

# **Отчёт по лабораторной работе 5**

**Основы работы с Midnight Commander. Структура программы на языке  
ассемблера NASM. Системные вызовы в ОС GNU Linux**

Калашникова Дарья Викторовна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Знакомство с Midnight Commander . . . . .	6
2.2	Подключение внешнего файла in_out.asm . . . . .	11
2.3	Задание для самостоятельной работы . . . . .	15
<b>3</b>	<b>Выводы</b>	<b>18</b>

## Список иллюстраций

2.1	Запуск Midnight Commander . . . . .	6
2.2	Создание нового каталога . . . . .	7
2.3	Создание файла lab05-1.asm . . . . .	8
2.4	Выбор редактора mcedit . . . . .	9
2.5	Написание программы lab05-1.asm . . . . .	10
2.6	Просмотр кода lab05-1.asm . . . . .	11
2.7	Тестовый запуск lab05-1.asm . . . . .	11
2.8	Перемещение файла in_out.asm . . . . .	12
2.9	Создание копии файла lab05-1.asm . . . . .	13
2.10	Добавление подпрограмм в lab05-2.asm . . . . .	14
2.11	Запуск программы lab05-2.asm . . . . .	14
2.12	Обновлённая версия lab05-2.asm . . . . .	15
2.13	Запуск обновлённой lab05-2.asm . . . . .	15
2.14	Редактирование программы lab05-3.asm . . . . .	16
2.15	Тестовый запуск lab05-3.asm . . . . .	16
2.16	Обновлённая программа lab05-4.asm . . . . .	17
2.17	Тестовый запуск 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

### 2.1 Знакомство с Midnight Commander

Запускаю Midnight Commander (рис. 2.1), используя клавиши со стрелками и Enter, перехожу в каталог ~/work/arch-рс. Затем создаю новый каталог под названием lab05, нажав F7 (рис. 2.2).

