

Отчёт по лабораторной работе 5

Архитектура компьютера

Арслан Юсупов

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 | 16 |
| 3 | Выводы | 21 |

Список иллюстраций

| | | |
|------|---|----|
| 2.1 | Запуск Midnight Commander | 6 |
| 2.2 | Создание каталога | 7 |
| 2.3 | Создание файла lab05-1.asm | 8 |
| 2.4 | Программа в файле 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 | 12 |
| 2.8 | Копирование файла lab05-1.asm | 13 |
| 2.9 | Программа в файле lab05-2.asm | 14 |
| 2.10 | Запуск программы lab05-2.asm | 14 |
| 2.11 | Программа в файле lab05-2.asm | 15 |
| 2.12 | Запуск программы lab05-2.asm | 16 |
| 2.13 | Копирование файла lab05-1.asm | 17 |
| 2.14 | Программа в файле lab05-3.asm | 18 |
| 2.15 | Запуск программы lab05-3.asm | 18 |
| 2.16 | Копирование файла lab05-2.asm | 19 |
| 2.17 | Программа в файле lab05-4.asm | 20 |
| 2.18 | Запуск программы lab05-4.asm | 20 |

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

2.1 Знакомство с Midnight Commander

Открыл Midnight Commander, с помощью клавишь со стрелками и Enter перешел в каталог `~/work/arch-pc`. Далее нажал F7 и создал каталог `lab05`

