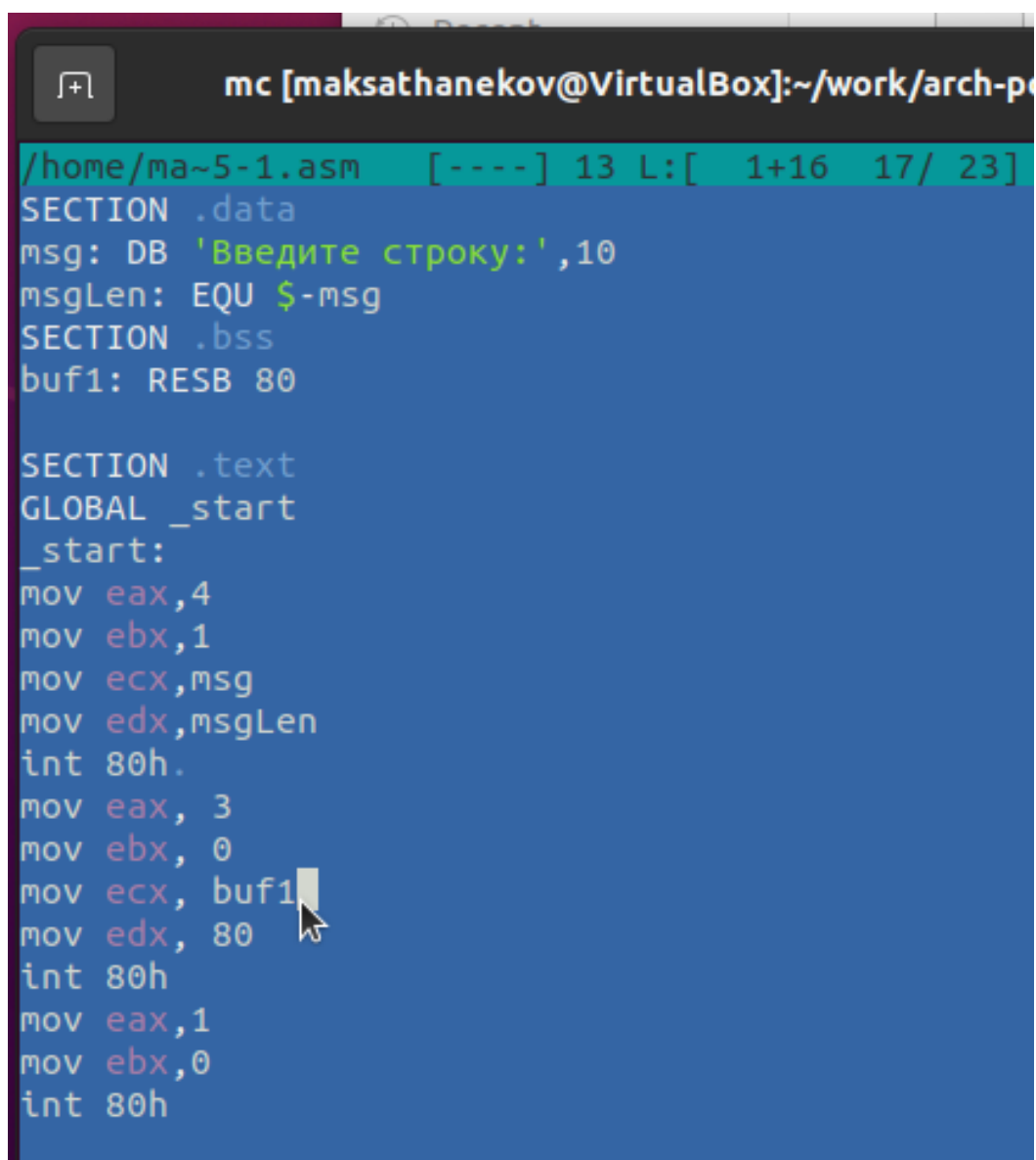


Рис. 2.2: Создание файла lab05-1.asm

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

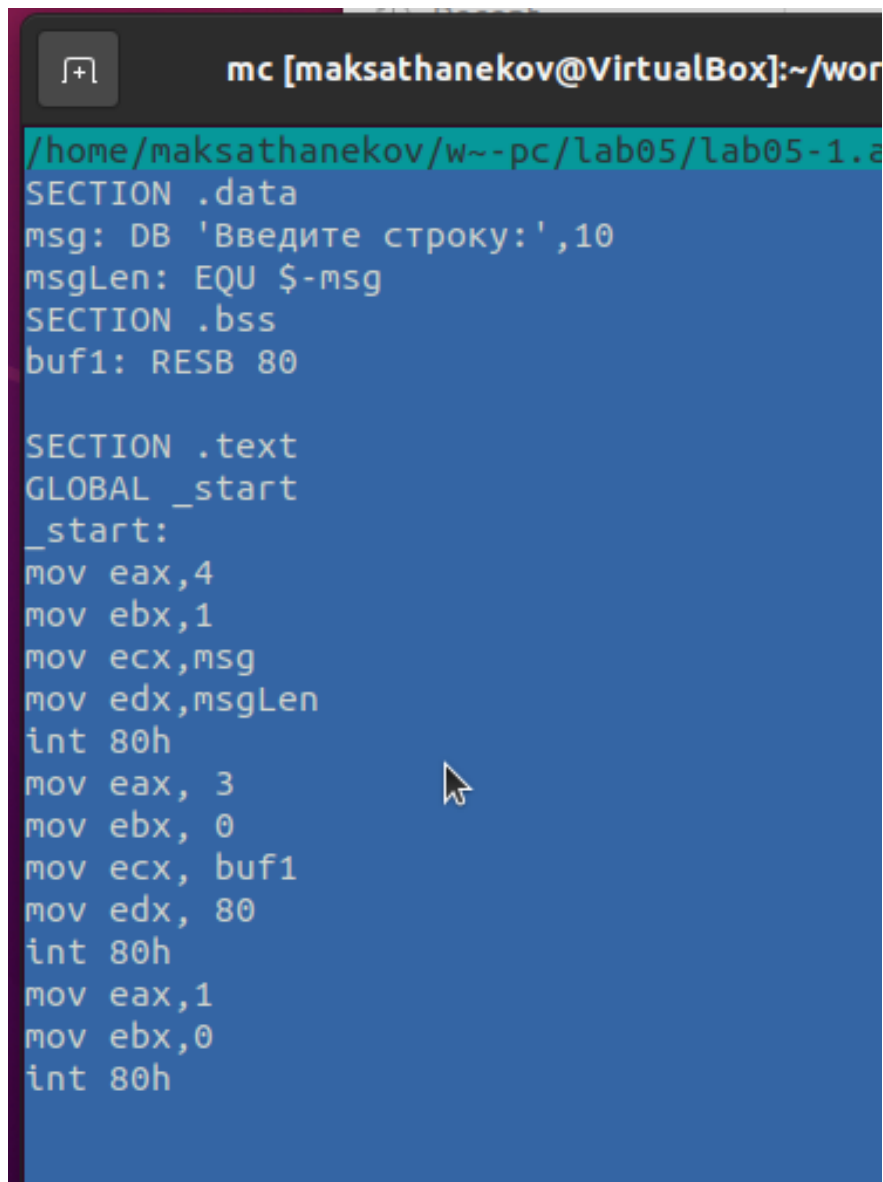


```
mc [maksathanekov@VirtualBox]:~/work/arch-pc
/home/ma~5-1.asm [----] 13 L:[ 1+16 17/ 23]
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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. 2.3: Программа в файле lab05-1.asm

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