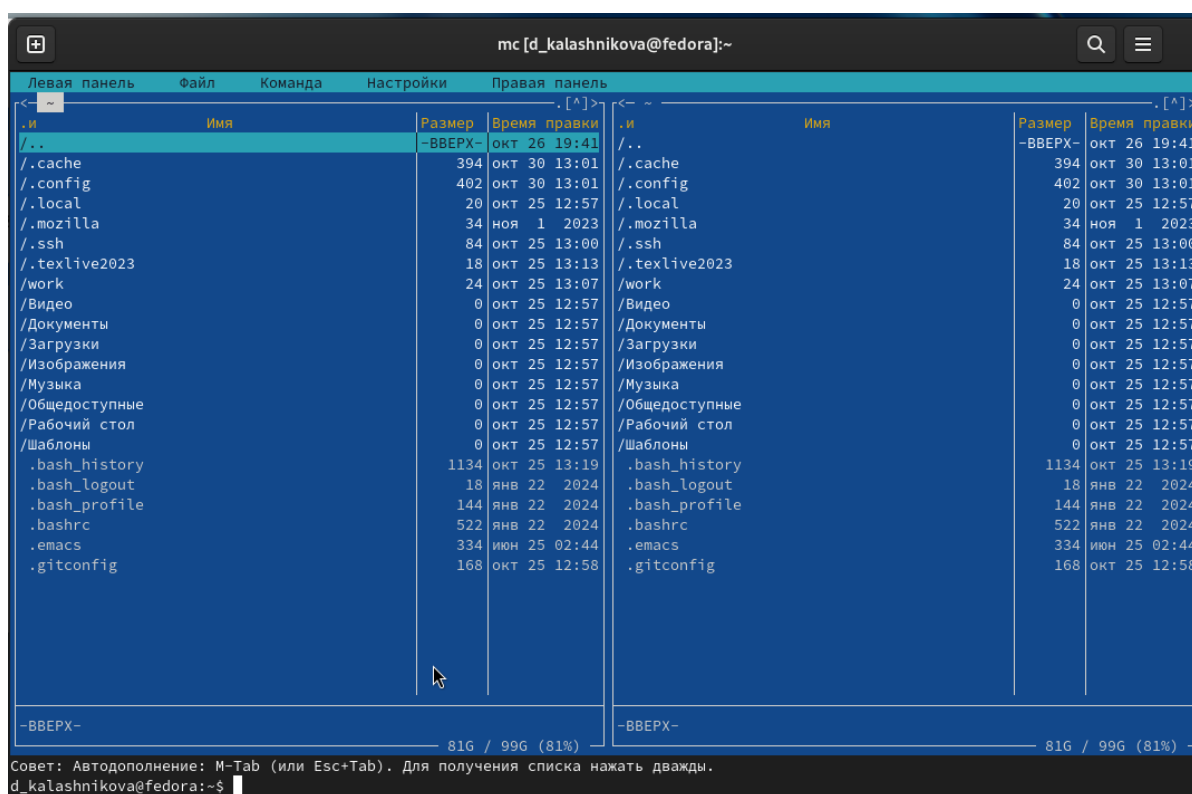


Рис. 2.1: Запуск Midnight Commander

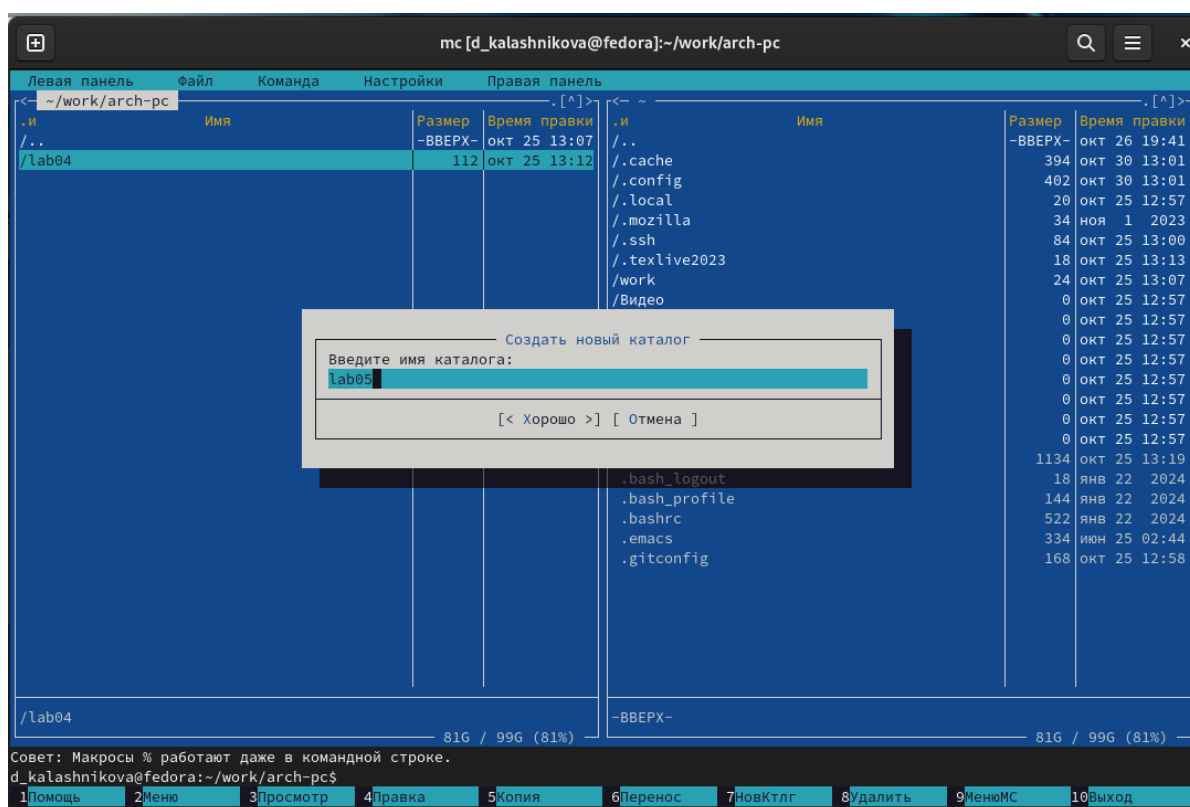


Рис. 2.2: Создание нового каталога

Использую команду `touch`, чтобы создать файл `lab05-1.asm` (рис. 2.3).

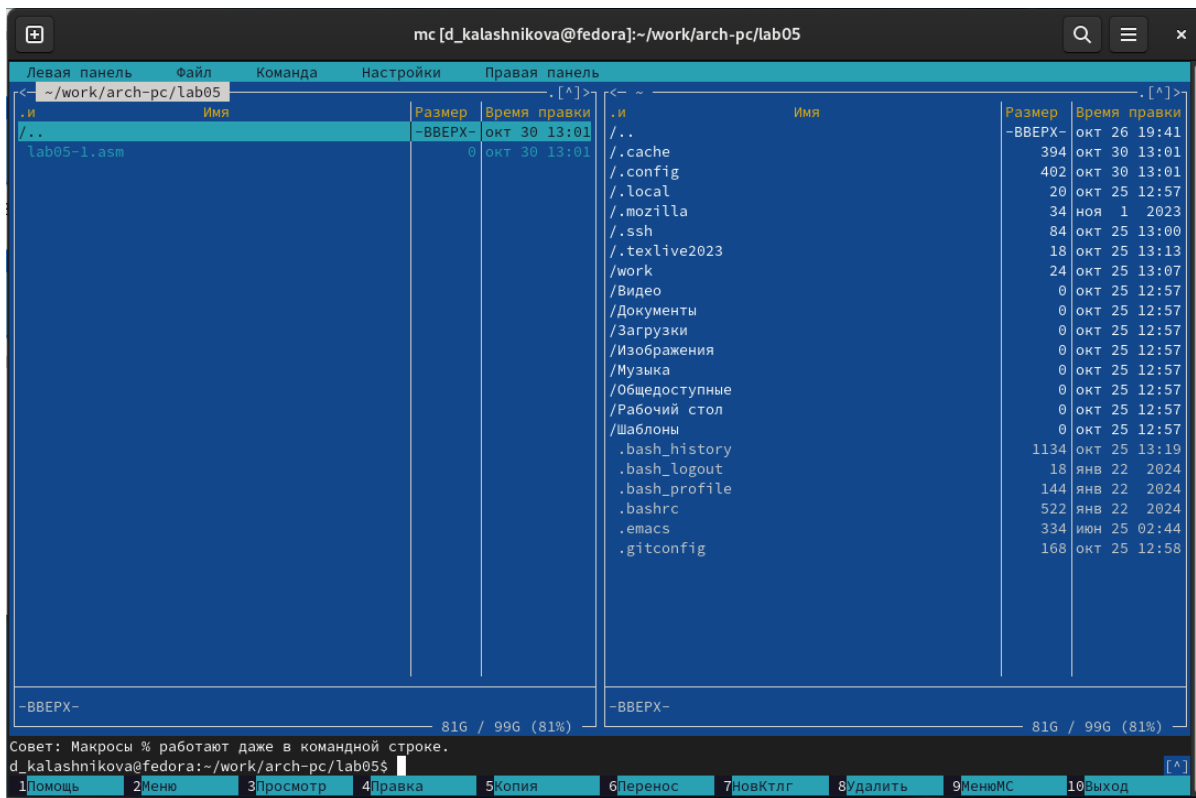


Рис. 2.3: Создание файла lab05-1.asm

Открываю файл для редактирования с помощью F4, выбираю редактор mcedit, и пишу программу в соответствии с заданием (рис. 2.5).