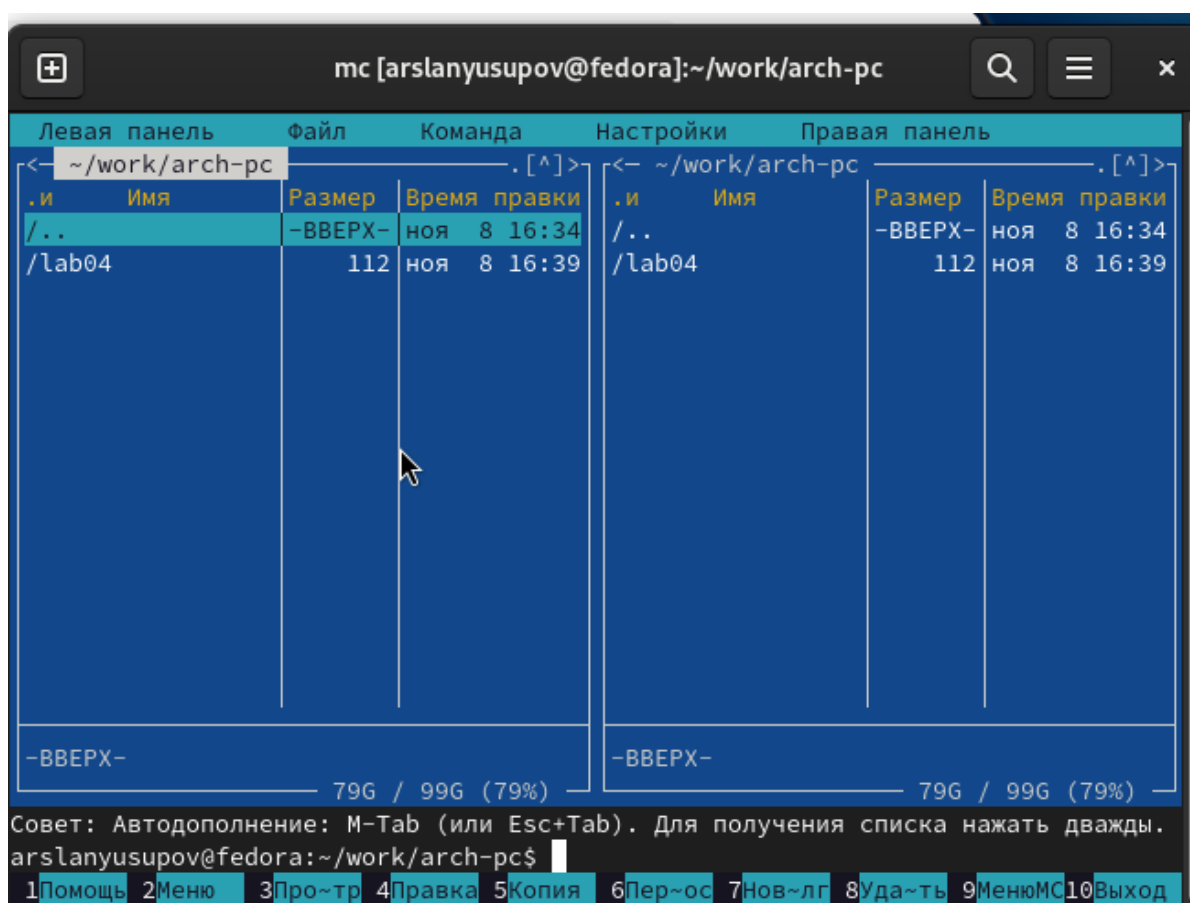


Рис. 2.1: Запуск Midnight Commander

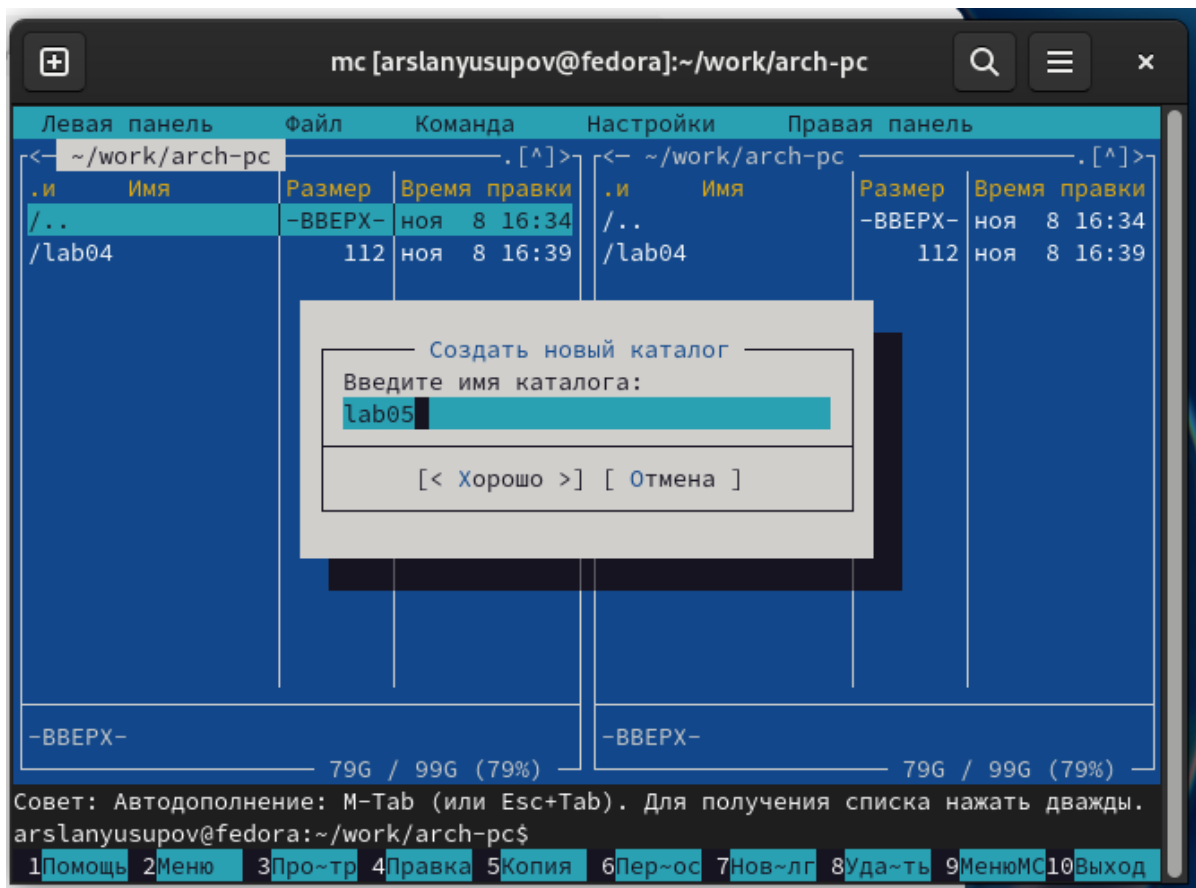


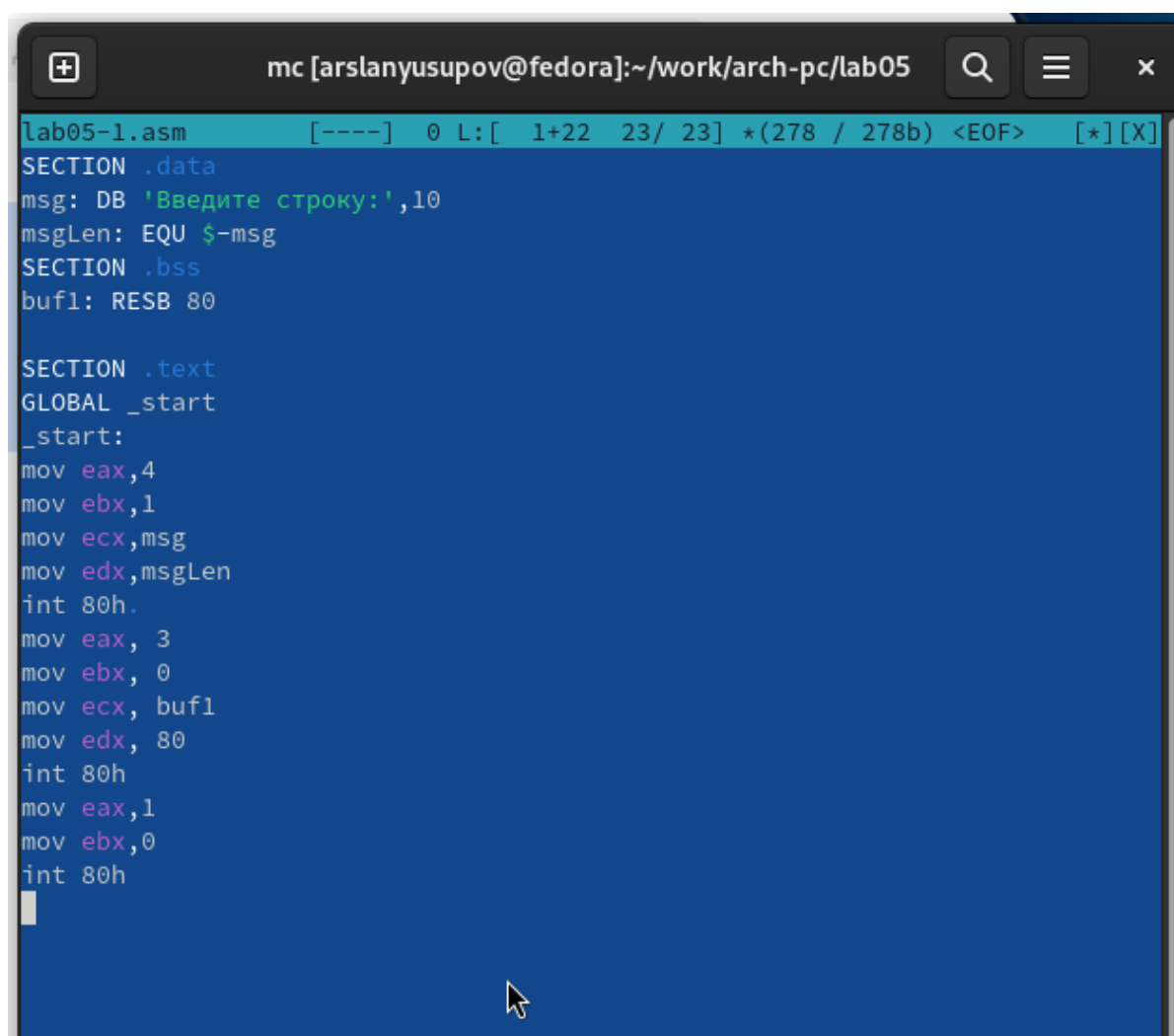
Рис. 2.2: Создание каталога

При помощи touch создал файл lab05-1.asm



Рис. 2.3: Создание файла lab05-1.asm

Открыл файл на редактирование клавишей F4, выбрал редактор mceditor, написал код программы из задания.