```
mc [maksathanekov@VirtualBox]:~/wor
/home/maksathanekov/w~-pc/lab05/lab05-1.a
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])

```

maksathanekov@VirtualBox:~/work/arch-pc/lab05$ nasm -f elf lab05-1.asm
maksathanekov@VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-1.o -o lab05-1
maksathanekov@VirtualBox:~/work/arch-pc/lab05$ ./lab05-1
Введите строку:
Maksat
maksathanekov@VirtualBox:~/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])

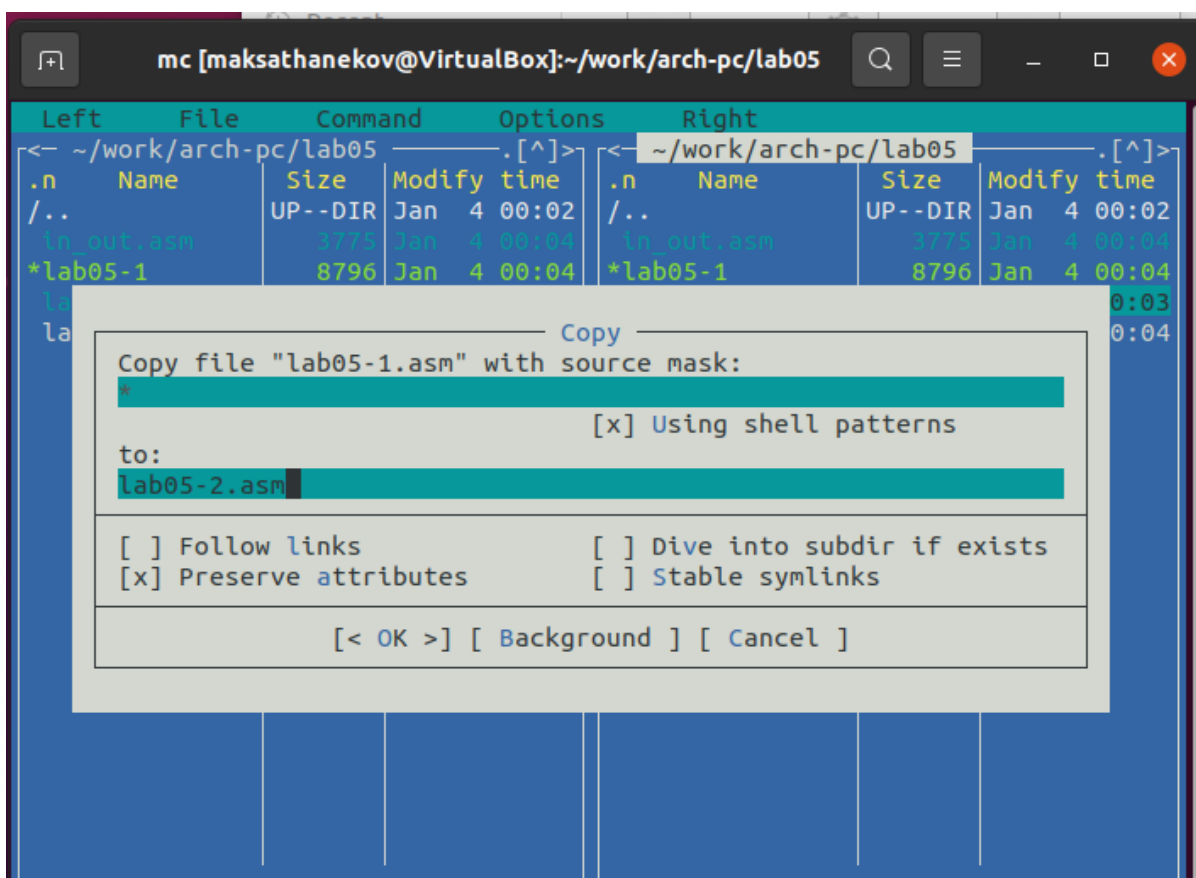
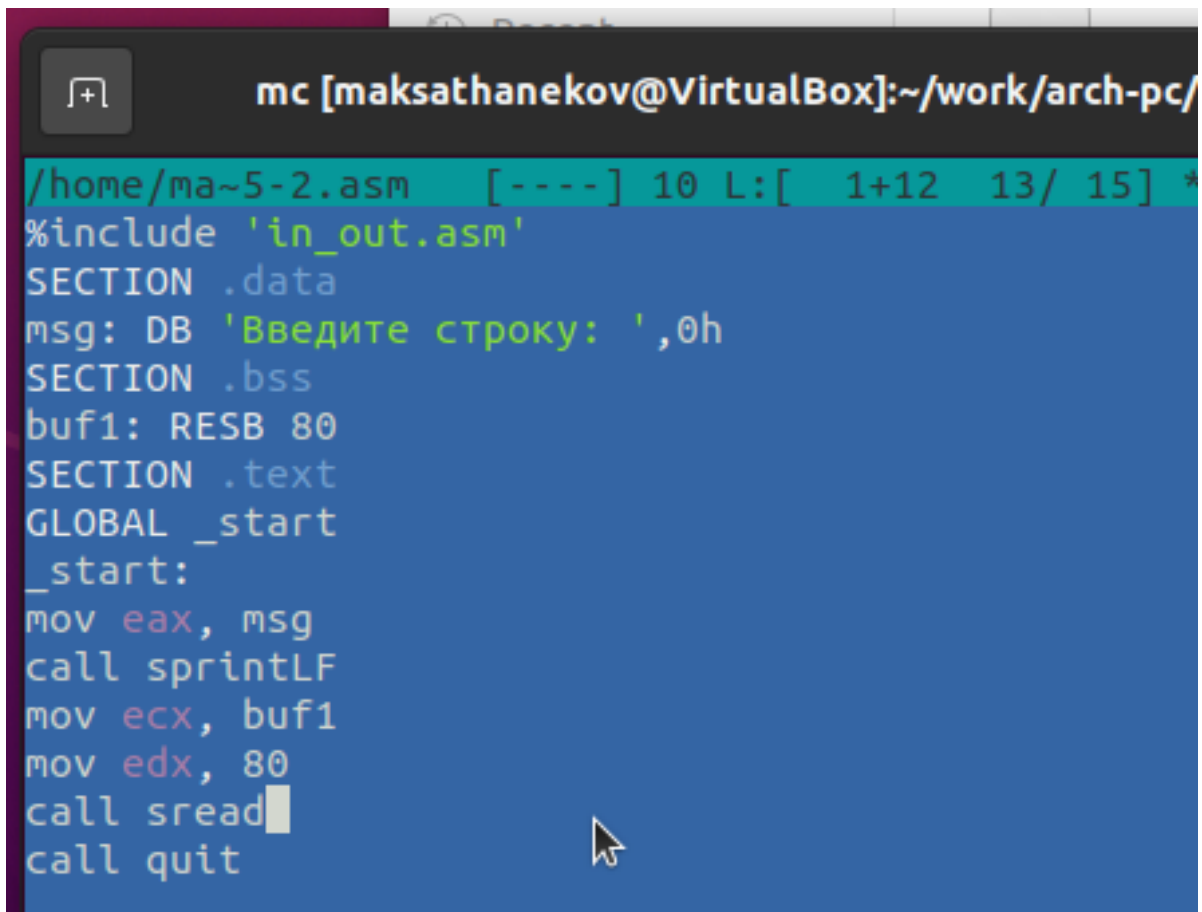


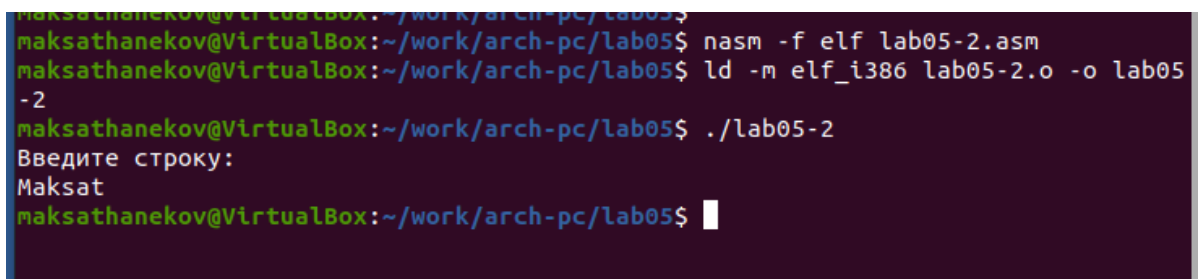
Рис. 2.6: Копирование файла

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



```
mc [maksathanekov@VirtualBox]:~/work/arch-pc/  
/home/ma~5-2.asm  [ - - - ] 10 L:[ 1+12 13/ 15] *  
%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 sprintf  
mov ecx, buf1  
mov edx, 80  
call sread  
