

Отчёт по лабораторной работе 5

Архитектура компьютеров и операционные системы

Сатторов Икромджон Абдувохидович

Содержание

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

Список иллюстраций

2.1	Создание каталога в Midnight Commander	6
2.2	Создание файла lab05-1.asm в Midnight Commander	7
2.3	Редактирование файла lab05-1.asm	8
2.4	Просмотр файла lab05-1.asm	9
2.5	Компиляция и проверка программы lab05-1.asm	9
2.6	Копирование файла	10
2.7	Копирование файла	11
2.8	Редактирование файла lab05-2.asm	12
2.9	Компиляция и проверка программы lab05-2.asm	12
2.10	Редактирование файла lab05-2.asm	13
2.11	Компиляция и проверка программы lab05-2.asm	13
2.12	Редактирование файла lab05-3.asm	14
2.13	Компиляция и проверка программы lab05-3.asm	14
2.14	Редактирование файла lab05-4.asm	15
2.15	Компиляция и проверка 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

Я открыл Midnight Commander и перешел в каталог ~/work/arch-rc. Затем я создал каталог с именем lab05.

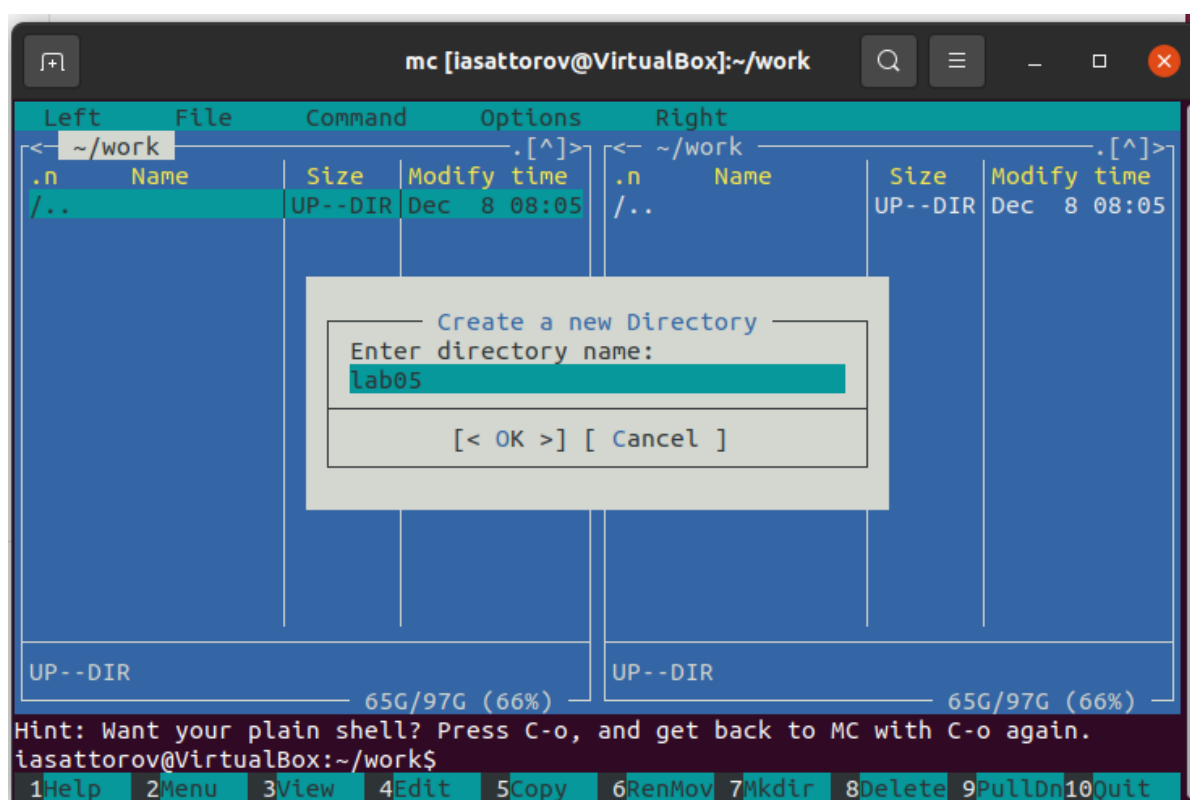


Рис. 2.1: Создание каталога в Midnight Commander

Создал файл с именем lab05-1.asm.

