

Отчёт по лабораторной работе 5

Архитектура компьютера

Абдурахмонова Рухшона Бахтиёровна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	18

Список иллюстраций

2.1	Запуск Midnight Commander	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 lab05-1.asm	11
2.7	Программа в файле lab05-2.asm	12
2.8	Запуск программы lab05-2.asm	12
2.9	Программа в файле lab05-2.asm	13
2.10	Запуск программы lab05-2.asm	13
2.11	Программа в файле lab05-3.asm	14
2.12	Запуск программы lab05-3.asm	15
2.13	Программа в файле lab05-4.asm	16
2.14	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`.

2 Выполнение лабораторной работы

Открыла Midnight Commander, с помощью клавиш со стрелками и Enter перехожу в каталог ~/work/arch-pc. Далее нажимаю F7 и создаю каталог lab05

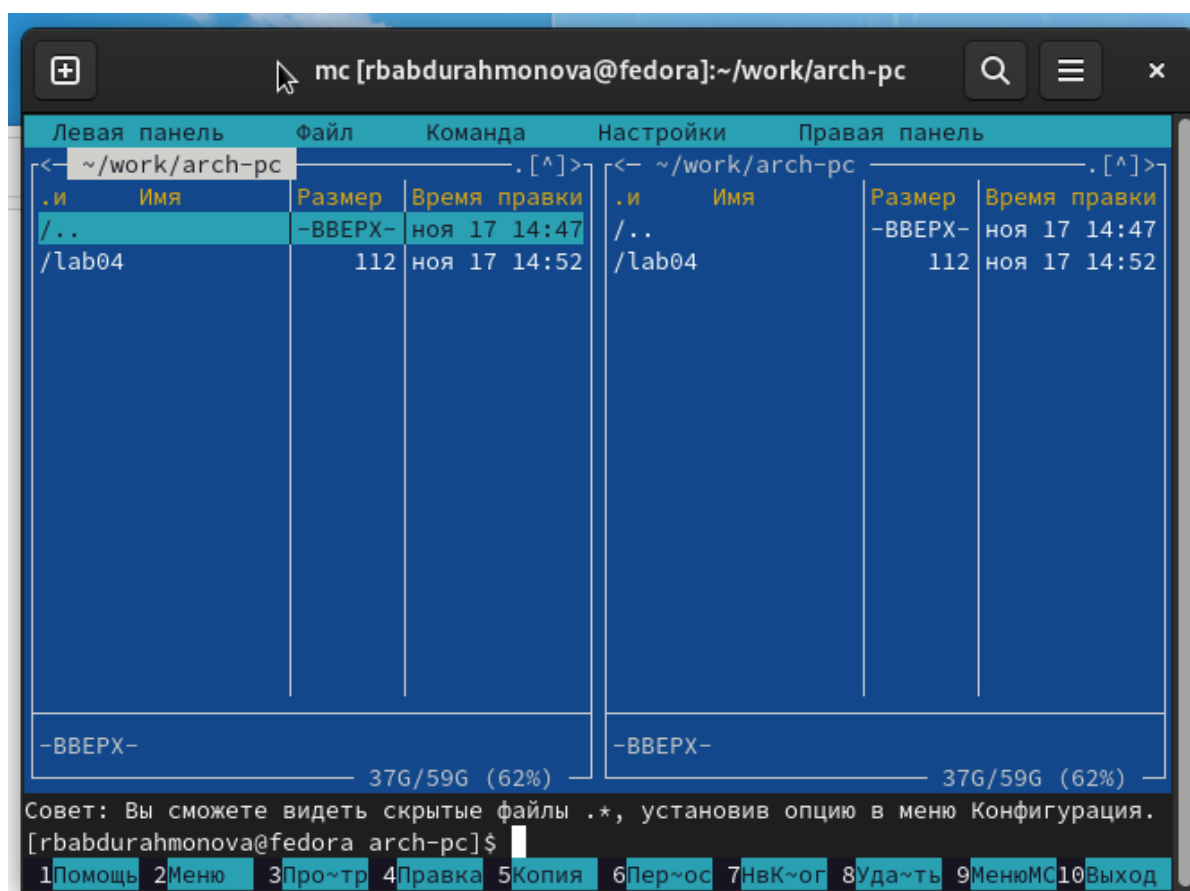


Рис. 2.1: Запуск Midnight Commander

При помощи touch создала файл lab05-1.asm

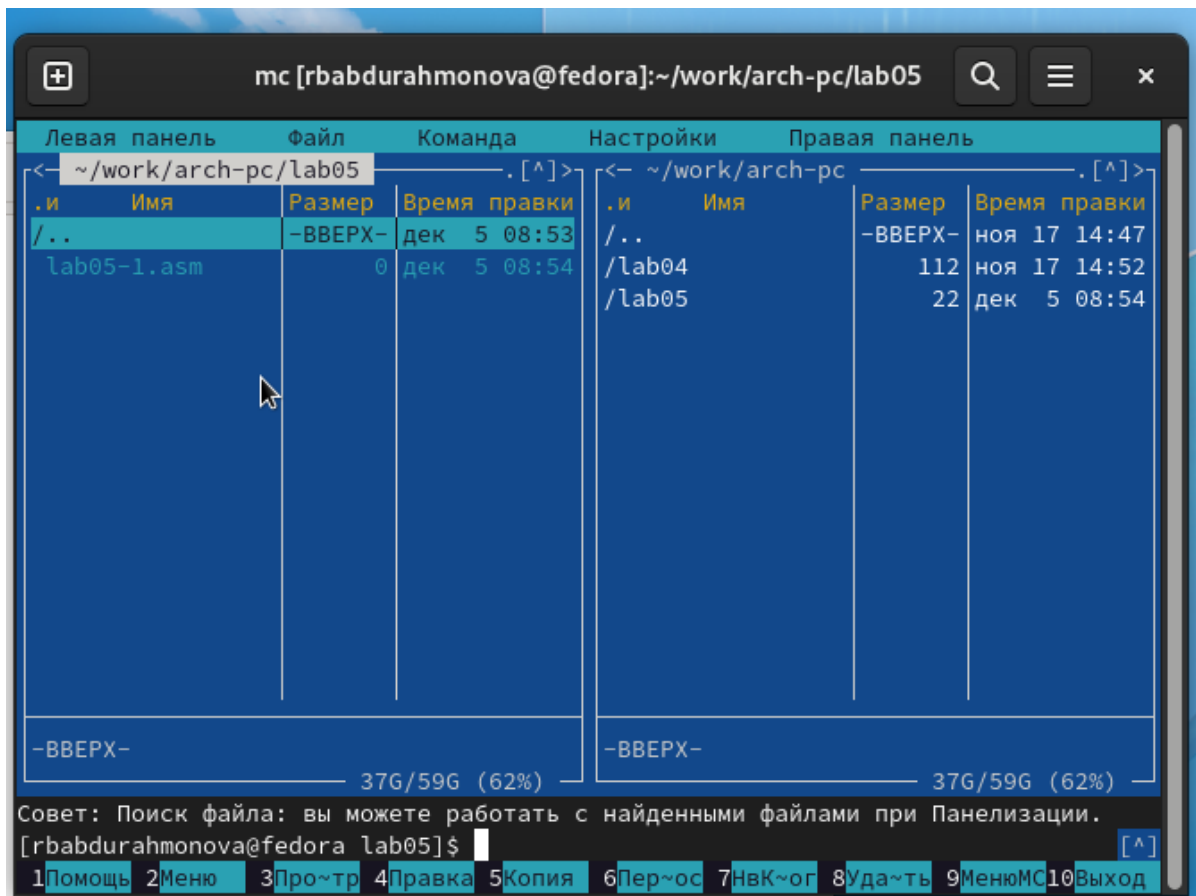
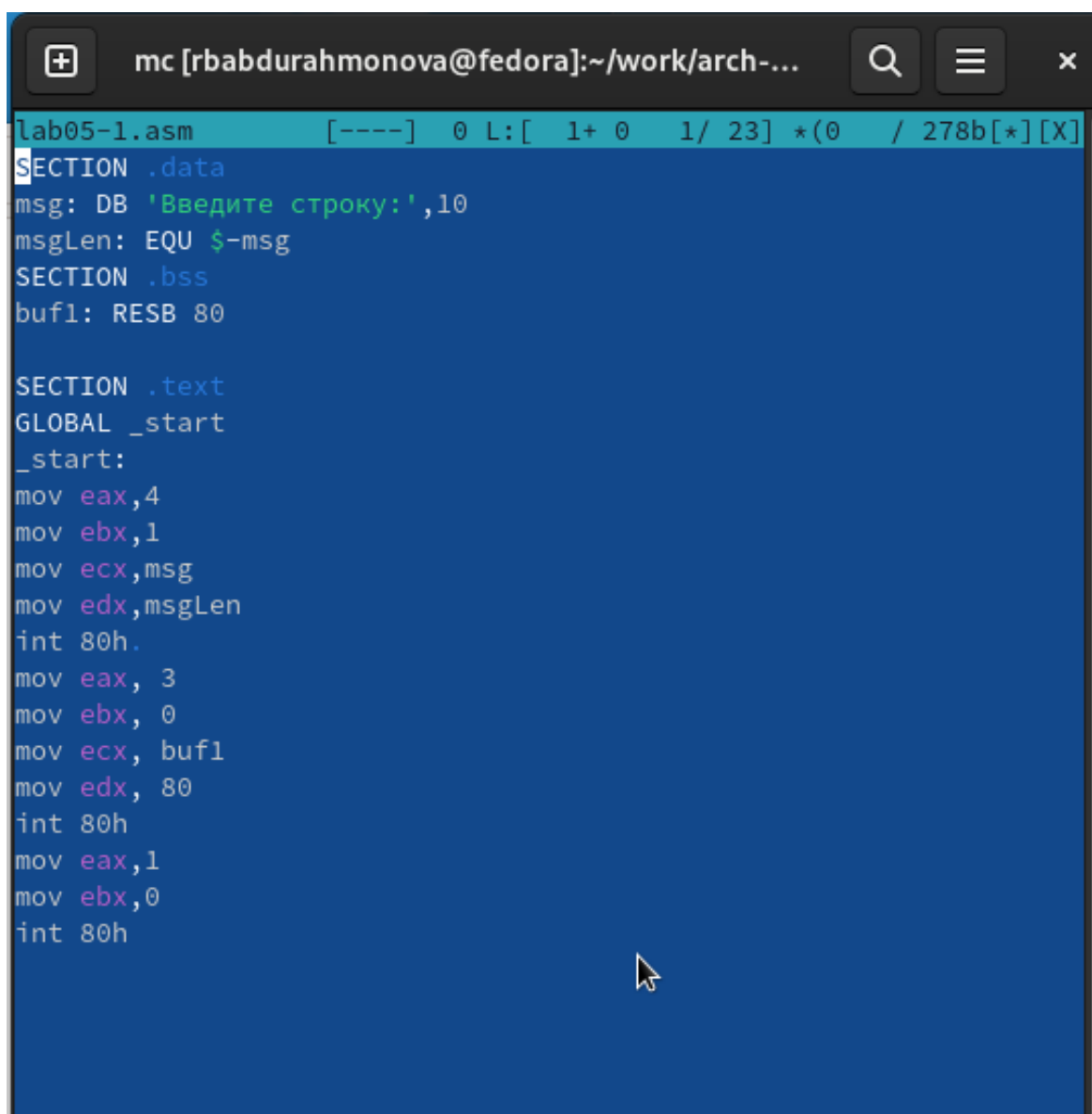