call quit
```

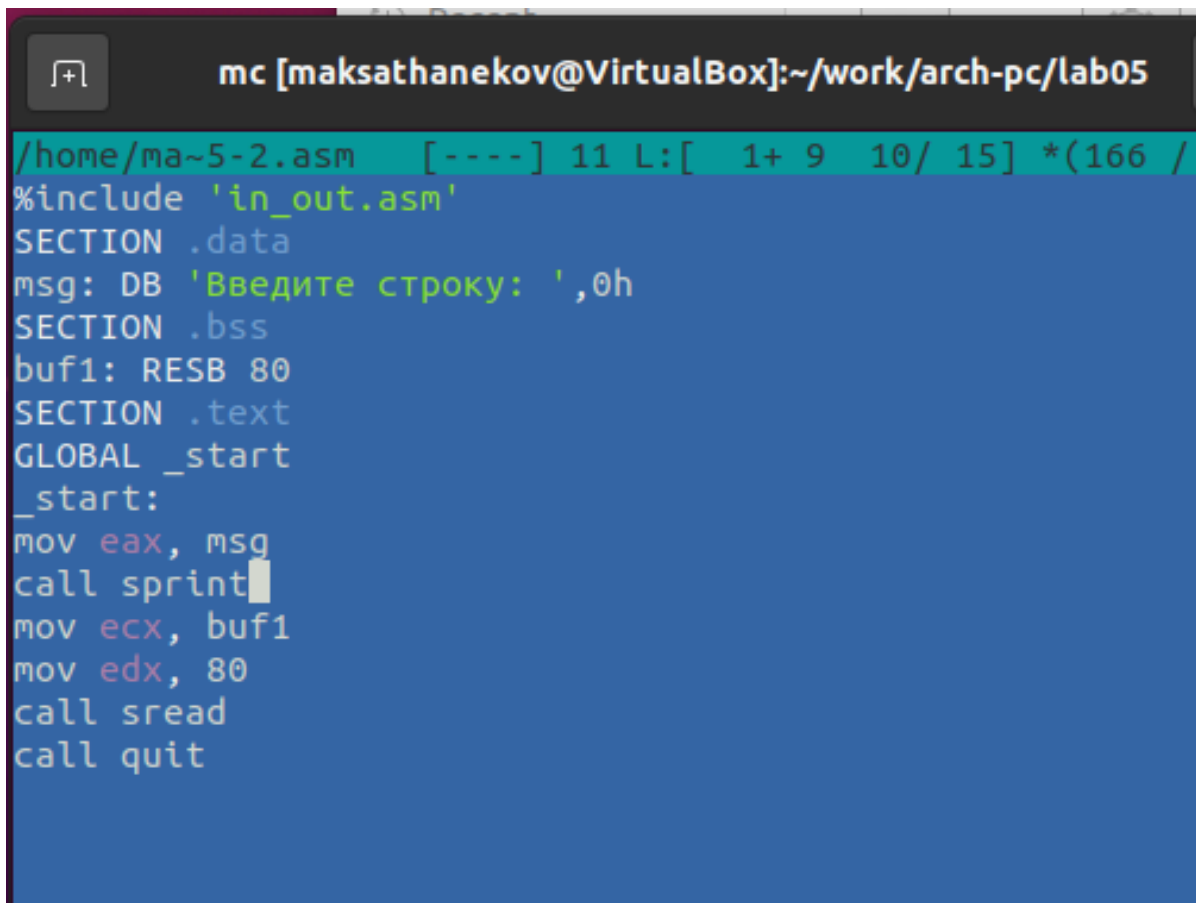
Рис. 2.7: Программа в файле lab05-2.asm



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

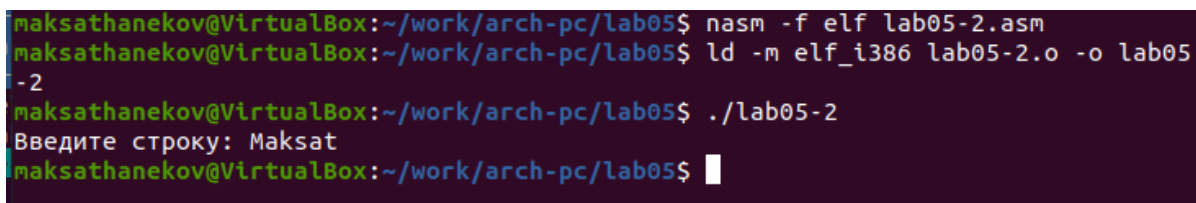
Рис. 2.8: Запуск программы lab05-2.asm

В файле lab5-2.asm я заменил подпрограмму sprintf 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 [maksathanekov@VirtualBox]:~/work/arch-pc/lab05
/home/ma~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