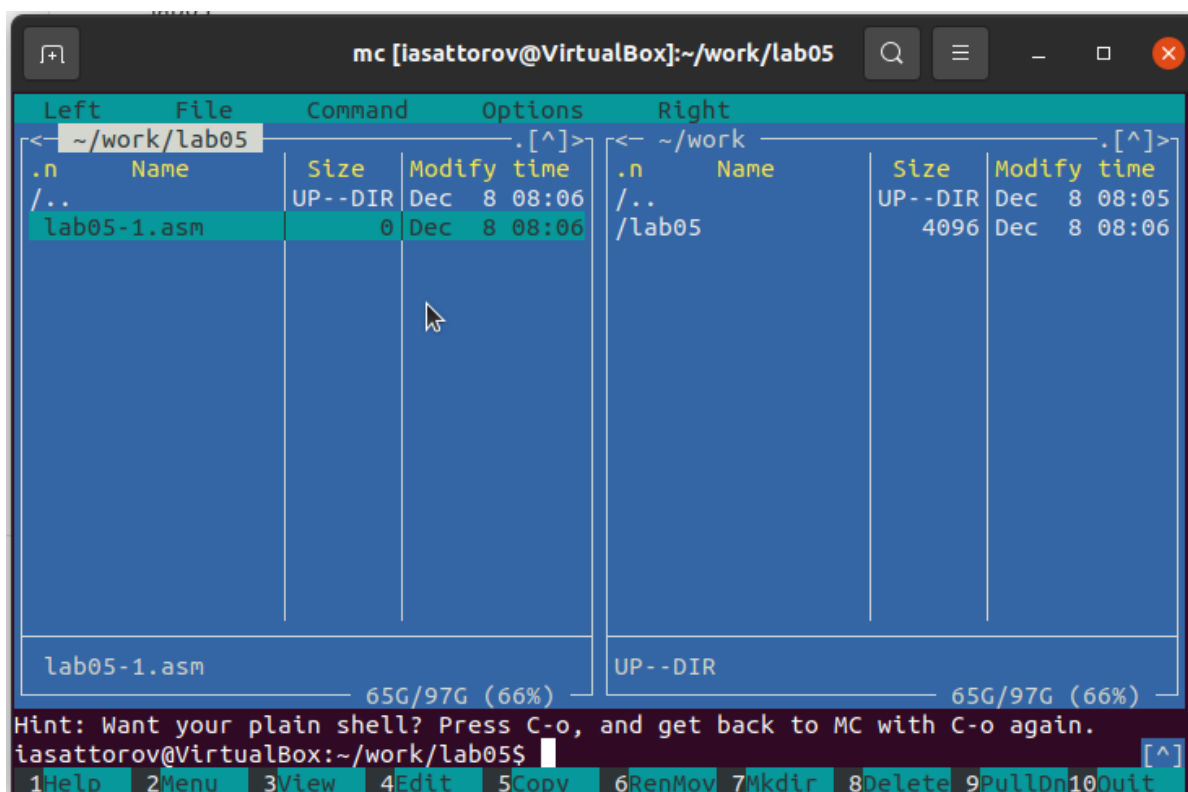
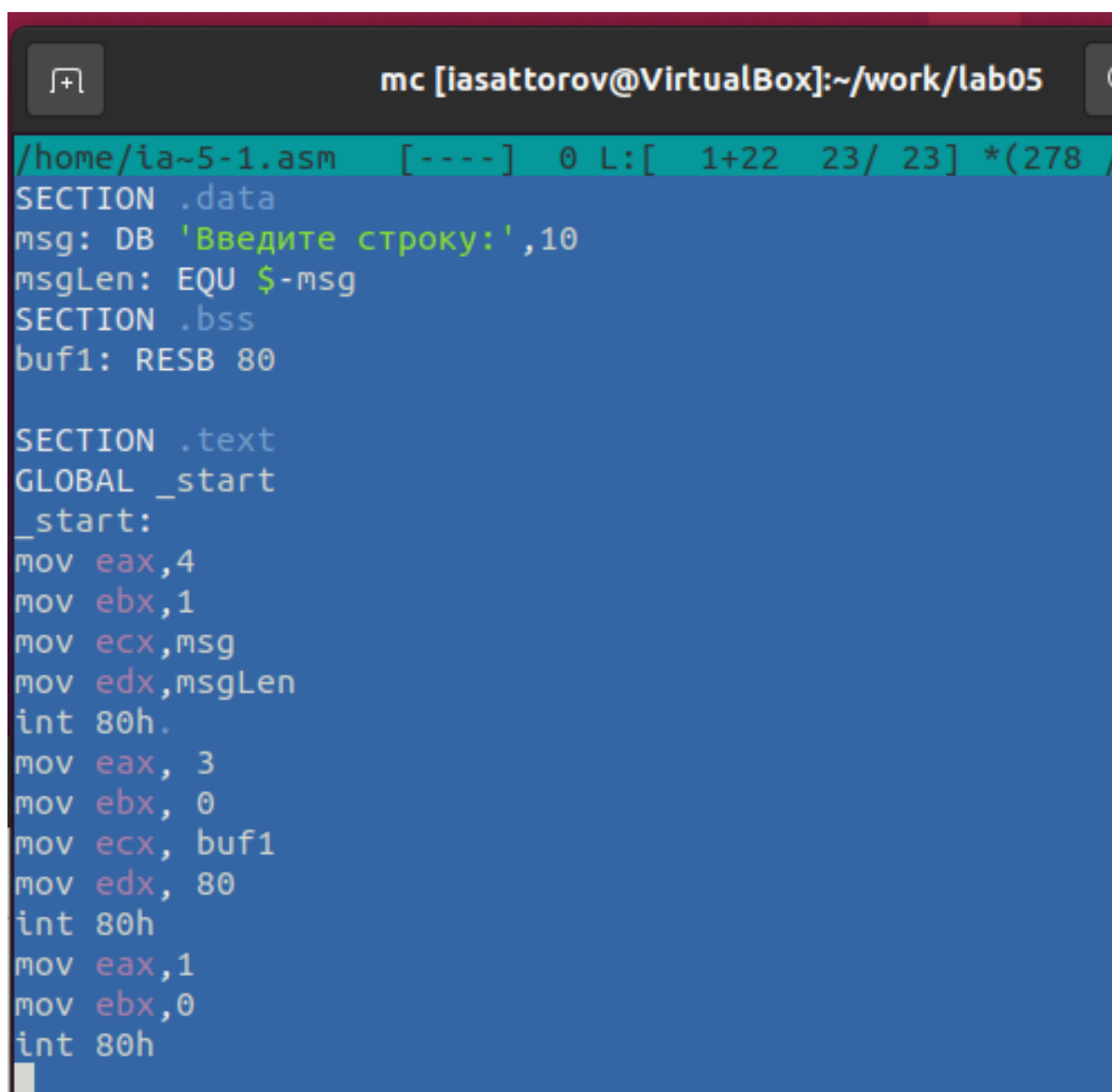


Рис. 2.2: Создание файла lab05-1.asm в Midnight Commander

Открыл файл для редактирования и написал код программы.