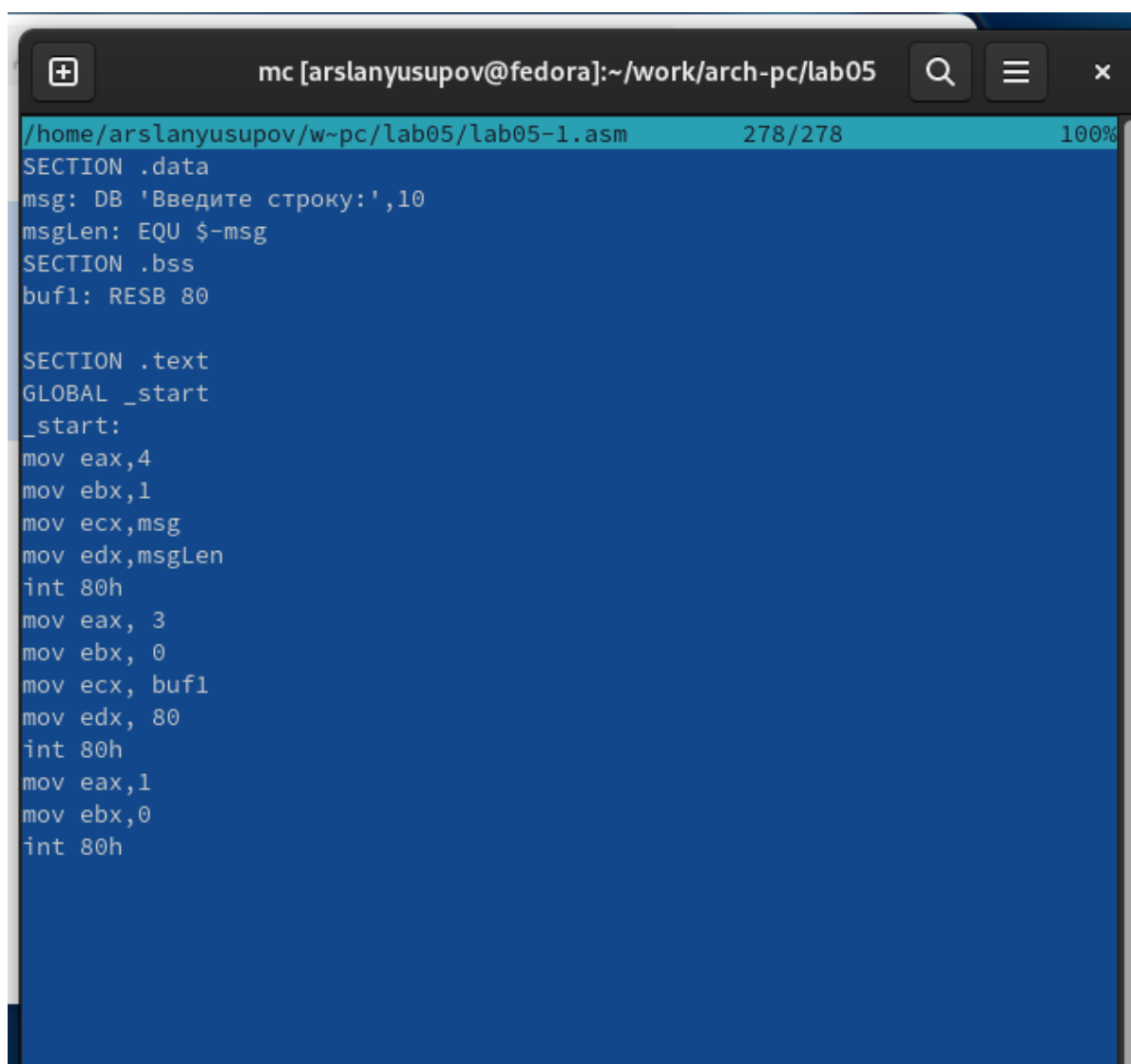


```
mc [arslanyusupov@fedora]:~/work/arch-pc/lab05
lab05-1.asm [----] 0 L: [ 1+22 23/ 23] *(278 / 278b) <EOF> [*] [X]
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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. 2.4: Программа в файле lab05-1.asm

Открыл файл на просмотр клавишей F3 и убедился, что он содержит набранный код.

A screenshot of a code editor window. The title bar shows the user 'mc [arslanyusupov@fedora]' and the file path '~/work/arch-pc/lab05'. The editor displays assembly code for 'lab05-1.asm'. The code is divided into sections: .data, .bss, and .text. The .data section contains a message string and its length. The .bss section reserves space for a buffer. The .text section contains the main logic, starting with a global _start label. The code uses x86 assembly instructions like mov, int, and GLOBAL to set up registers and perform system calls (int 80h) for user input and output.

```
mc [arslanyusupov@fedora]:~/work/arch-pc/lab05 278/278 100%
/home/arslanyusupov/w~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

Транслировал файл программы в объектный файл, выполнил компоновку объектного файла, получил исполняемый файл программы и проверил ее работу.

```
arslanyusupov@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-1.asm
arslanyusupov@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-1.o -o lab05-1
arslanyusupov@fedora:~/work/arch-pc/lab05$ ./lab05-1
Введите строку:
1032244341
arslanyusupov@fedora:~/work/arch-pc/lab05$
arslanyusupov@fedora:~/work/arch-pc/lab05$
```

Рис. 2.6: Запуск программы lab05-1.asm

2.2 Подключение внешнего файла in_out.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.

Для выполнения лабораторных работ используется файл in_out.asm, который содержит следующие подпрограммы:

- `slen` – вычисление длины строки (используется в подпрограммах печати сообщения для определения количества выводимых байтов);
- `sprint` – вывод сообщения на экран, перед вызовом `sprint` в регистр `eax` необходимо записать выводимое сообщение (`mov eax,;`);
- `sprintLF` – работает аналогично `sprint`, но при выводе на экран добавляет к сообщению символ перевода строки;
- `sread` – ввод сообщения с клавиатуры, перед вызовом `sread` в регистр `eax` необходимо записать адрес переменной в которую введенное сообщение буд записано (`mov eax,;`), в регистр `ebx` – длину вводимой строки (`mov ebx,;`);
- `iprint` – вывод на экран чисел в формате ASCII, перед вызовом `iprint` в регистр `eax` необходимо записать выводимое число (`mov eax,;`);

- `iprintLF` – работает аналогично `iprint`, но при выводе на экран после числа добавляет к символ перевода строки;
- `atoi` – функция преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`, перед вызовом `atoi` в регистр `eax` необходимо записать число (`mov eax,;`);
- `quit` – завершение программы.