```

Рис. 2.9: Программа в файле lab05-2.asm



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

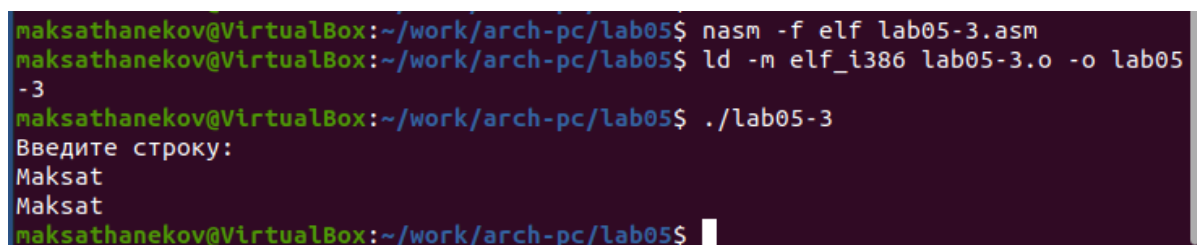
Рис. 2.10: Запуск программы lab05-2.asm

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])



```
mc [maksathanekov@VirtualBox]:~/work/arch-  
/home/ma~5-3.asm [----] 11 L:[ 1+17 18/ 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
```

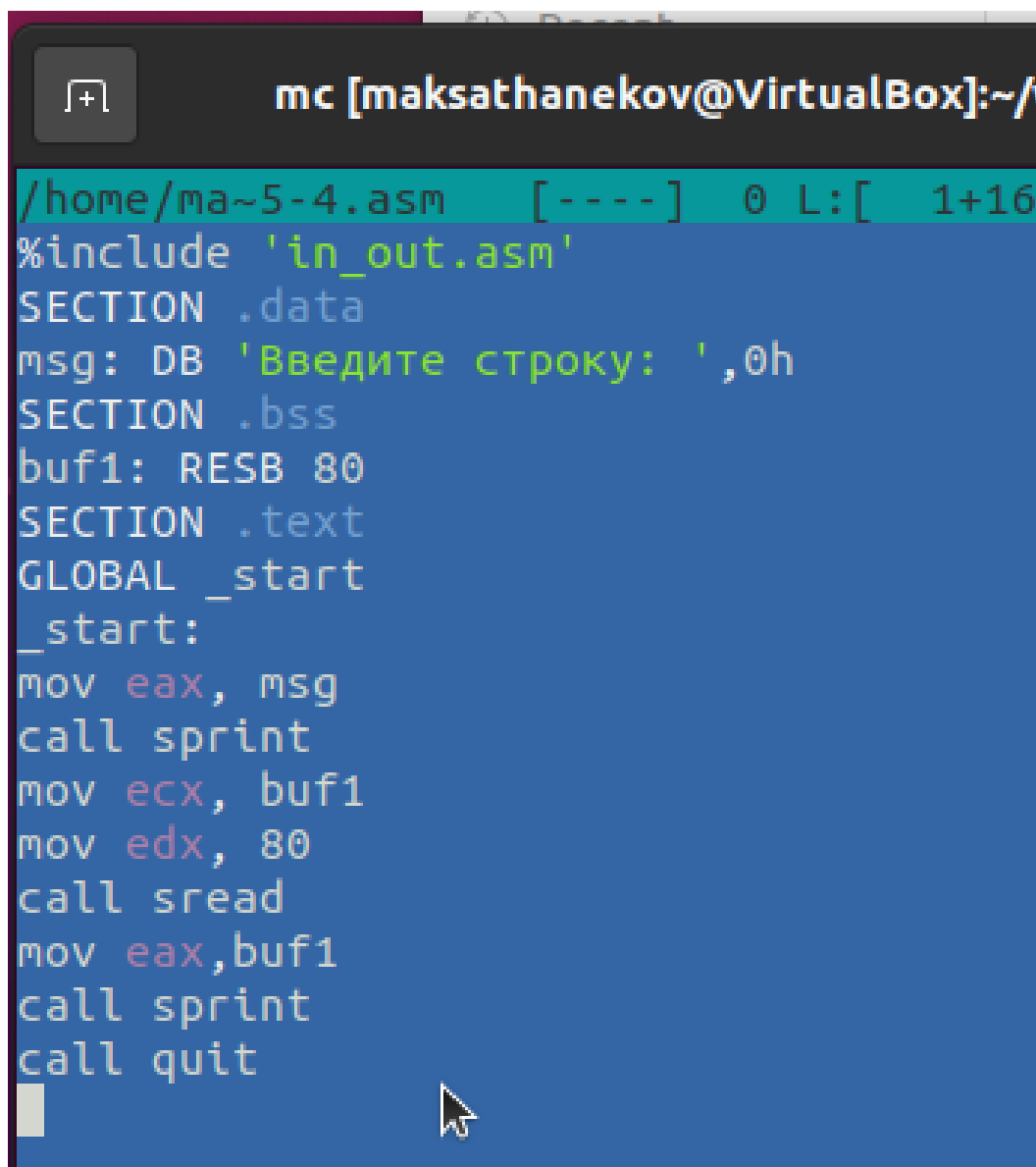
Рис. 2.11: Программа в файле lab05-3.asm



```
maksathanekov@VirtualBox:~/work/arch-pc/lab05$ nasm -f elf lab05-3.asm  
maksathanekov@VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3  
maksathanekov@VirtualBox:~/work/arch-pc/lab05$ ./lab05-3  
Введите строку:  
Maksat  
Maksat  
maksathanekov@VirtualBox:~/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 'mc [maksathanekov@VirtualBox]:~/'. The terminal content shows assembly code for a program. The code includes a data section with a message, a bss section with a buffer, and a text section with the main logic. The logic involves printing the message, reading input into the buffer, and then printing the buffer's contents before exiting.

```
mc [maksathanekov@VirtualBox]:~/
/home/ma~5-4.asm [ - - - - ] 0 L:[ 1+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. 2.13: Программа в файле lab05-4.asm

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