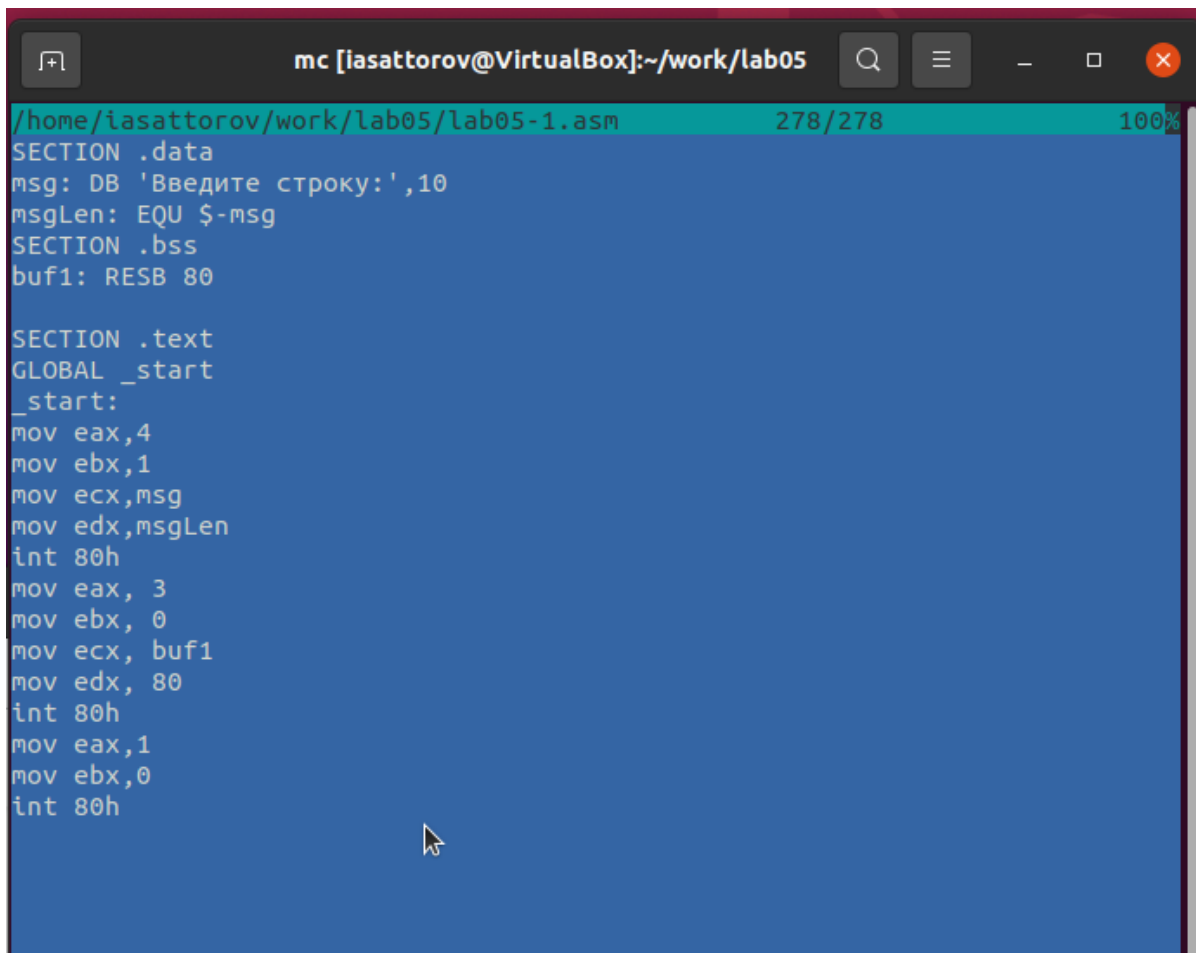
A screenshot of a terminal window with a dark background. The title bar at the top shows 'mc [iasattorov@VirtualBox]:~/work/lab05'. The terminal displays assembly code for a file named 'lab05-1.asm'. The code is color-coded: section names are in white, data/bss declarations in green, and instructions in pink. The code defines a data section with a message 'Введите строку:', a bss section with a buffer 'buf1', and a text section with assembly instructions for inputting a string. The instructions use registers like eax, ebx, ecx, and edx, and include system calls like 'int 80h'.

```
/home/ia~5-1.asm [----] 0 L:[ 1+22 23/ 23] *(278 /
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: Редактирование файла lab05-1.asm

Открыл файл для просмотра и убедился, что он содержит написанный код.

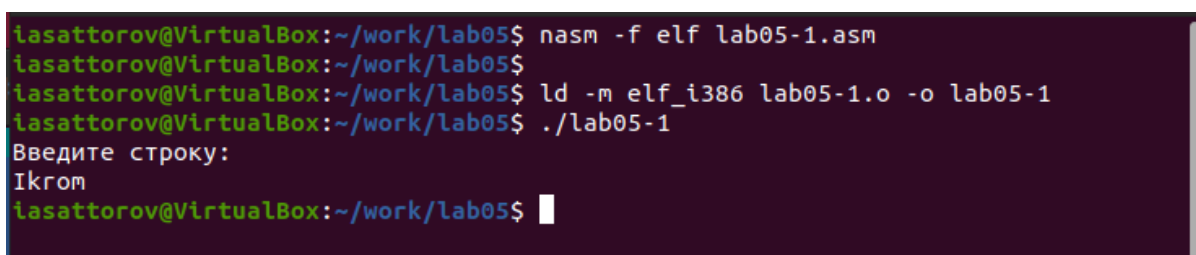