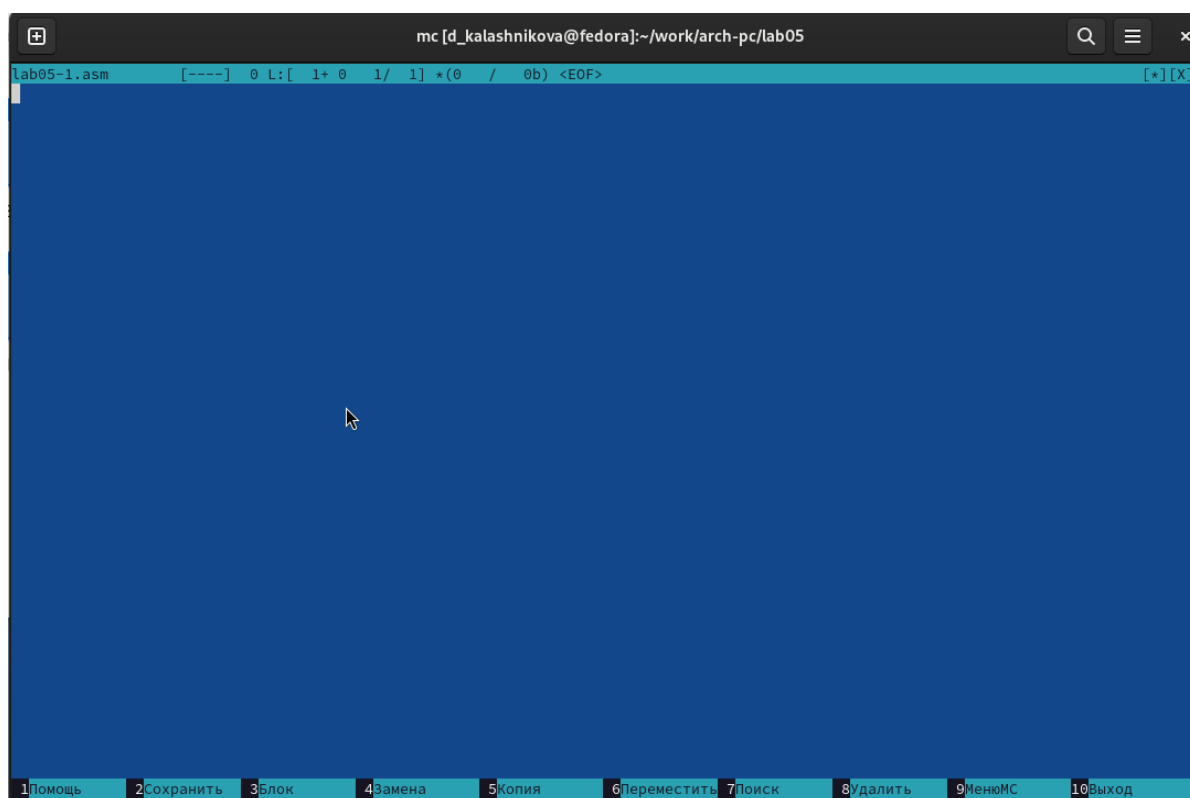
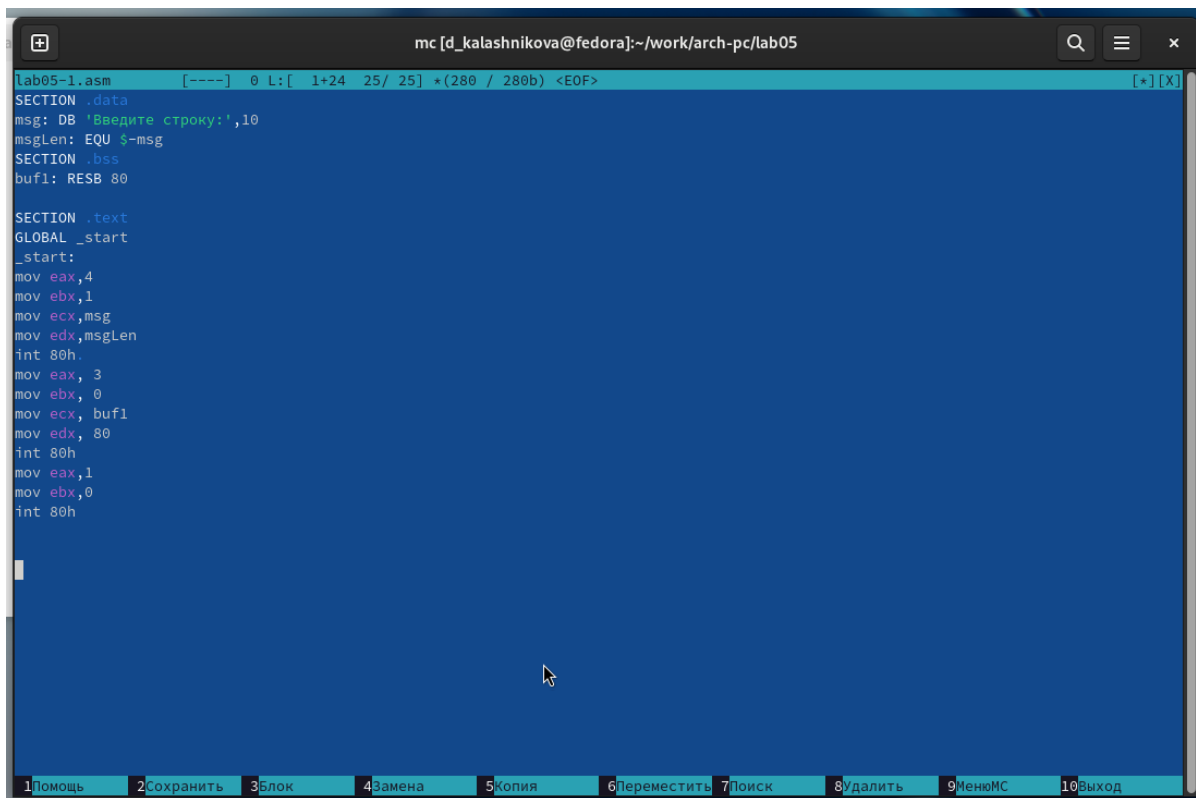