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

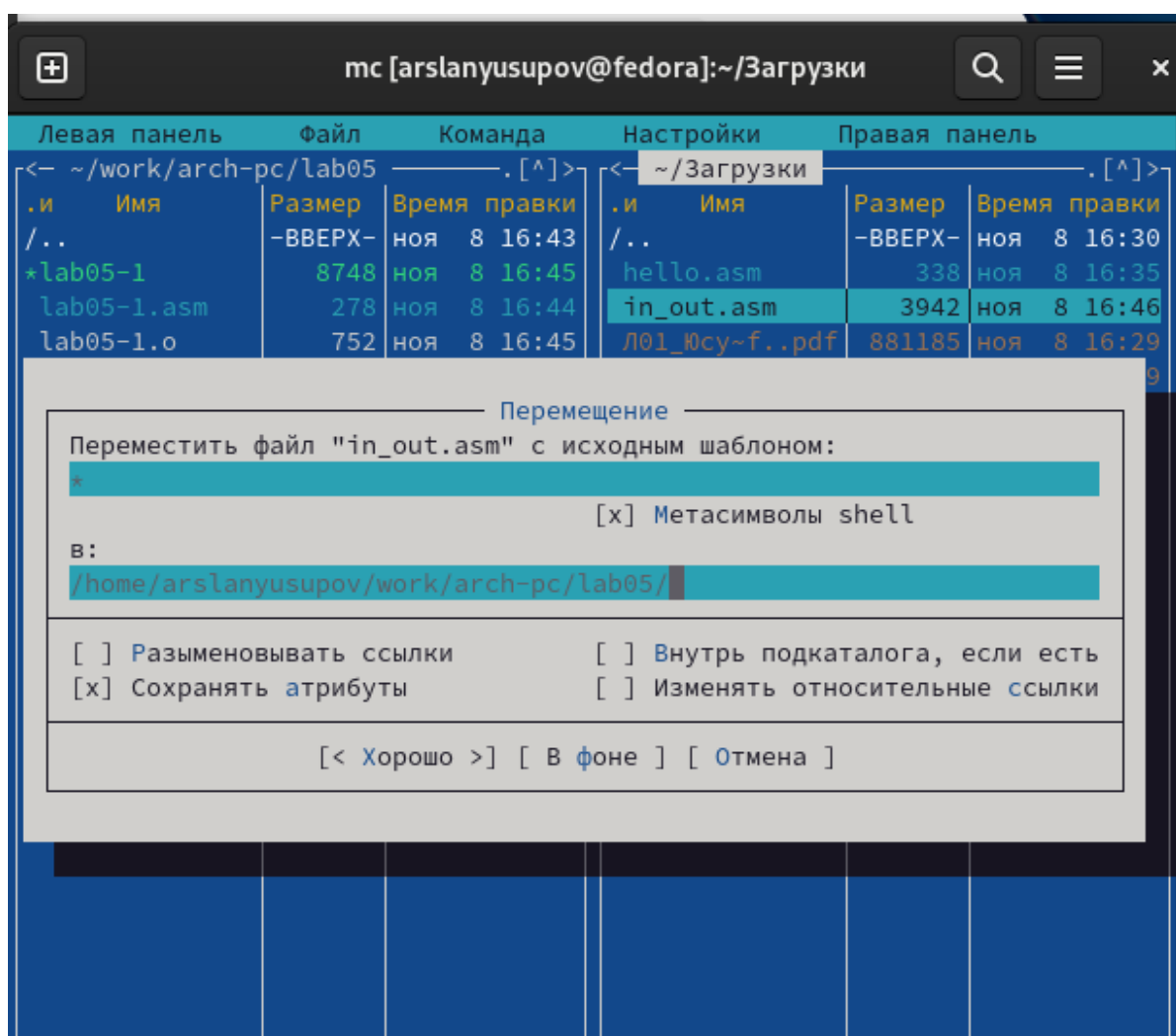


Рис. 2.7: Копирование файла `in_out.asm`