```
mc [iasattorov@VirtualBox]:~/work/lab05
/home/iasattorov/work/lab05/lab05-1.asm 278/278 100%
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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.



```
iasattorov@VirtualBox:~/work/lab05$ nasm -f elf lab05-1.asm
iasattorov@VirtualBox:~/work/lab05$
iasattorov@VirtualBox:~/work/lab05$ ld -m elf_i386 lab05-1.o -o lab05-1
iasattorov@VirtualBox:~/work/lab05$ ./lab05-1
Введите строку:
Ikrom
iasattorov@VirtualBox:~/work/lab05$
```

Рис. 2.5: Компиляция и проверка программы lab05-1.asm

Скачал файл in_out.asm и добавил его в рабочий каталог. Затем скопировал содержимое файла lab05-1.asm в файл lab05-2.asm.

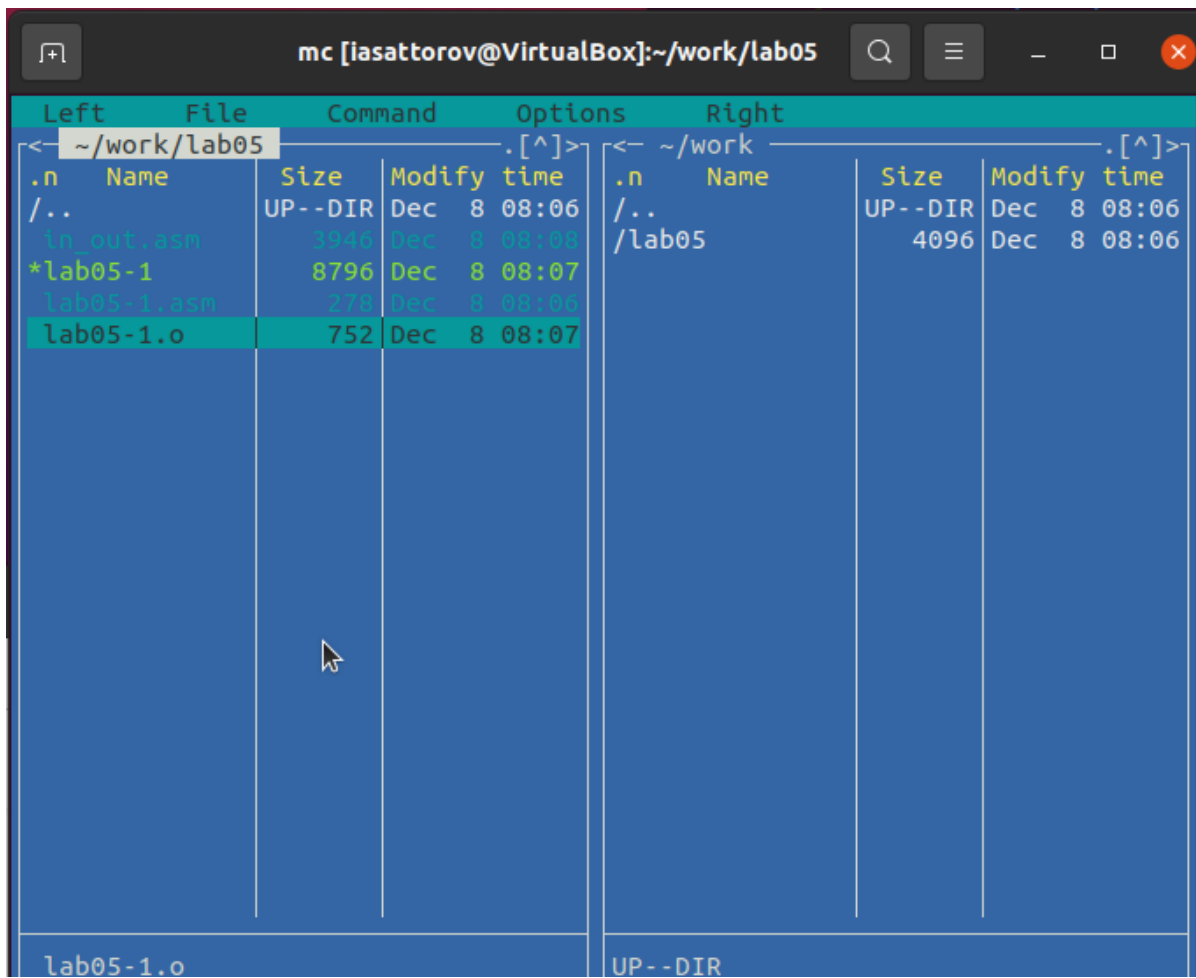


Рис. 2.6: Копирование файла

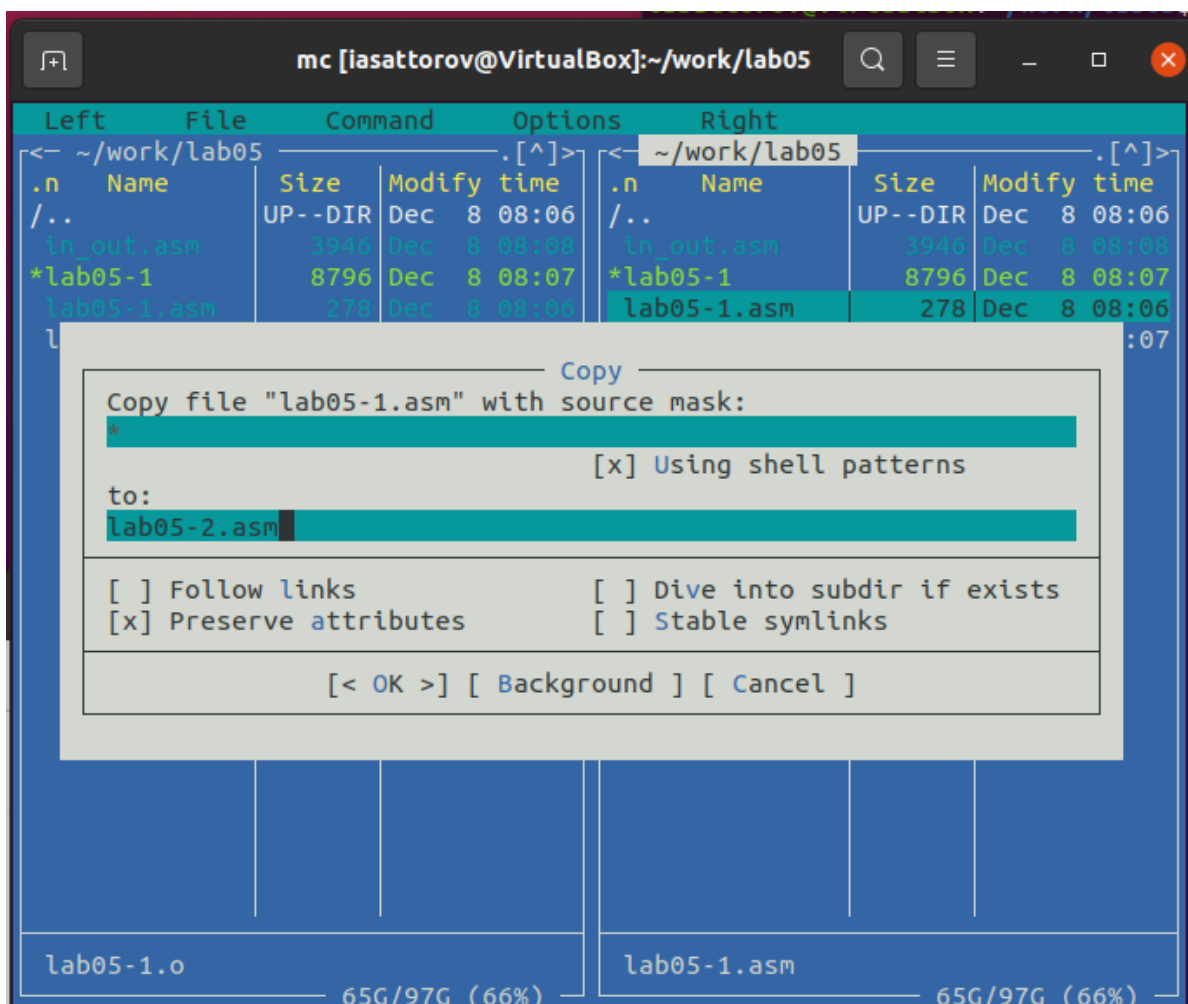
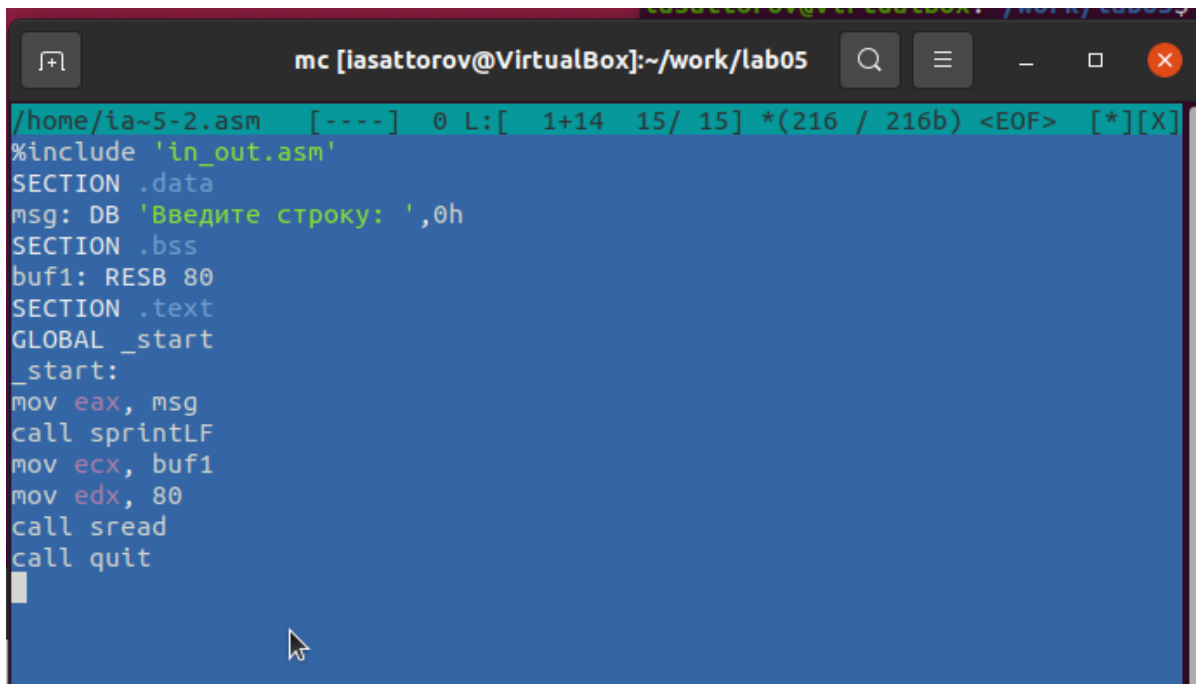