Рис. 2.4: Выбор редактора mcedit



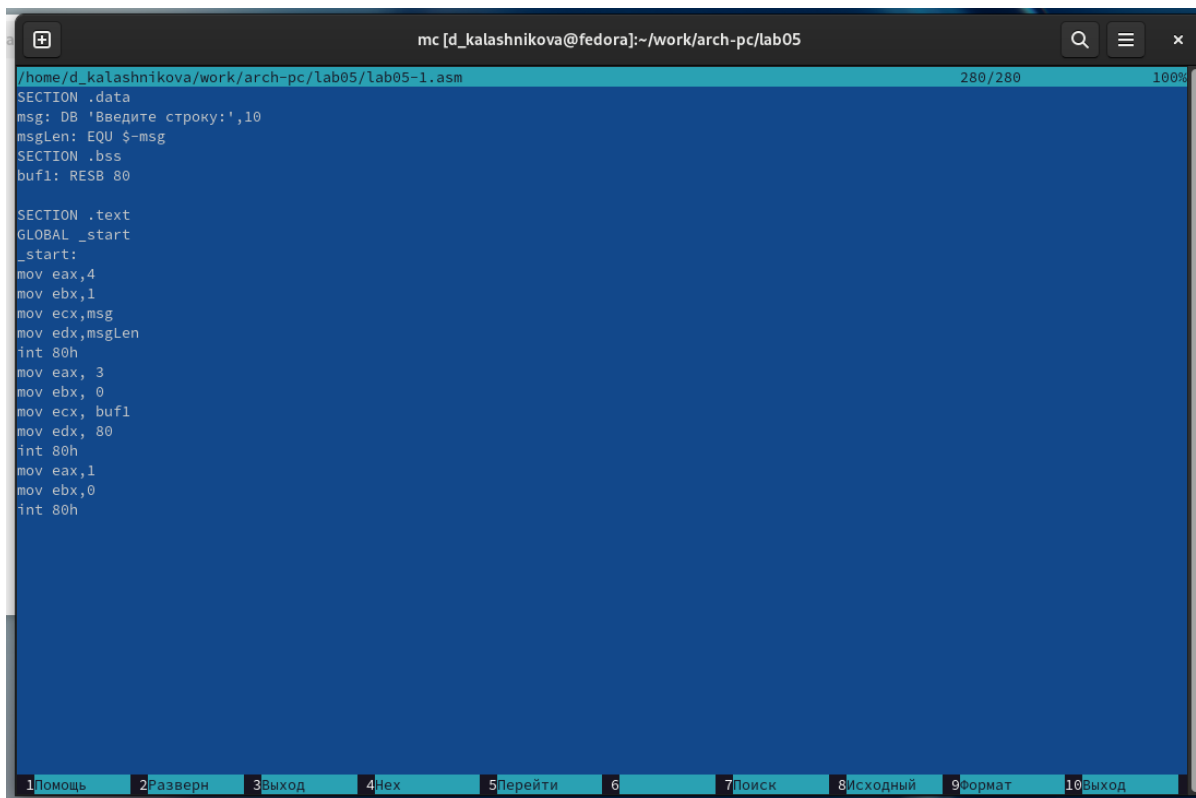
```
lab05-1.asm [----] 0 L: [ 1+24 25/ 25] *(280 / 280b) <EOF> [*] [X]
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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

1 Помощь 2 Сохранить 3 Блок 4 Замена 5 Копия 6 Переместить 7 Поиск 8 Удалить 9 МенюМС 10 Выход
```

Рис. 2.5: Написание программы lab05-1.asm

Для проверки содержимого файла просматриваю его с помощью F3 и убеждаюсь в корректности написанного кода (рис. 2.6).