Рис. 2.2: Создание файла lab05-1.asm

Открыла файл на редактирование клавишей F4, выбрала редактор mceditor, написала код программы из задания.

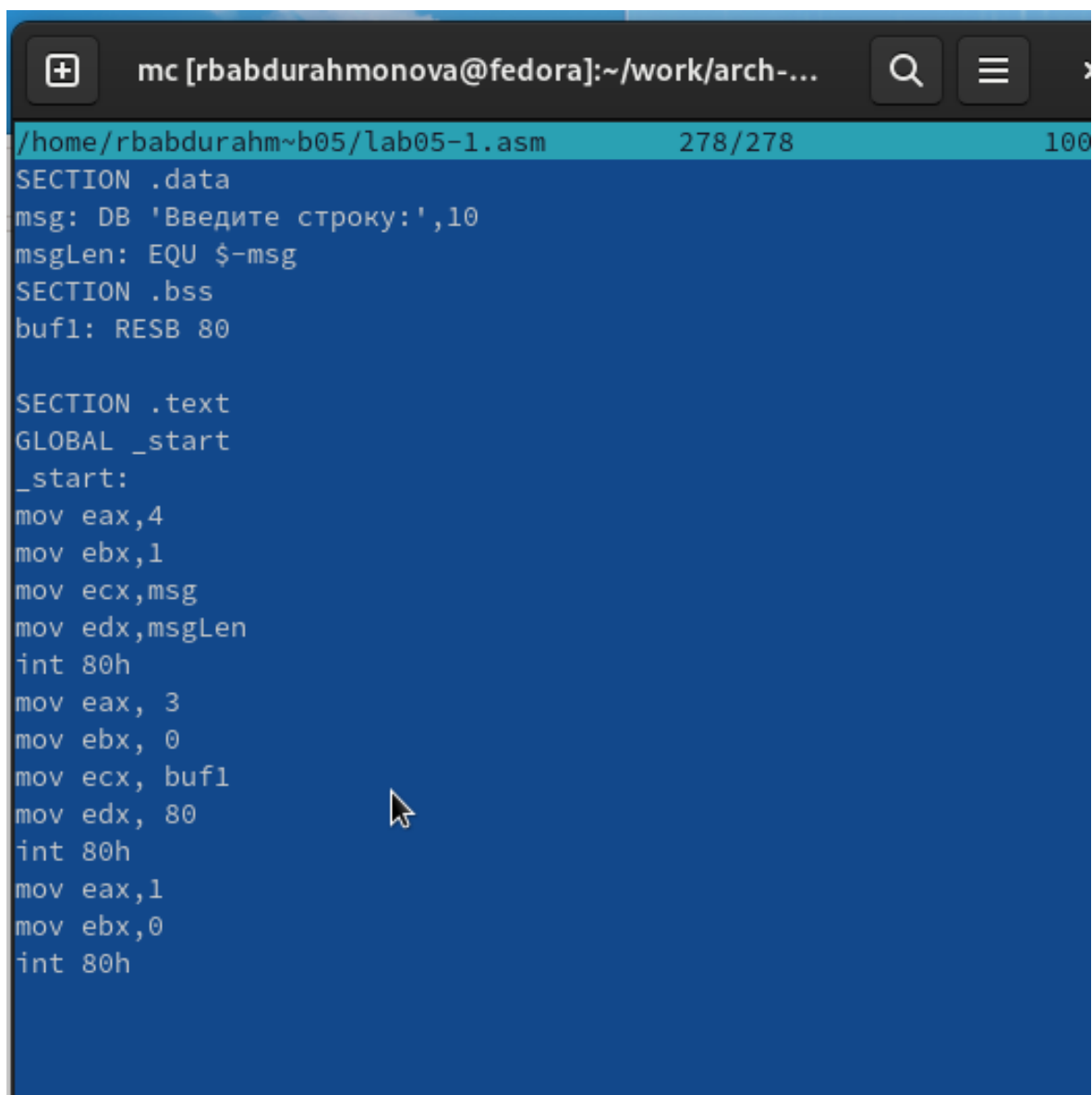


```
lab05-1.asm [----] 0 L: [ 1+ 0 1/ 23] *(0 / 278b[*] [X]
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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

Рис. 2.3: Программа в файле lab05-1.asm

Открыла файл на просмотр клавишей F3 и проверила, что он содержит набранный код.