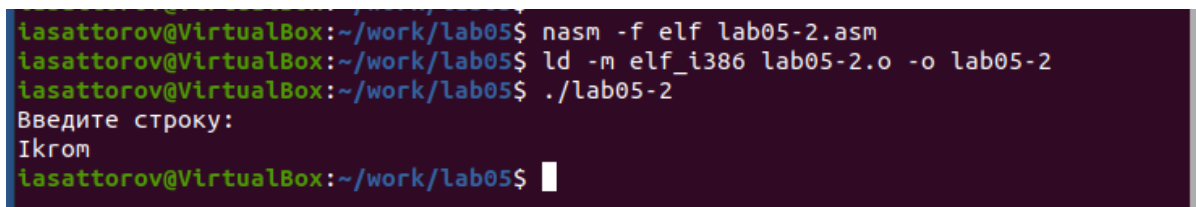
Рис. 2.7: Копирование файла

Написал код программы lab05-2.asm. Затем скомпилировал программу и проверил ее запуск.



```
mc [iasattorov@VirtualBox]:~/work/lab05
/home/ia~5-2.asm [----] 0 L: [ 1+14 15/ 15] *(216 / 216b) <EOF> [*][X]
#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
```

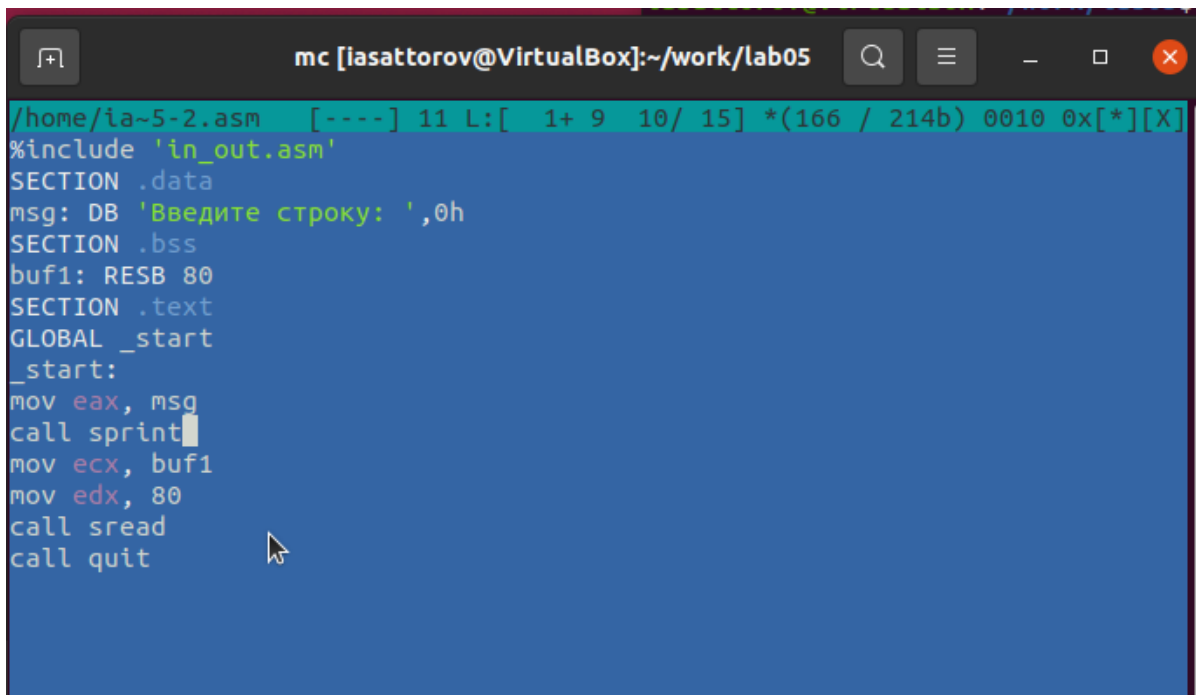
Рис. 2.8: Редактирование файла lab05-2.asm



```
iasattorov@VirtualBox:~/work/lab05$ nasm -f elf lab05-2.asm
iasattorov@VirtualBox:~/work/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
iasattorov@VirtualBox:~/work/lab05$ ./lab05-2
Введите строку:
Ikrom
iasattorov@VirtualBox:~/work/lab05$
```

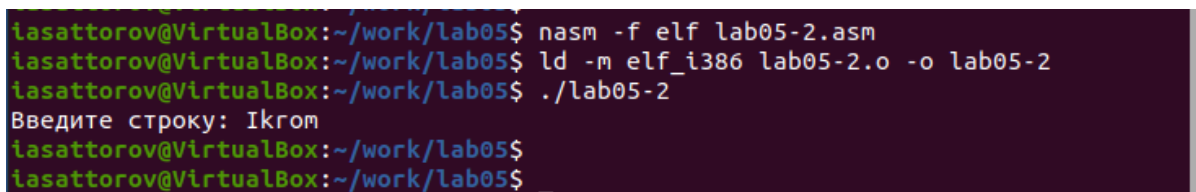
Рис. 2.9: Компиляция и проверка программы lab05-2.asm