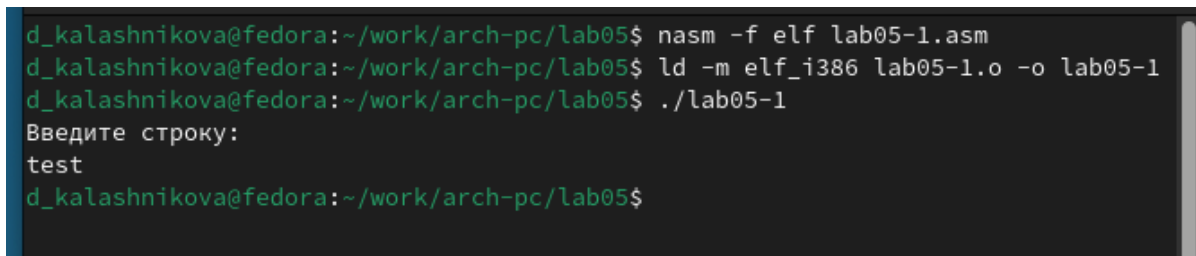


```
mc [d_kalashnikova@fedora]:~/work/arch-pc/lab05
/home/d_kalashnikova/work/arch-pc/lab05/lab05-1.asm 280/280 100%
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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. 2.6: Просмотр кода lab05-1.asm

Транслирую исходный код в объектный файл, затем выполняю компоновку для создания исполняемого файла программы (рис. 2.7).



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

Рис. 2.7: Тестовый запуск lab05-1.asm

## 2.2 Подключение внешнего файла in\_out.asm

Скачиваю файл in\_out.asm и переношу его в рабочий каталог (рис. 2.8). Для копирования использую F5, для перемещения — F6.

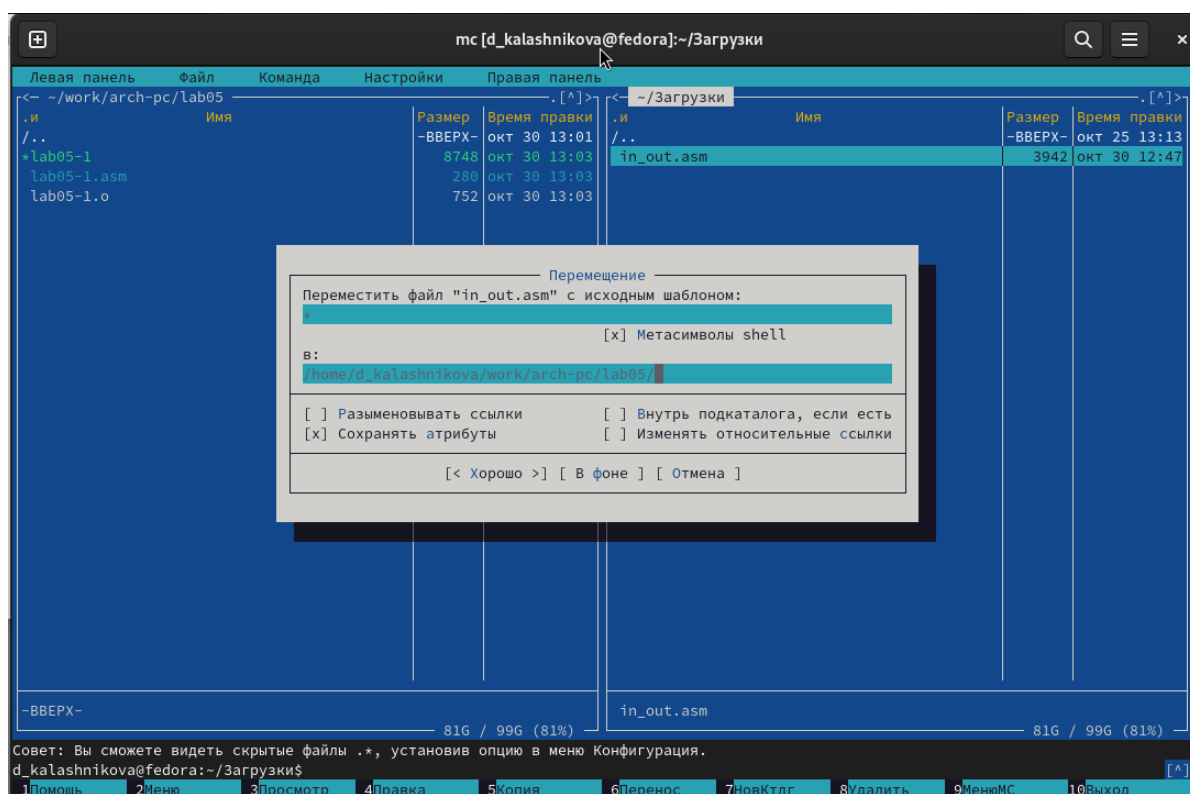


Рис. 2.8: Перемещение файла in\_out.asm

Копирую файл lab05-1.asm и создаю его копию под именем lab05-2.asm (рис. 2.9).

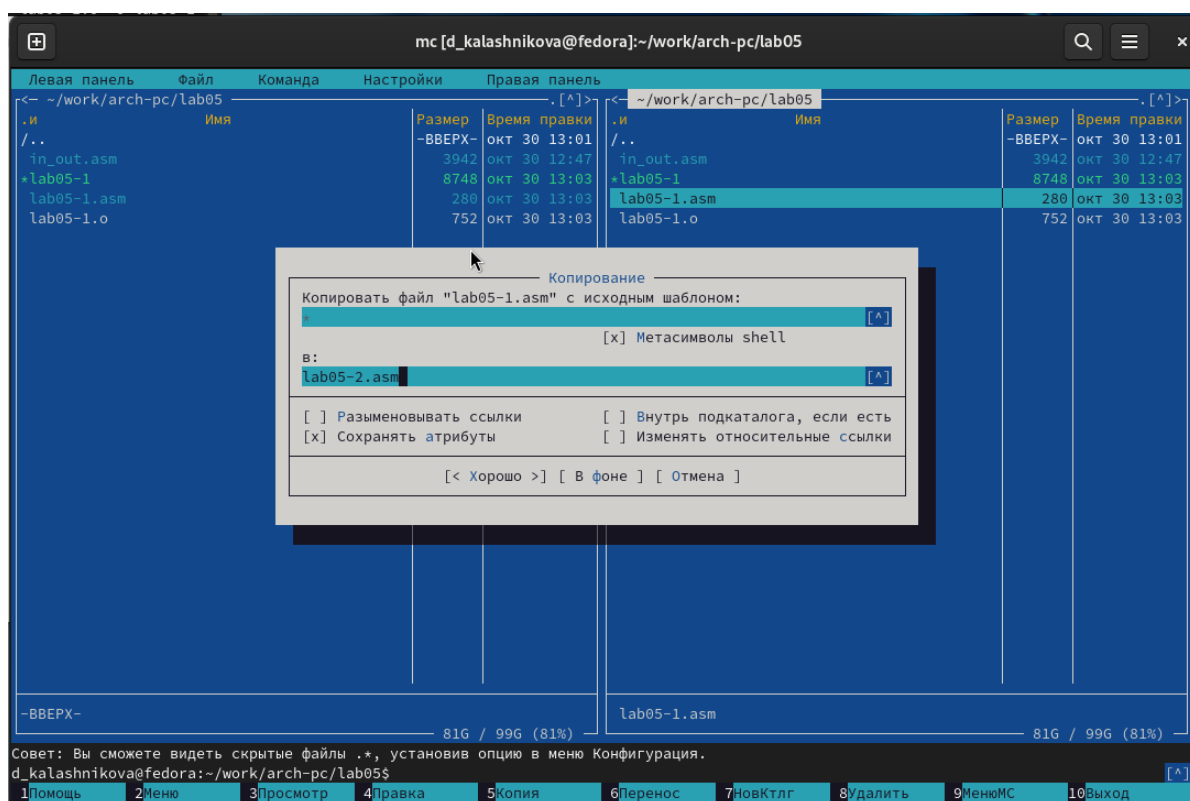
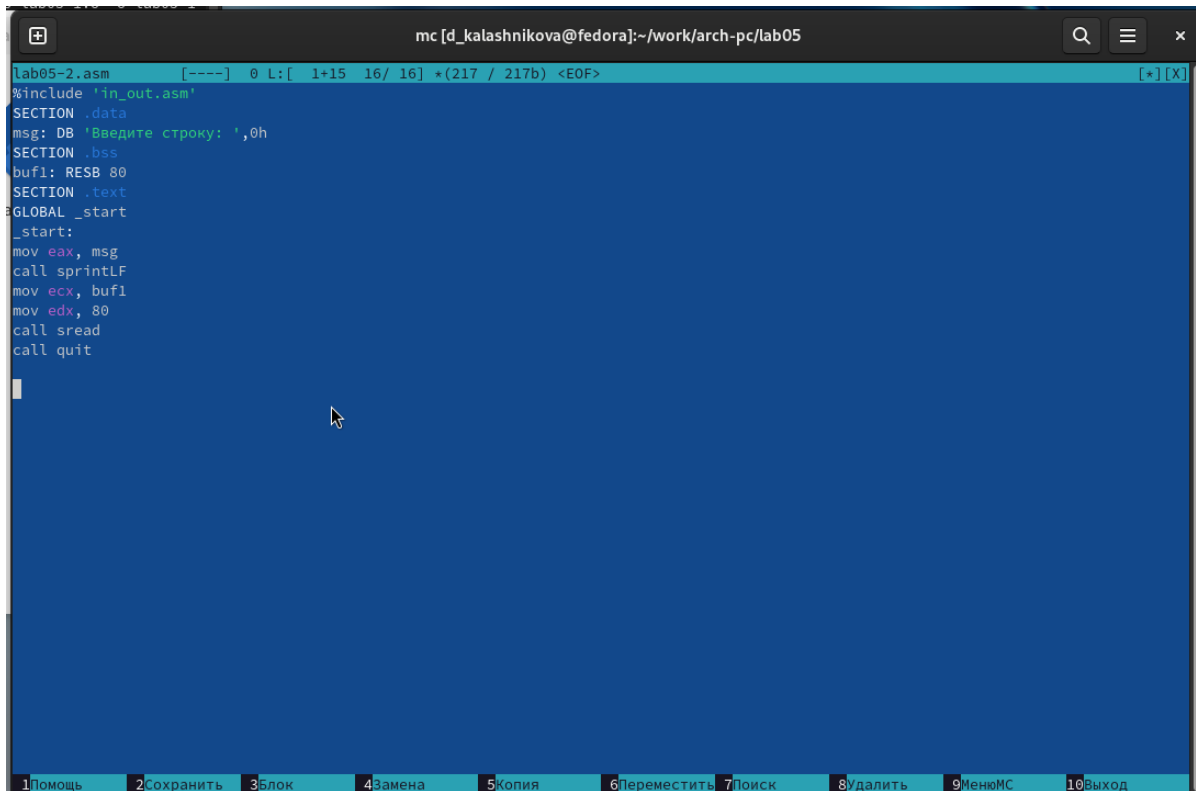


Рис. 2.9: Создание копии файла lab05-1.asm

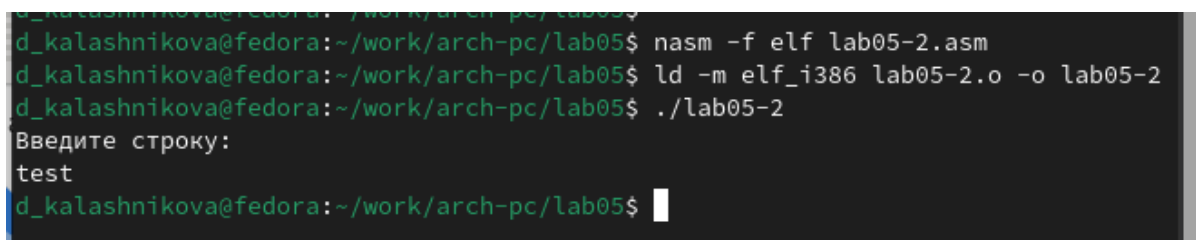
Редактирую код lab05-2.asm, добавляя подпрограммы из файла in\_out.asm (рис. 2.10).