```
mc [rbabdurahmonova@fedora]:~/work/arch-...
/home/rbabdurahm~b05/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

Транслировала файл программы в объектный файл, выполнила компоновку объектного файла, получила исполняемый файл программы и проверила ее работу.

```
[rbabdurahmonova@fedora lab05]$ nasm -f elf lab05-1.asm
[rbabdurahmonova@fedora lab05]$ ld -m elf_i386 lab05-1.o -o lab05-1
[rbabdurahmonova@fedora lab05]$ ./lab05-1
Введите строку:
test
[rbabdurahmonova@fedora lab05]$
[rbabdurahmonova@fedora lab05]$
```

Рис. 2.5: Запуск программы lab05-1.asm

Для упрощения написания программ часто встречающиеся одинаковые участки кода (такие как, например, вывод строки на экран или выход из программы) можно оформить в виде подпрограмм и сохранить в отдельные файлы, а во всех нужных местах поставить вызов нужной подпрограммы. Это позволяет сделать основную программу более удобной для написания и чтения.

Скачала файл `in_out.asm` и разместила его в рабочем каталоге. Для копирования используется клавиша F5. Для перемещения используется клавиша F6.

Скопировала `lab05-1.asm` в `lab05-2.asm`.

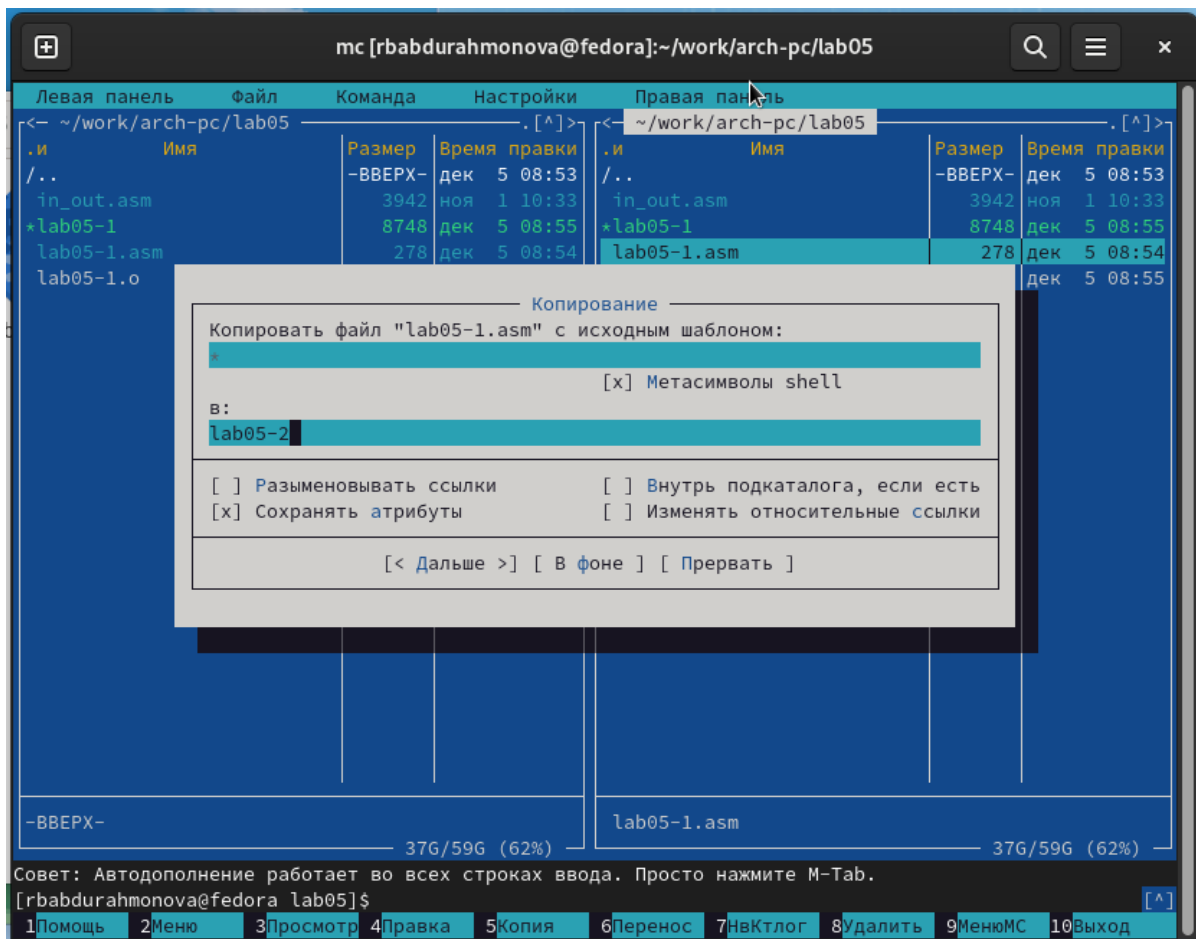
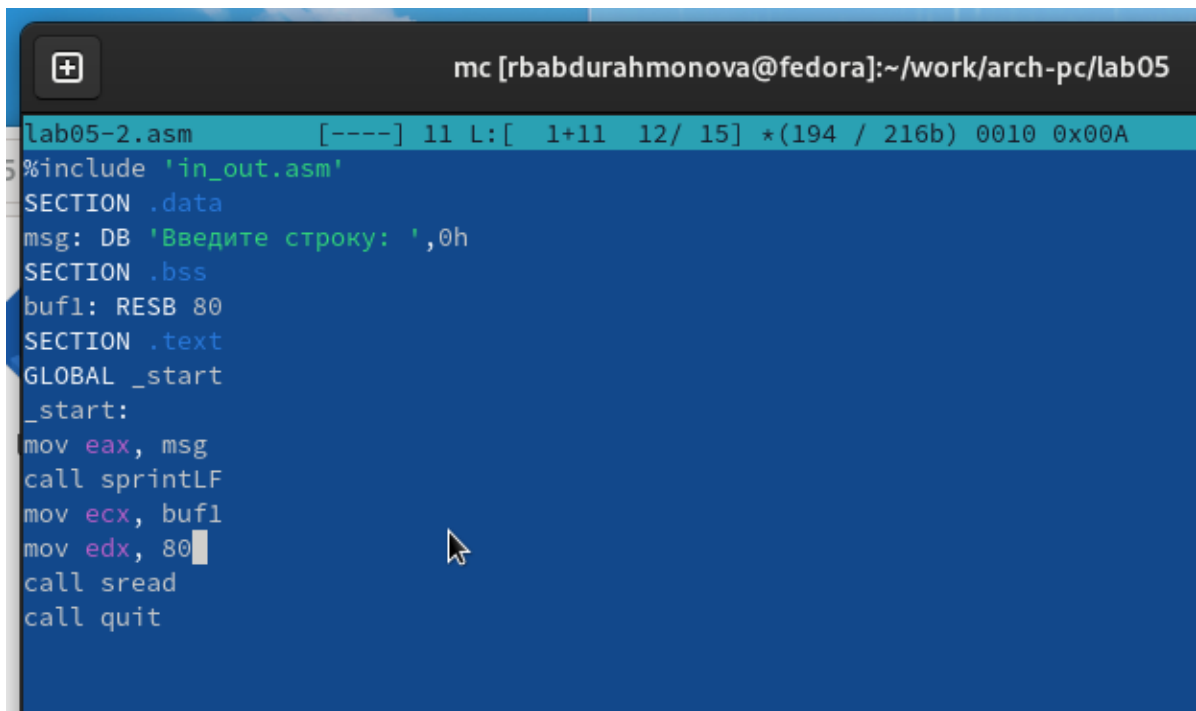


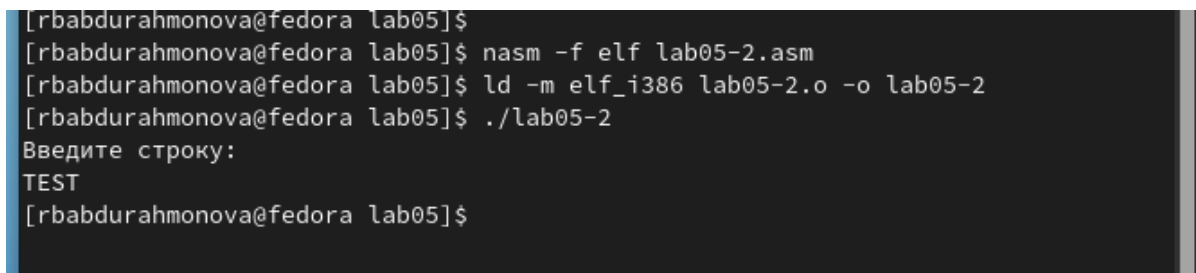
Рис. 2.6: Копирование файла lab05-1.asm

Написала код программы lab05-2.asm с использованием подпрограмм из внешнего файла in_out.asm.



```
mc [rbabdurahmonova@fedora]:~/work/arch-pc/lab05
lab05-2.asm [----] 11 L: [ 1+11 12/ 15] *(194 / 216b) 0010 0x00A
%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
```