В файле lab5-2.asm я заменил вызов подпрограммы sprintLF на sprint. Затем я снова собрал исполняемый файл. Теперь после вывода строки она не завершается символом перехода на новую строку.



```
mc [iasattorov@VirtualBox]:~/work/lab05
/home/ia~5-2.asm [----] 11 L:[ 1+ 9 10/ 15] *(166 / 214b) 0010 0x[*][X]
%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
```

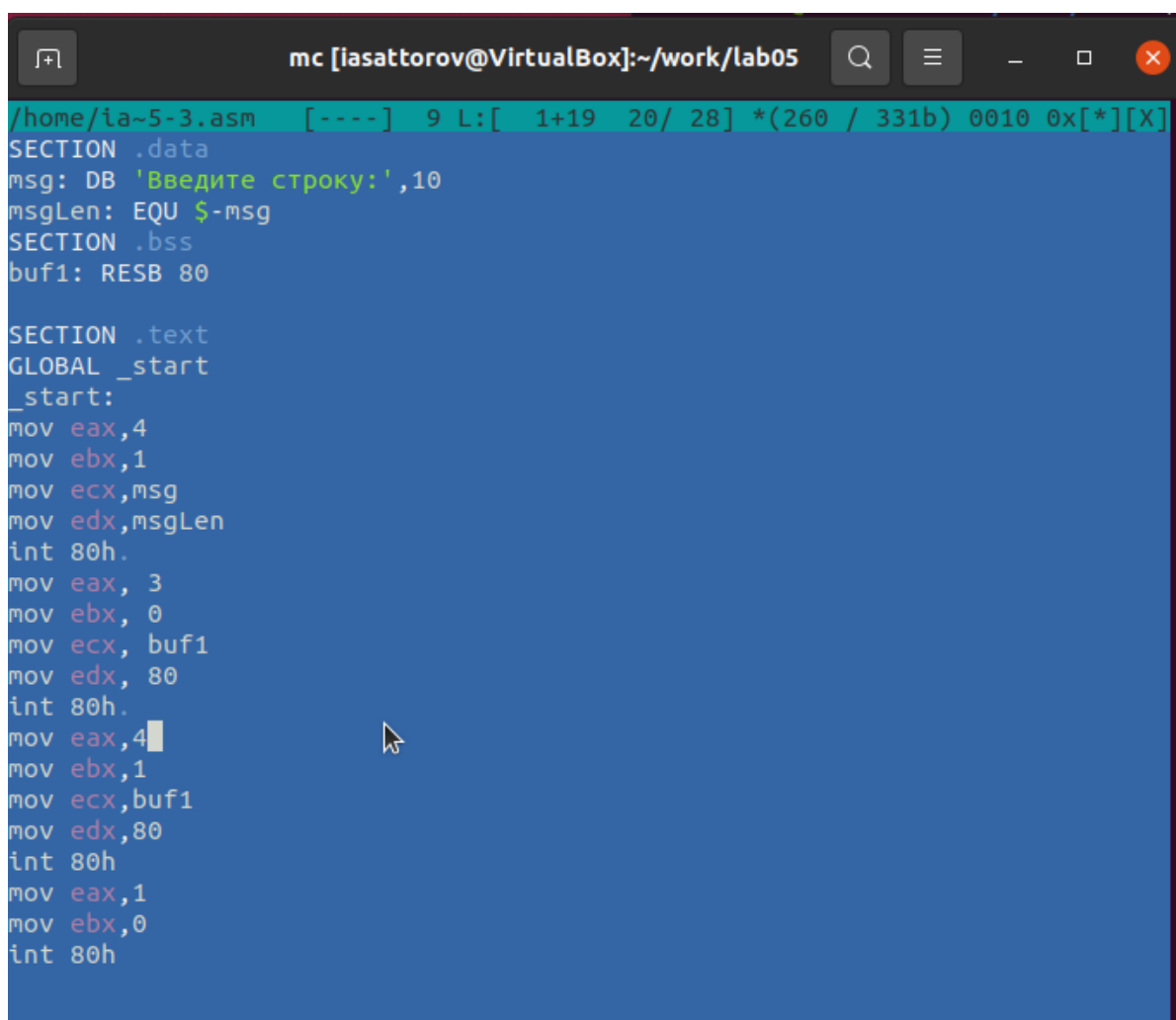
Рис. 2.10: Редактирование файла lab05-2.asm



```
iasattorov@VirtualBox:~/work/lab05$ nasm -f elf lab05-2.asm
iasattorov@VirtualBox:~/work/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
iasattorov@VirtualBox:~/work/lab05$ ./lab05-2
Введите строку: Ikrom
iasattorov@VirtualBox:~/work/lab05$
iasattorov@VirtualBox:~/work/lab05$
```

Рис. 2.11: Компиляция и проверка программы lab05-2.asm

Также я скопировал программу lab05-1.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.