```
lab05-2.asm  [----]  0  L: [ 1+15 16/ 16] *(217 / 217b) <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. 2.10: Добавление подпрограмм в lab05-2.asm

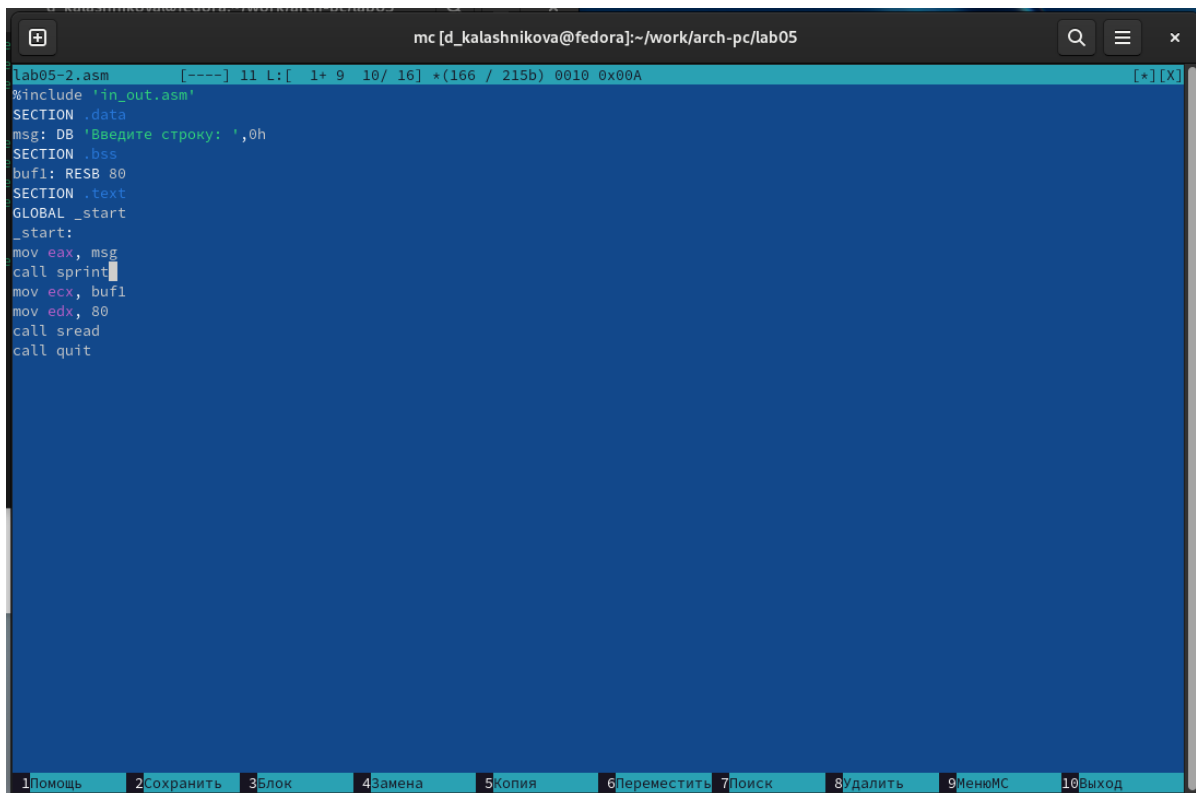
Компилирую программу и проверяю её выполнение (рис. 2.11).