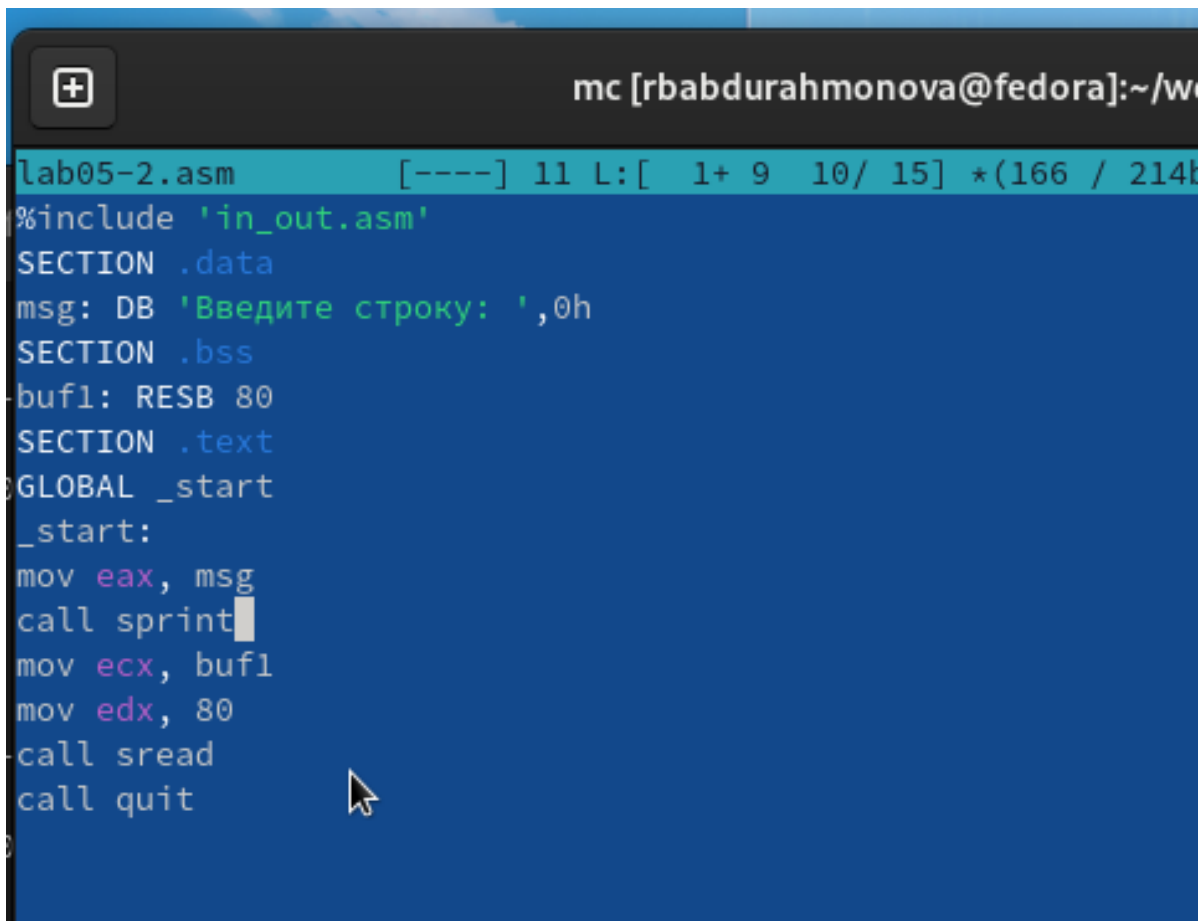
Рис. 2.7: Программа в файле lab05-2.asm



```
[rbabdurahmonova@fedora lab05]$
[rbabdurahmonova@fedora lab05]$ nasm -f elf lab05-2.asm
[rbabdurahmonova@fedora lab05]$ ld -m elf_i386 lab05-2.o -o lab05-2
[rbabdurahmonova@fedora lab05]$ ./lab05-2
Введите строку:
TEST
[rbabdurahmonova@fedora lab05]$
```