Скопировал lab05-1.asm в lab05-2.asm.

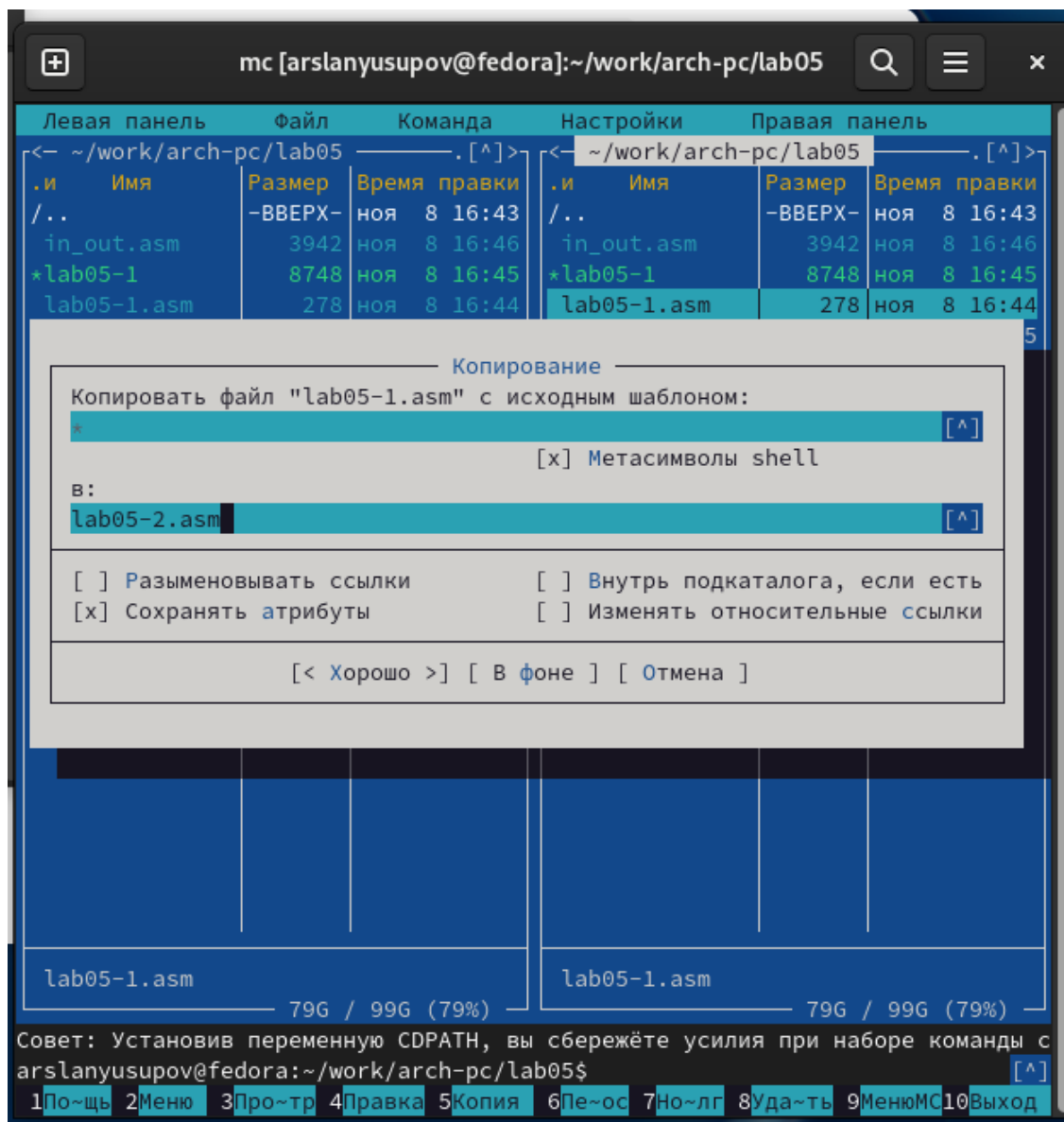
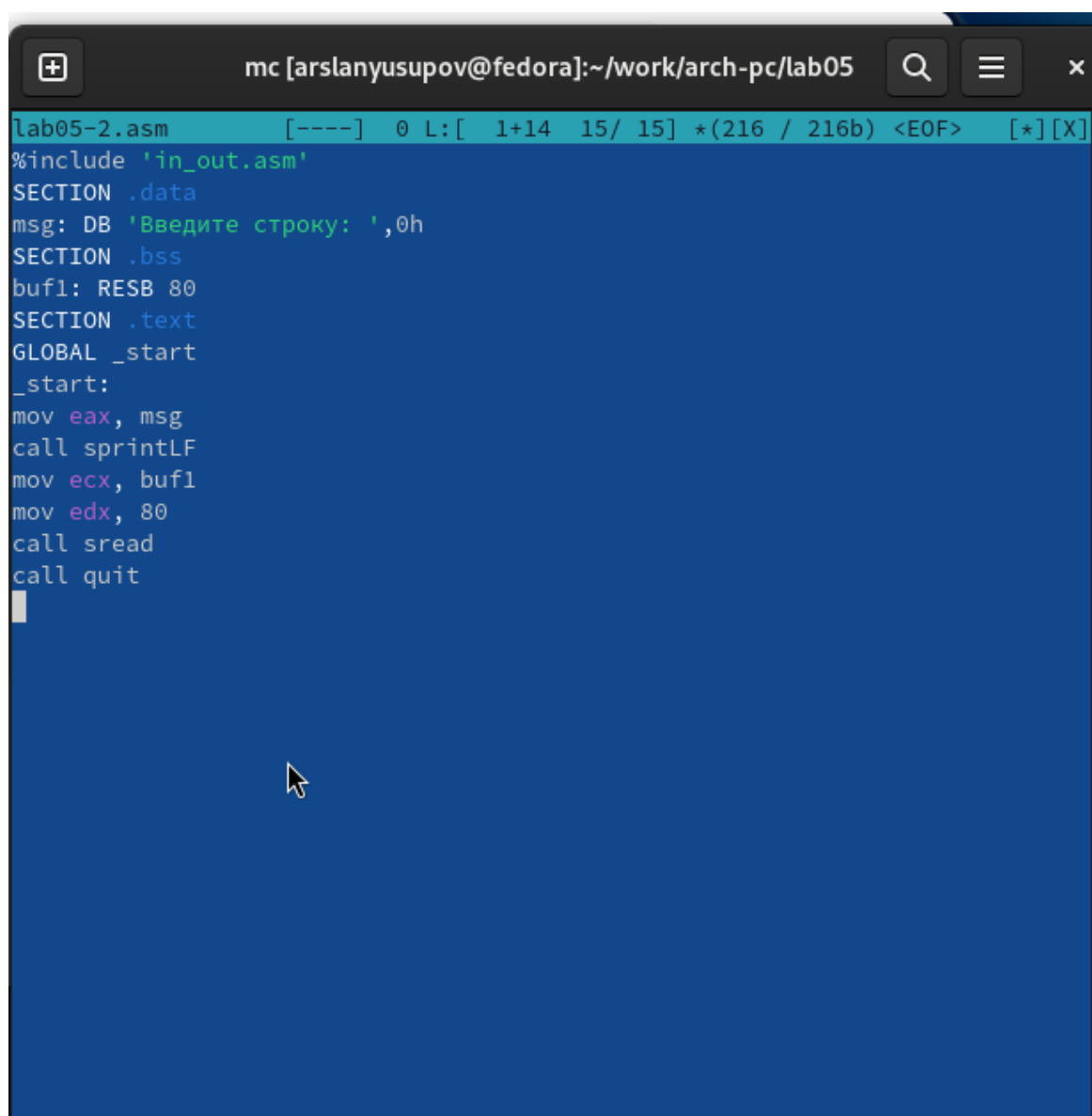