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

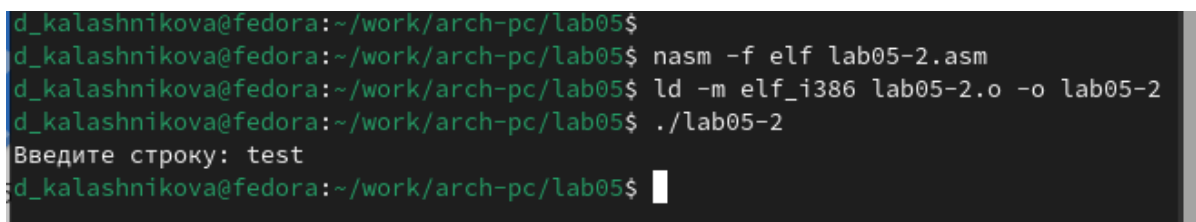
Рис. 2.11: Запуск программы lab05-2.asm

В файле lab05-2.asm заменяю подпрограмму sprintLF на sprint, после чего повторно собираю программу (рис. 2.12 и 2.13).



```
lab05-2.asm [---] 11 L: [ 1+ 9 10/ 16] *(166 / 215b) 0010 0x00A [~][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. 2.12: Обновлённая версия lab05-2.asm



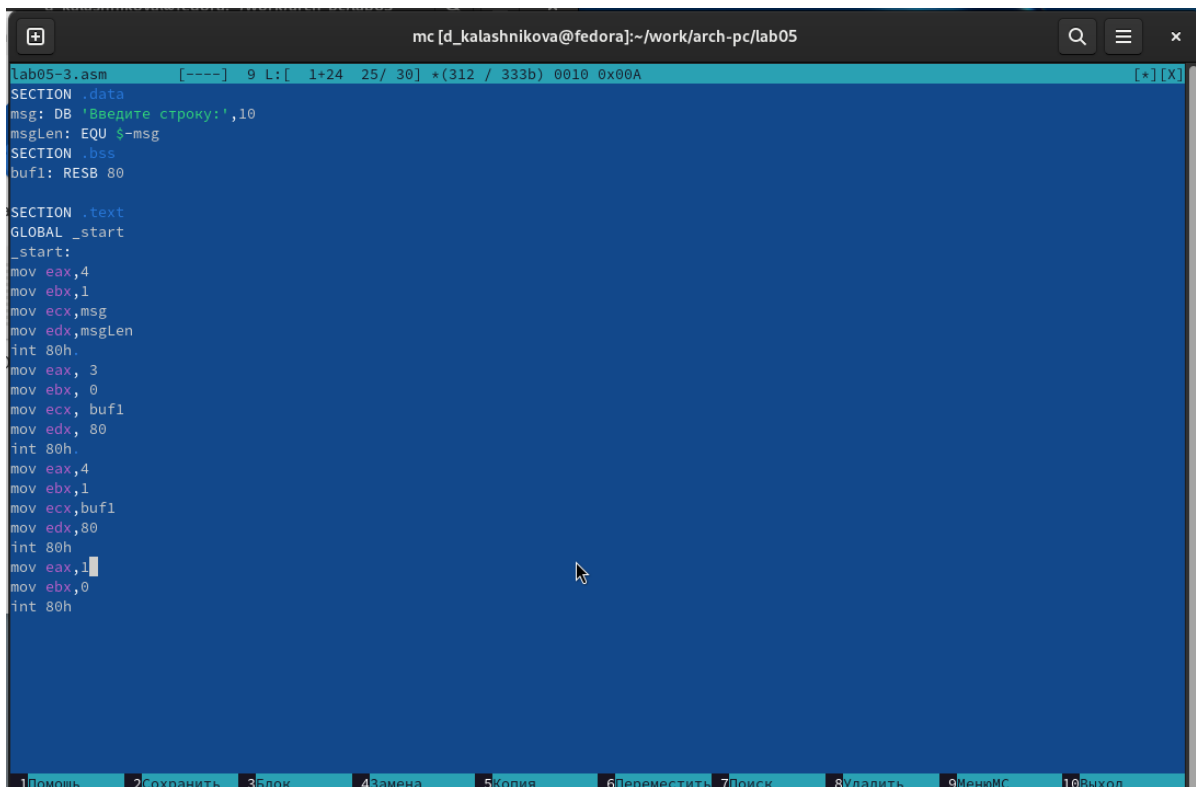
```
d_kalashnikova@fedora:~/work/arch-pc/lab05$
d_kalashnikova@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
d_kalashnikova@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
d_kalashnikova@fedora:~/work/arch-pc/lab05$ ./lab05-2
Введите строку: test
d_kalashnikova@fedora:~/work/arch-pc/lab05$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

Рис. 2.13: Запуск обновлённой lab05-2.asm

Теперь программа выводит строку без переноса на новую строку.

## 2.3 Задание для самостоятельной работы

Создаю копию программы lab05-1.asm и изменяю код для выполнения следующего алгоритма (рис. 2.14 и 2.15): - отображает запрос “Введите строку:”; - принимает строку с клавиатуры; - выводит введенную строку на экран.