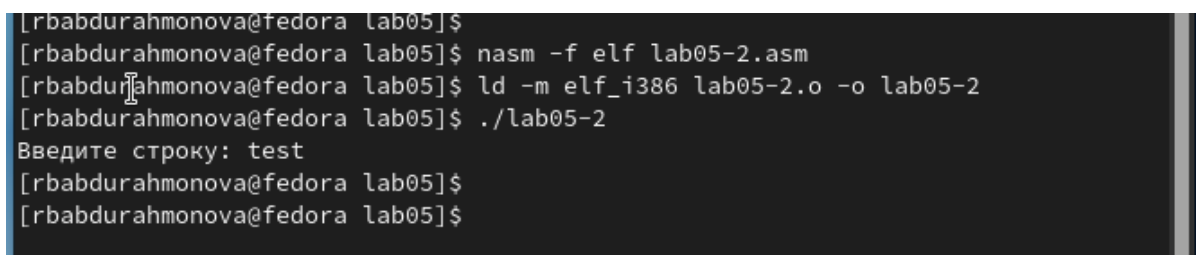
Рис. 2.8: Запуск программы lab05-2.asm

В файле lab5-2.asm заменила подпрограмму sprintf на sprint. Заново собрала исполняемый файл. Теперь после вывода строки она не завершается символом перехода на новую строку.



```
mc [rbabdurahmonova@fedora]:~/w
lab05-2.asm [----] 11 L: [ 1+ 9 10/ 15] *(166 / 2148
%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



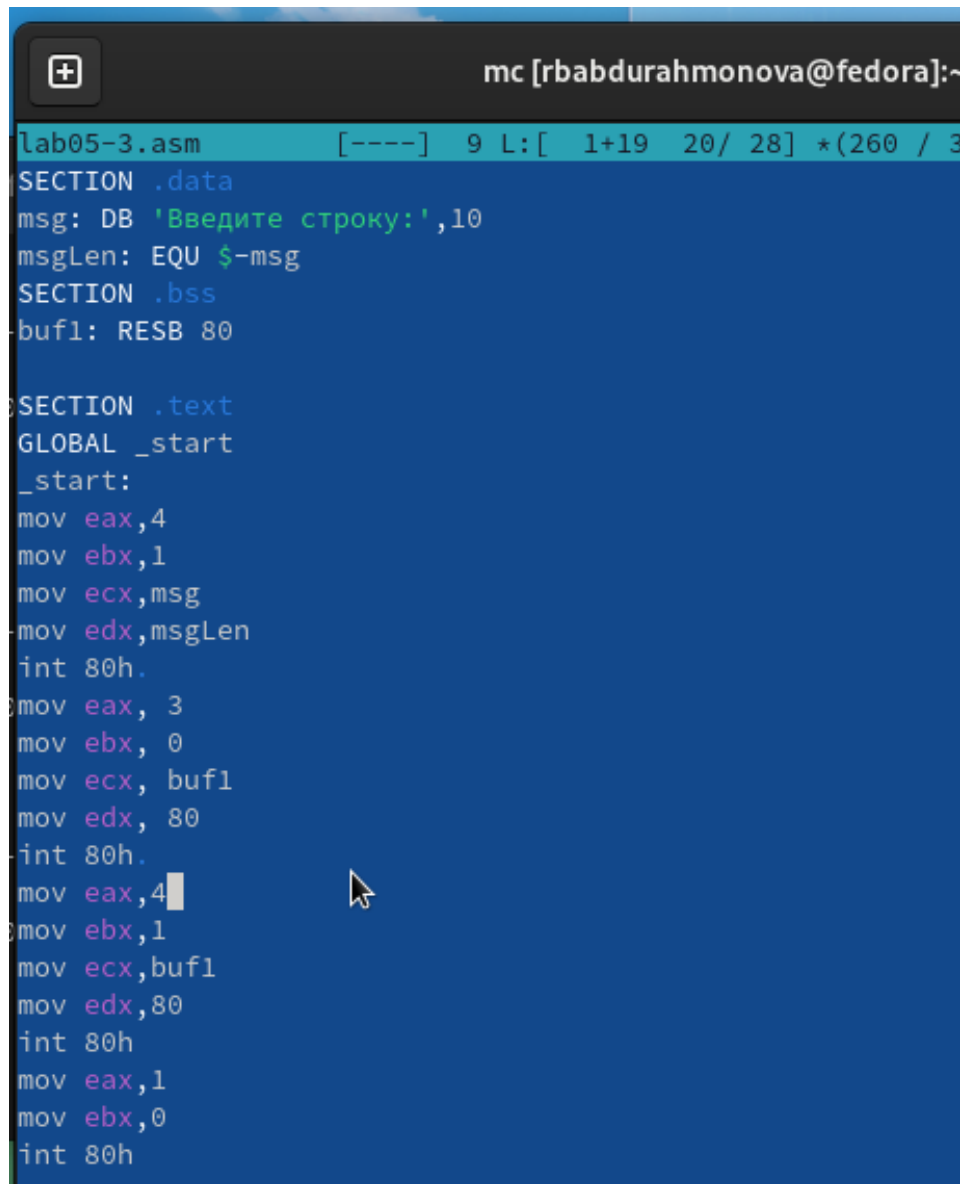
```
[rbabdurahmonova@fedora lab05]$
[rbabdurahmonova@fedora lab05]$ nasm -f elf lab05-2.asm
[rbabdurahmonova@fedora lab05]$ ld -m elf_i386 lab05-2.o -o lab05-2
[rbabdurahmonova@fedora lab05]$ ./lab05-2
Введите строку: test
[rbabdurahmonova@fedora lab05]$
[rbabdurahmonova@fedora lab05]$
```

Рис. 2.10: Запуск программы lab05-2.asm

Скопировала программу lab05-1.asm и изменила код, так чтобы она работала по следующему алгоритму:

- вывести приглашение типа “Введите строку:”;

- ввести строку с клавиатуры;
- вывести введенную строку на экран.