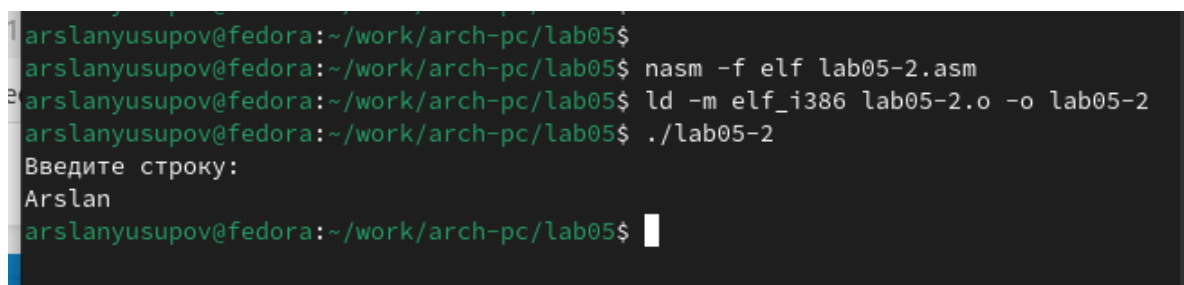
Рис. 2.8: Копирование файла lab05-1.asm

Написал код программы lab05-2.asm с использованием подпрограмм из внешнего файла in_out.asm. Скомпилировал программу и проверил запуск.



```
lab05-2.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
```

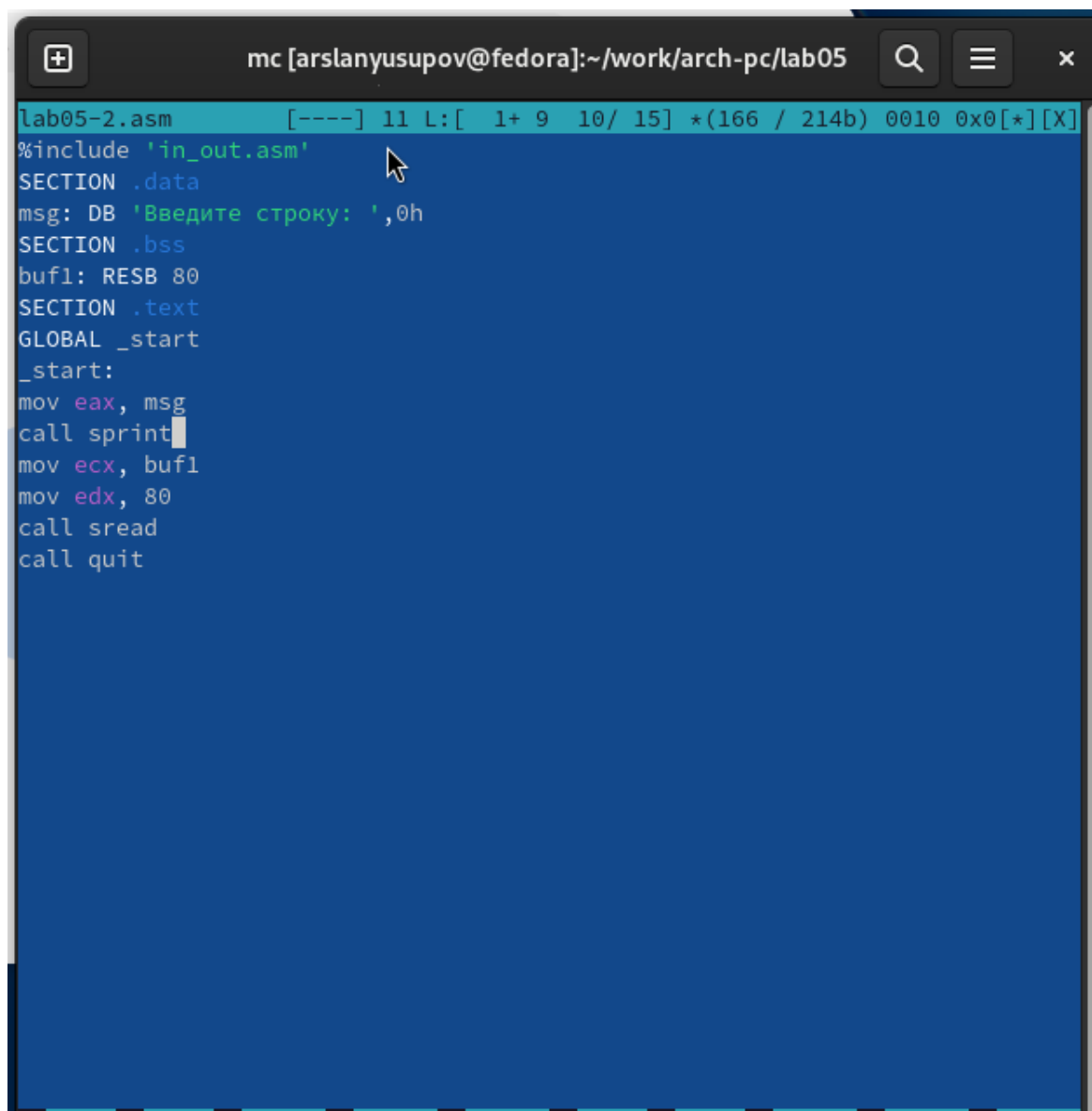
Рис. 2.9: Программа в файле lab05-2.asm



```
1 arslanyusupov@fedora:~/work/arch-pc/lab05$
2 arslanyusupov@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
3 arslanyusupov@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
4 arslanyusupov@fedora:~/work/arch-pc/lab05$ ./lab05-2
Введите строку:
Arslan
arslanyusupov@fedora:~/work/arch-pc/lab05$
```