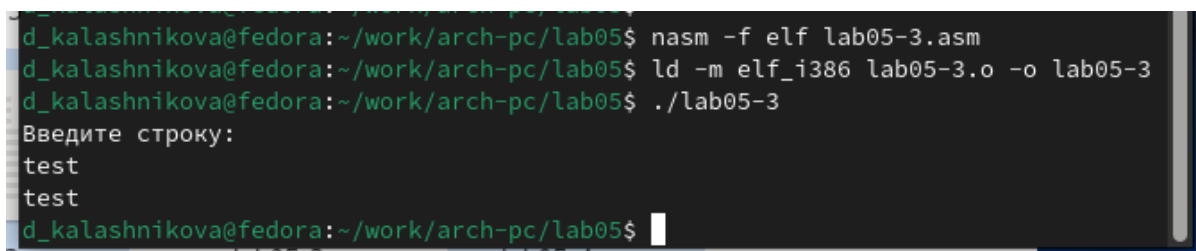


```
mc [d_kalashnikova@fedora]:~/work/arch-pc/lab05
lab05-3.asm [----] 9 L: [ 1+24 25/ 30] *(312 / 333b) 0010 0x00A [*] [X]
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Переместить 7Поиск 8Удалить 9МенюMC 10Выход
```

Рис. 2.14: Редактирование программы lab05-3.asm

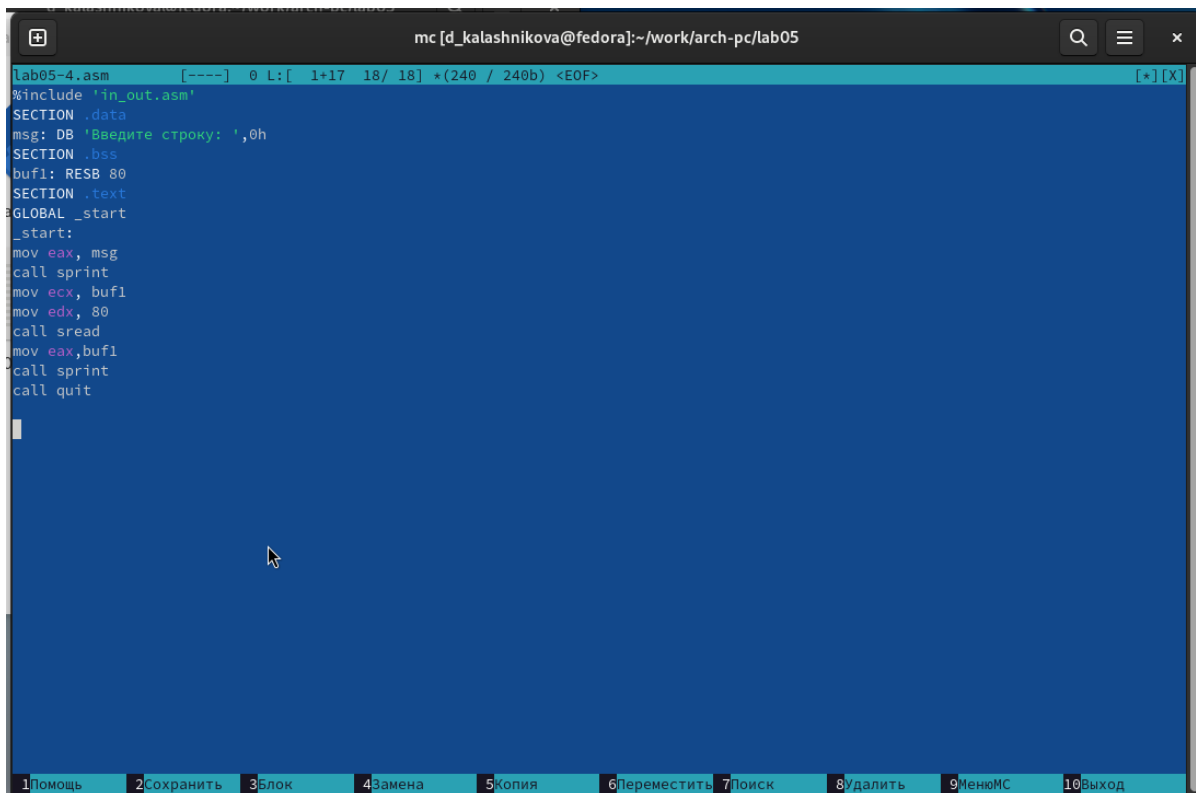


```
d_kalashnikova@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-3.asm
d_kalashnikova@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3
d_kalashnikova@fedora:~/work/arch-pc/lab05$ ./lab05-3
Введите строку:
test
test
d_kalashnikova@fedora:~/work/arch-pc/lab05$
```

Рис. 2.15: Тестовый запуск lab05-3.asm

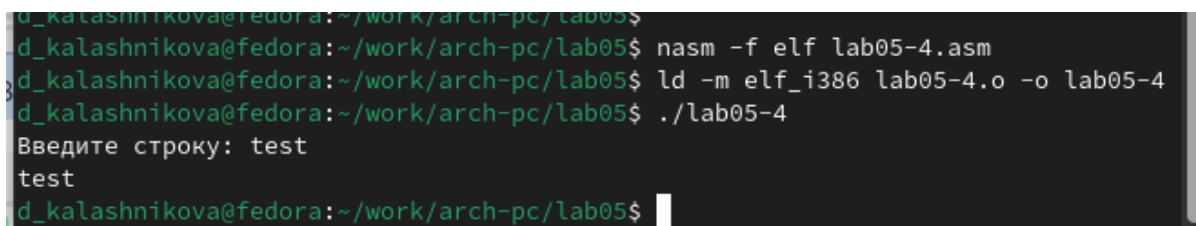
Аналогично, копирую lab05-2.asm и изменяю код, добавляя подпрограммы из файла in\_out.asm (рис. 2.16 и 2.17).





```
lab05-4.asm  [----]  0 L: [ 1+17 18/ 18] *(240 / 240b) <EOF>
#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. 2.16: Обновлённая программа lab05-4.asm



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

Рис. 2.17: Тестовый запуск lab05-4.asm

## 3 Выводы

В ходе работы освоены базовые навыки написания ассемблерных программ, а также инструкции `mov` и `int`.