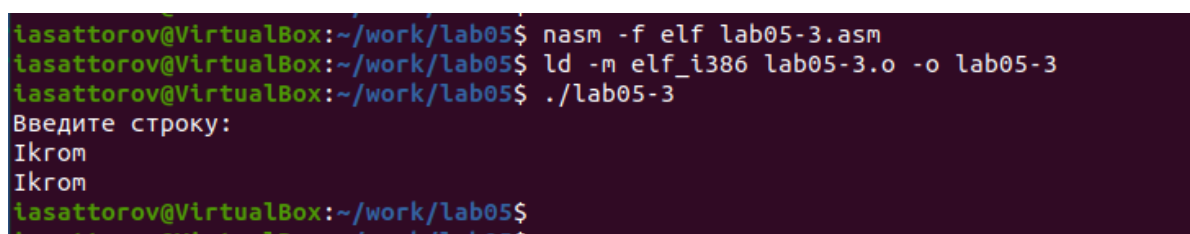


The screenshot shows a text editor window titled 'mc [iasattorov@VirtualBox]:~/work/lab05'. The editor displays the assembly file 'lab05-3.asm'. The code is as follows:

```
/home/ia~5-3.asm [---] 9 L:[ 1+19 20/ 28] *(260 / 331b) 0010 0x[*][X]
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. 2.12: Редактирование файла lab05-3.asm

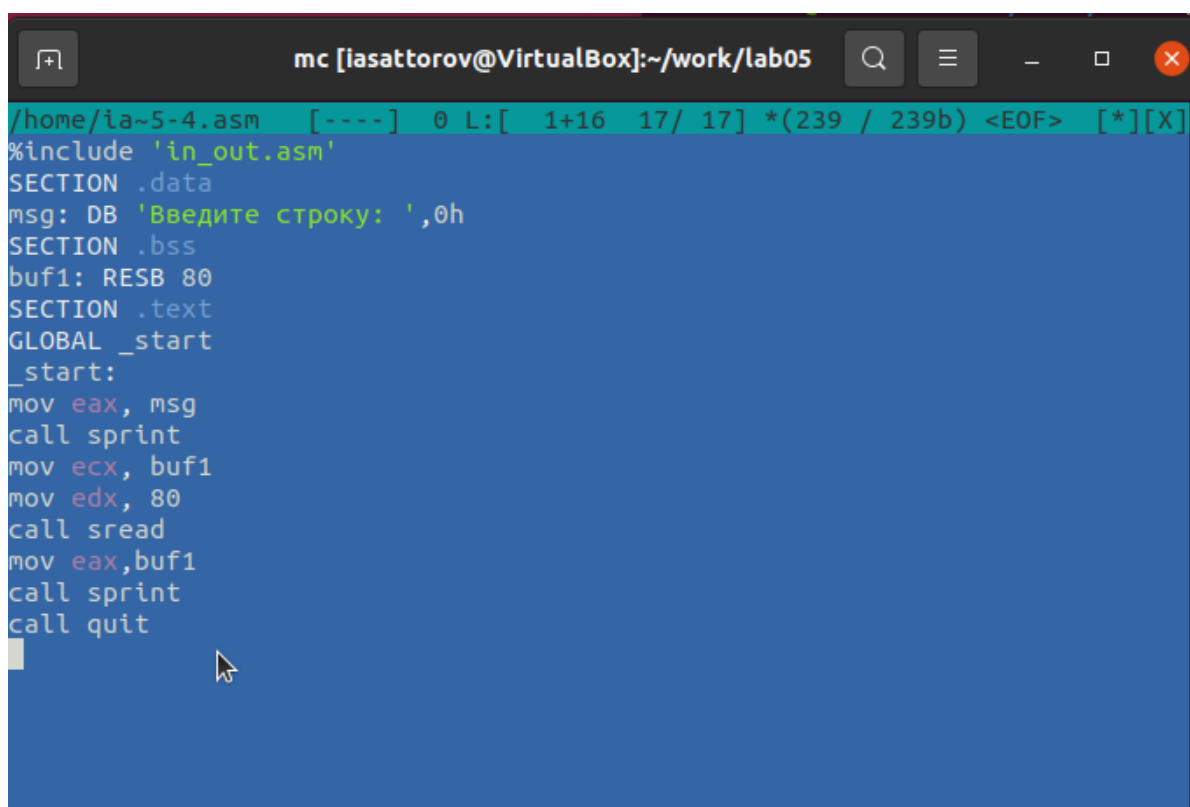


The screenshot shows a terminal window with the following commands and output:

```
iasattorov@VirtualBox:~/work/lab05$ nasm -f elf lab05-3.asm
iasattorov@VirtualBox:~/work/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3
iasattorov@VirtualBox:~/work/lab05$ ./lab05-3
Введите строку:
Ikrom
Ikrom
iasattorov@VirtualBox:~/work/lab05$
```

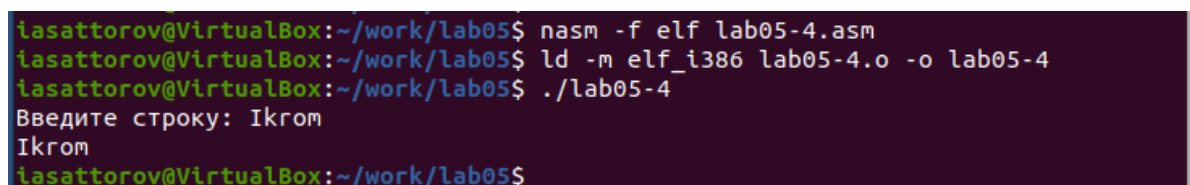
Рис. 2.13: Компиляция и проверка программы lab05-3.asm

Аналогично изменил программу lab05-2.asm.



```
mc [iasattorov@VirtualBox]:~/work/lab05
/home/ia~5-4.asm  [----] 0 L:[ 1+16 17/ 17] *(239 / 239b) <EOF> [*][X]
%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. 2.14: Редактирование файла lab05-4.asm



```
iasattorov@VirtualBox:~/work/lab05$ nasm -f elf lab05-4.asm
iasattorov@VirtualBox:~/work/lab05$ ld -m elf_i386 lab05-4.o -o lab05-4
iasattorov@VirtualBox:~/work/lab05$ ./lab05-4
Введите строку: Ikrom
Ikrom
iasattorov@VirtualBox:~/work/lab05$
```

Рис. 2.15: Компиляция и проверка 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`.