Рис. 2.10: Запуск программы lab05-2.asm

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



```
lab05-2.asm [-----] 11 L: [ 1+ 9 10/ 15] *(166 / 214b) 0010 0x0[*] [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. 2.11: Программа в файле lab05-2.asm

```
arslanyusupov@fedora:~/work/arch-pc/lab05$  
arslanyusupov@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm  
arslanyusupov@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2  
arslanyusupov@fedora:~/work/arch-pc/lab05$ ./lab05-2  
Введите строку:  
Arslan  
arslanyusupov@fedora:~/work/arch-pc/lab05$  
arslanyusupov@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm  
arslanyusupov@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2  
arslanyusupov@fedora:~/work/arch-pc/lab05$ ./lab05-2  
Введите строку: RUDN  
arslanyusupov@fedora:~/work/arch-pc/lab05$
```

Рис. 2.12: Запуск программы lab05-2.asm

2.3 Задание для самостоятельной работы

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

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;
- вывести введенную строку на экран.

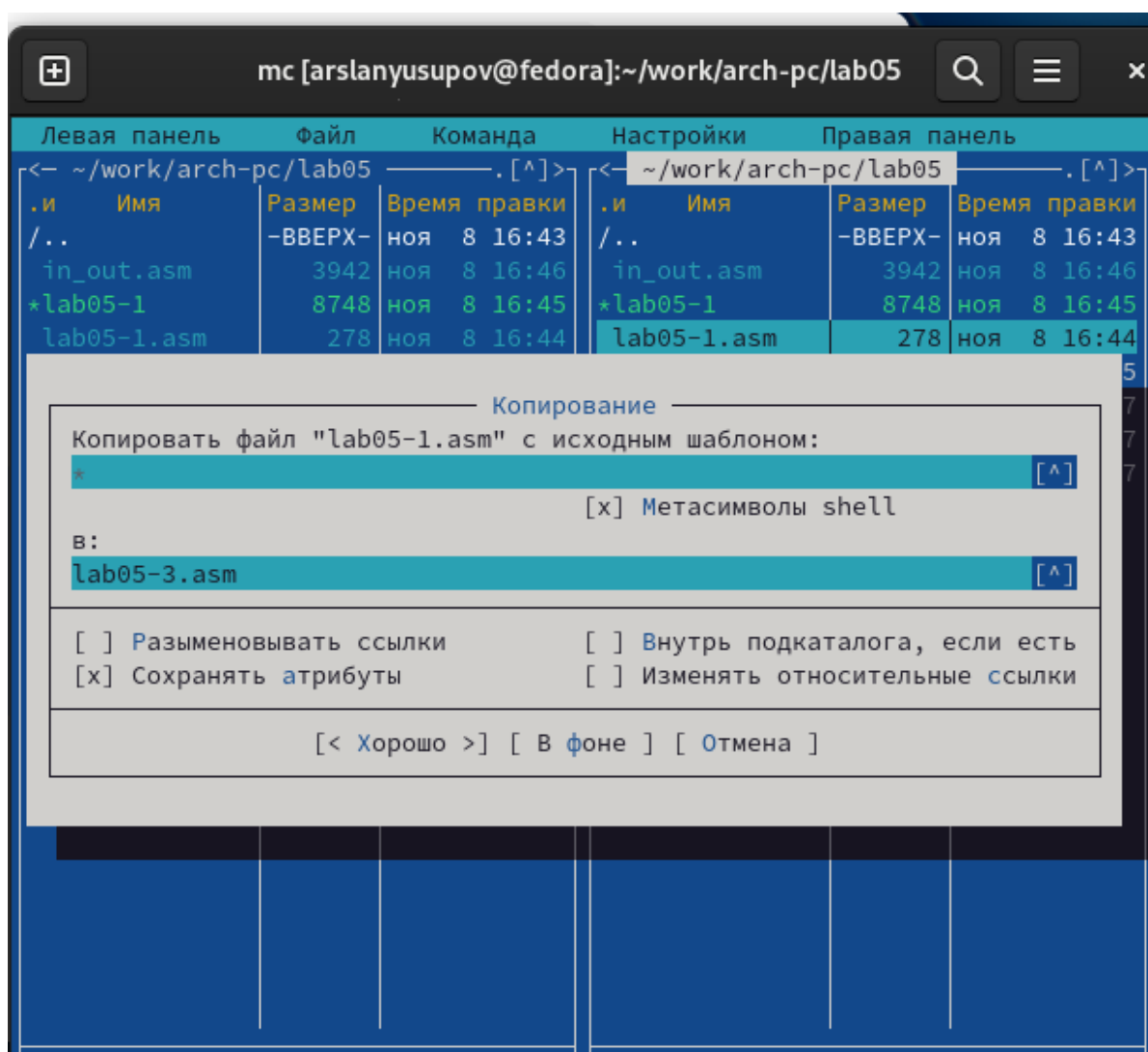
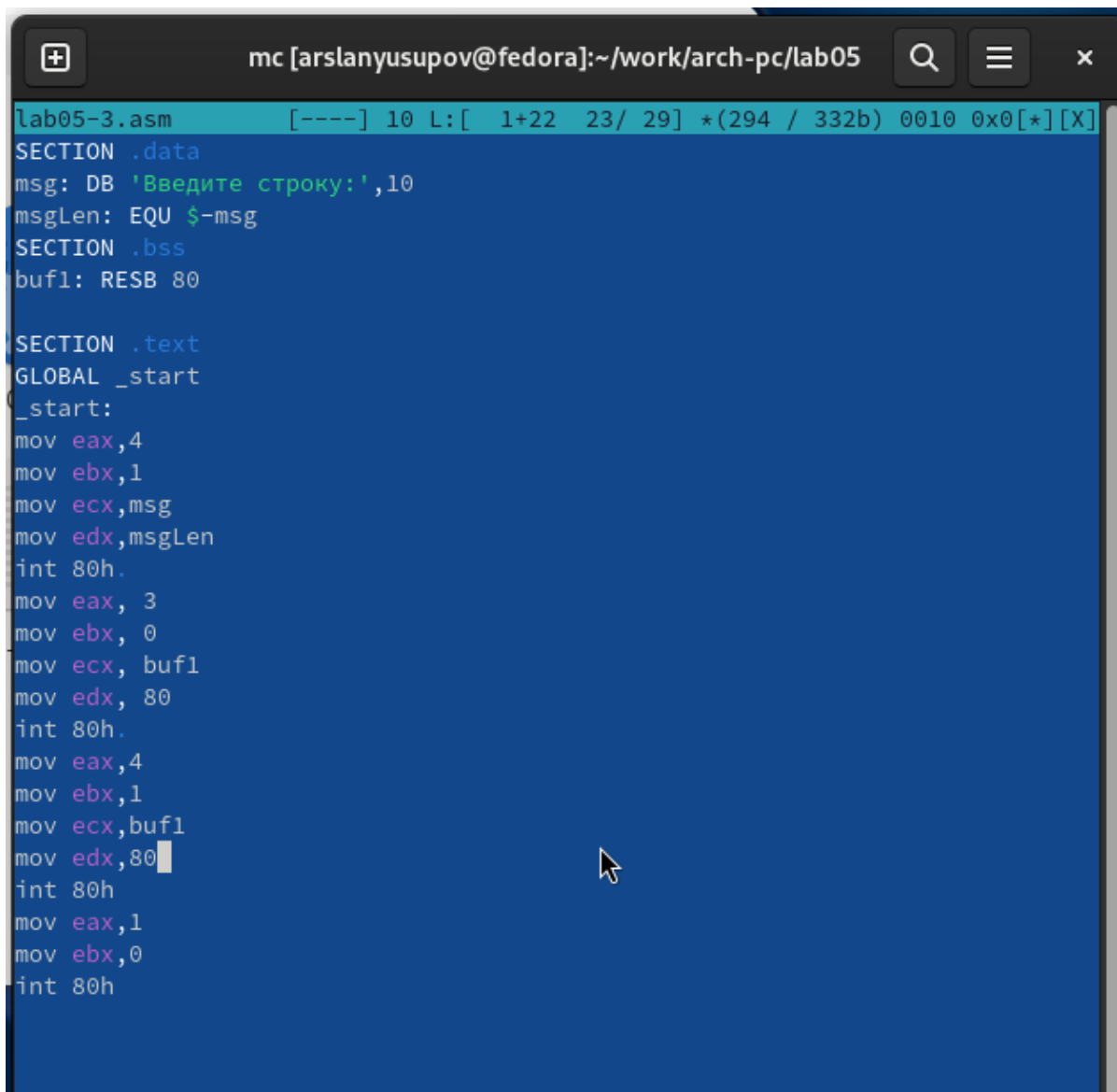


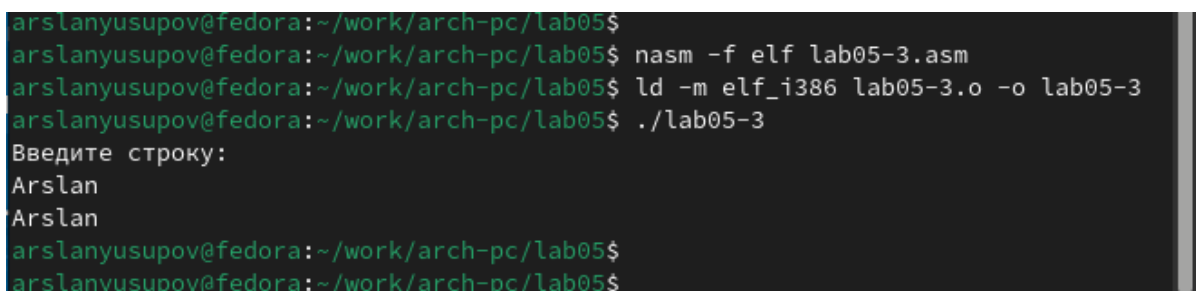
Рис. 2.13: Копирование файла lab05-1.asm