```
mc [rbabdurahmonova@fedora]:~
lab05-3.asm [----] 9 L: [ 1+19 20/ 28] *(260 / 3
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

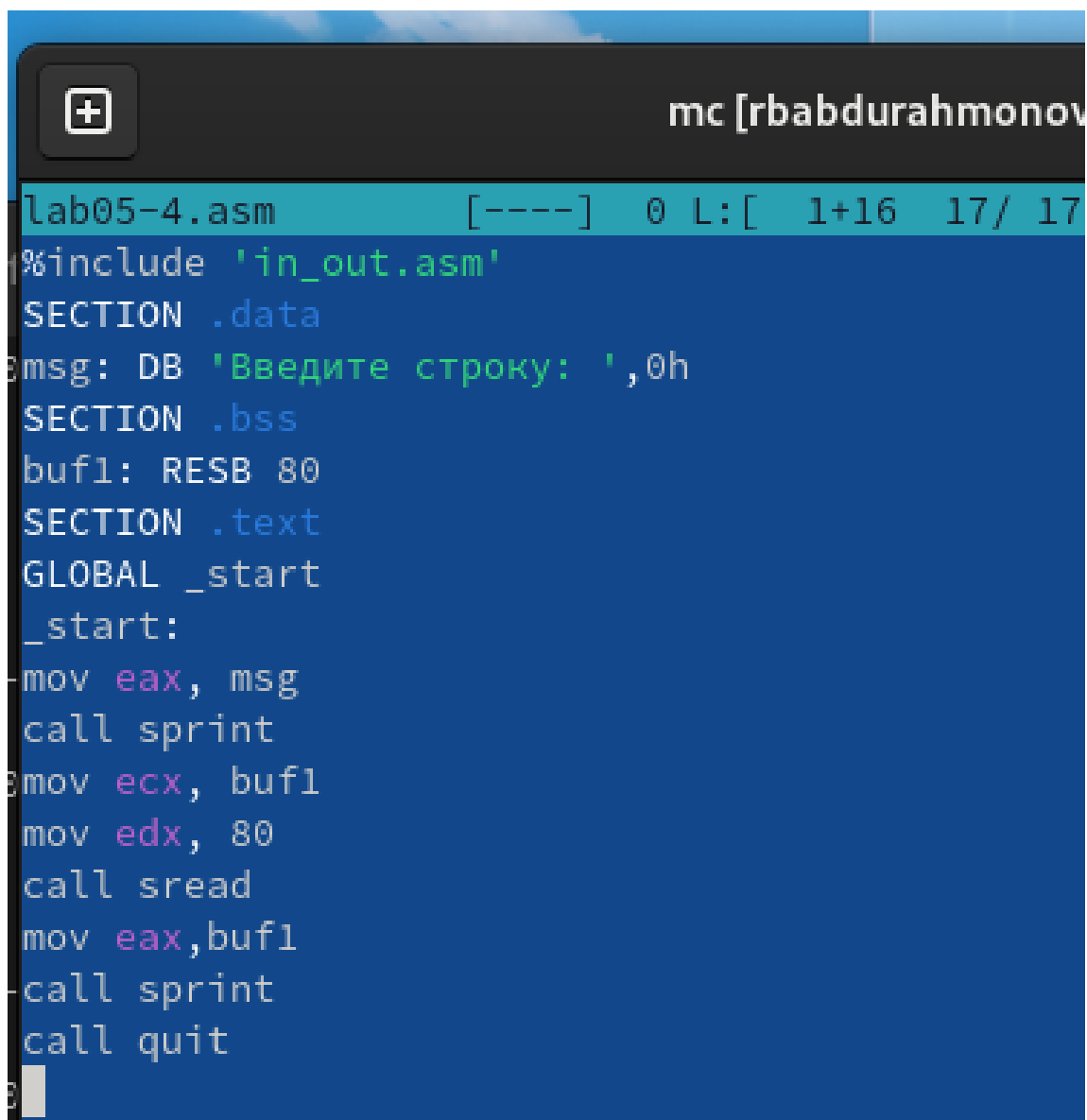
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h.
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h.
mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,80
int 80h
mov eax,1
mov ebx,0
int 80h
```

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

```
[rbabdurahmonova@fedora lab05]$ nasm -f elf lab05-3.asm
[rbabdurahmonova@fedora lab05]$ ld -m elf_i386 lab05-3.o -o lab05-3
[rbabdurahmonova@fedora lab05]$ ./lab05-3
Введите строку:
test
test
[rbabdurahmonova@fedora lab05]$
```

Рис. 2.12: Запуск программы lab05-3.asm

Аналогично скопировала программу lab05-2.asm и изменила код, но теперь использовал подпрограммы из файла in_out.asm.



```
lab05-4.asm [----] 0 L: [ 1+16 17/ 17]
%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


```
[rbabdurahmonova@fedora lab05]$  
[rbabdurahmonova@fedora lab05]$ nasm -f elf lab05-4.asm  
[rbabdurahmonova@fedora lab05]$ ld -m elf_i386 lab05-4.o -o lab05-4  
[rbabdurahmonova@fedora lab05]$ ./lab05-4  
Введите строку: test  
test  
[rbabdurahmonova@fedora lab05]$  
[rbabdurahmonova@fedora lab05]$
```

Рис. 2.14: Запуск программы lab05-4.asm

3 Выводы

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