```
lab05-3.asm [----] 10 L: [ 1+22 23/ 29] *(294 / 332b) 0010 0x0[*] [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. 2.14: Программа в файле lab05-3.asm



```
arslanyusupov@fedora:~/work/arch-pc/lab05$
arslanyusupov@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-3.asm
arslanyusupov@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3
arslanyusupov@fedora:~/work/arch-pc/lab05$ ./lab05-3
Введите строку:
Arslan
Arslan
arslanyusupov@fedora:~/work/arch-pc/lab05$
arslanyusupov@fedora:~/work/arch-pc/lab05$
```

Рис. 2.15: Запуск программы lab05-3.asm

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

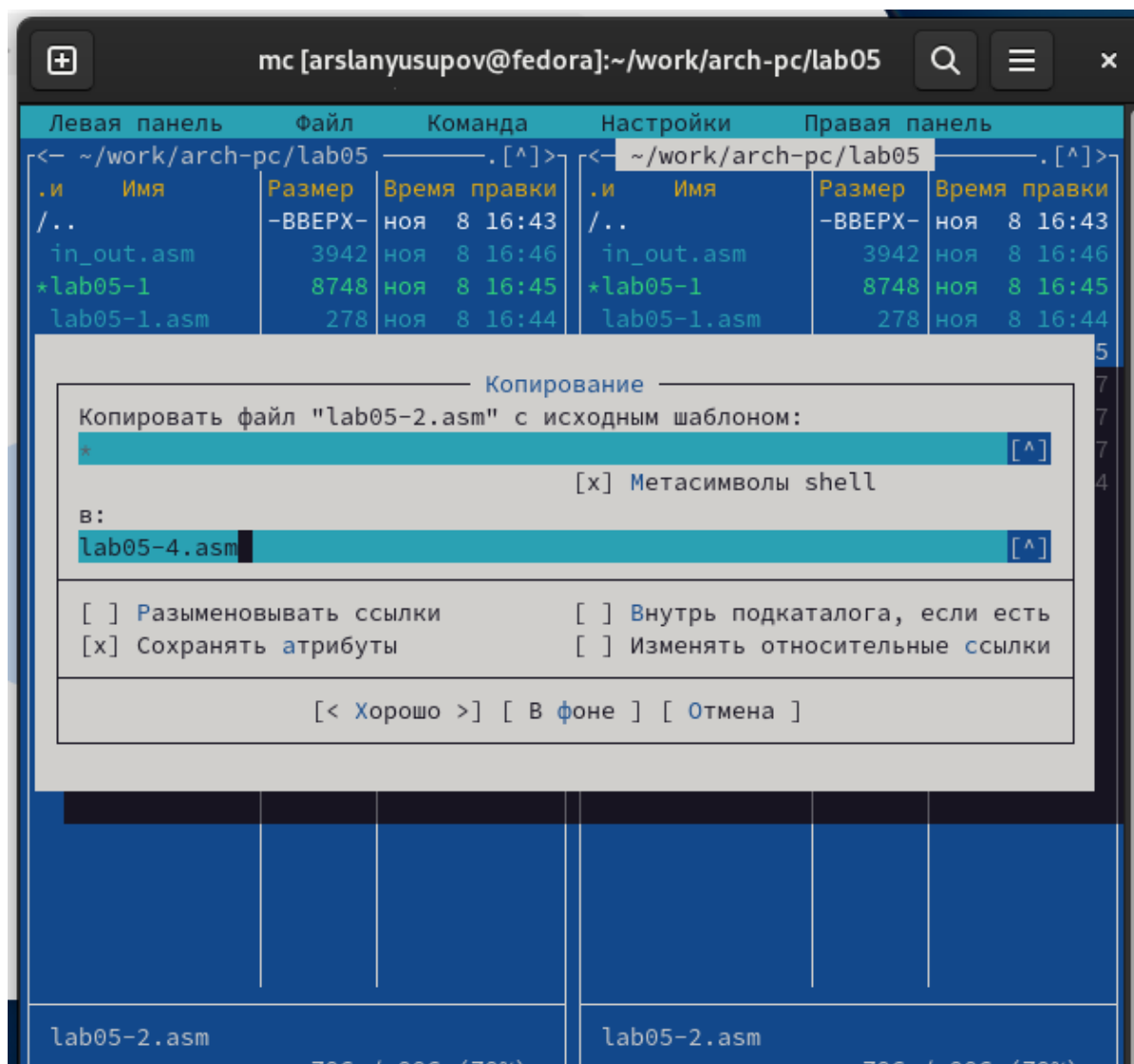
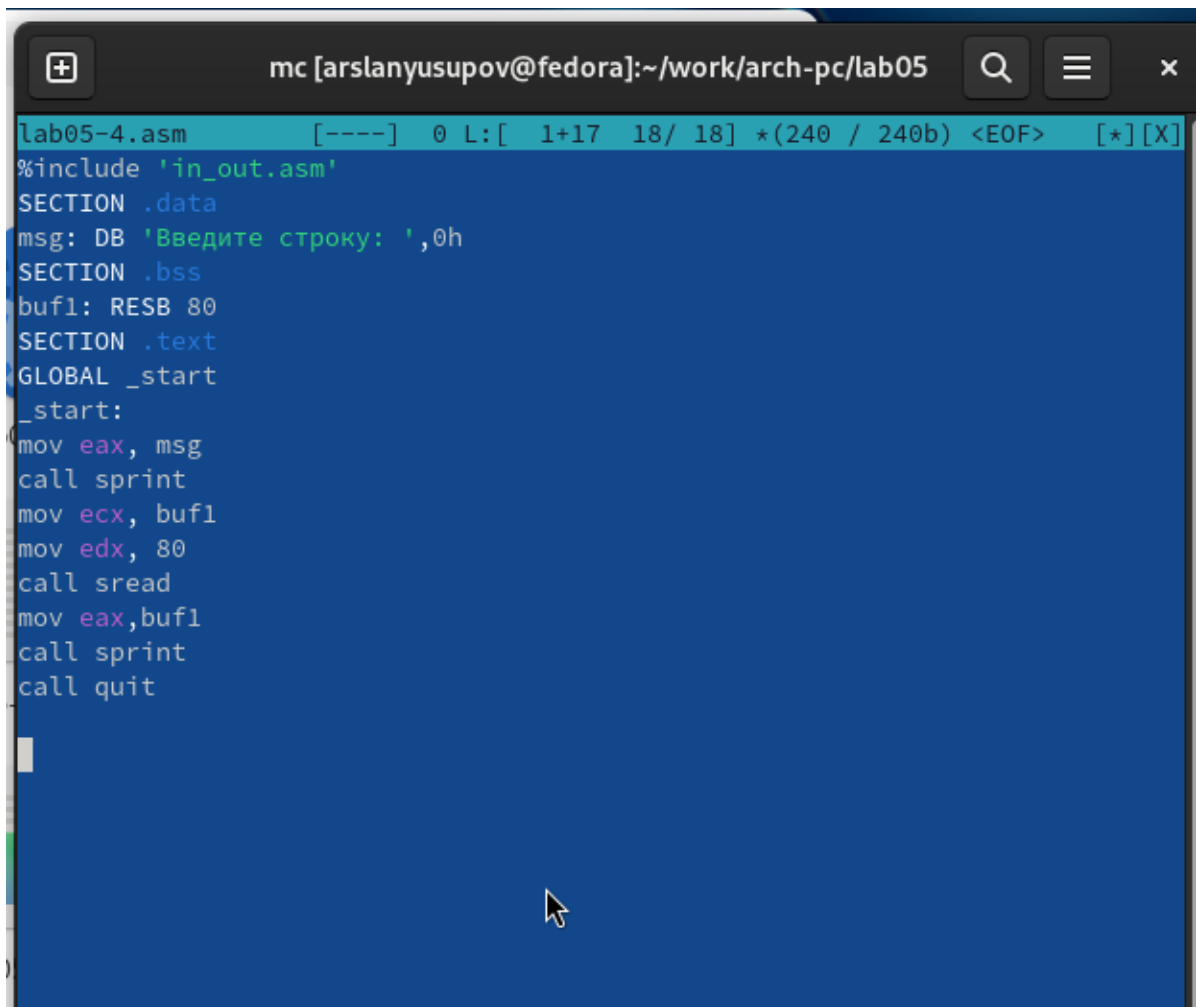
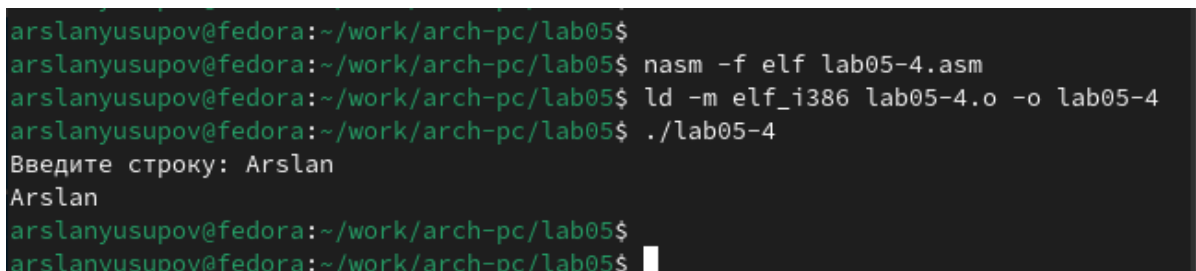


Рис. 2.16: Копирование файла lab05-2.asm



```
lab05-4.asm [----] 0 L: [ 1+17 18/ 18] *(240 / 240b) <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. 2.17: Программа в файле lab05-4.asm



```
arslanyusupov@fedora:~/work/arch-pc/lab05$
arslanyusupov@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-4.asm
arslanyusupov@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-4.o -o lab05-4
arslanyusupov@fedora:~/work/arch-pc/lab05$ ./lab05-4
Введите строку: Arslan
Arslan
arslanyusupov@fedora:~/work/arch-pc/lab05$
arslanyusupov@fedora:~/work/arch-pc/lab05$
```

Рис. 2.18: Запуск программы lab05-4.asm

3 Выводы

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