

# **Отчёт по лабораторной работе №5**

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

Юсупова Ксения Равиловна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Порядок выполнения лабораторной работы . . . . .	6
2.2	Задания для самостоятельной работы . . . . .	14
<b>3</b>	<b>Выводы</b>	<b>19</b>

# Список иллюстраций

2.1	Открыли Midnight Commander с помощью команды mc . . . . .	6
2.2	Перешли в каталог . . . . .	7
2.3	Создали текстовый папку lab05 . . . . .	8
2.4	Воспользовались командой touch и получили файл lab5-1.asm . . .	8
2.5	Открыли файл функциональной клавишей, заполнили и сохранили его . . . . .	9
2.6	Проверили, что файл содержит текст программы . . . . .	10
2.7	Проверили, что данная программа работает . . . . .	10
2.8	Скачали файл со страницы курса . . . . .	11
2.9	Скопировали файл в каталог . . . . .	11
2.10	Создали копию файла lab5-1.asm . . . . .	12
2.11	Открывали и заполнили файл . . . . .	13
2.12	Проверили работу программы . . . . .	13
2.13	Отредактировали файл . . . . .	14
2.14	Проверили работу программы . . . . .	14
2.15	Создали копию файла lab5-1.asm . . . . .	15
2.16	Внесли изменения в программу . . . . .	16
2.17	Проверили работу программы . . . . .	16
2.18	Создали копию файла lab5-2.asm . . . . .	17
2.19	Внесли изменения в программу . . . . .	18
2.20	Проверили работу программы . . . . .	18

## **Список таблиц**

# 1 Цель работы

Приобрести практические навыки работы в Midnight Commander и освоить инструкции языка ассемблера `mov` и `int`.

# 2 Выполнение лабораторной работы

## 2.1 Порядок выполнения лабораторной работы

Открываем Midnight Commander (рис. 2.1).

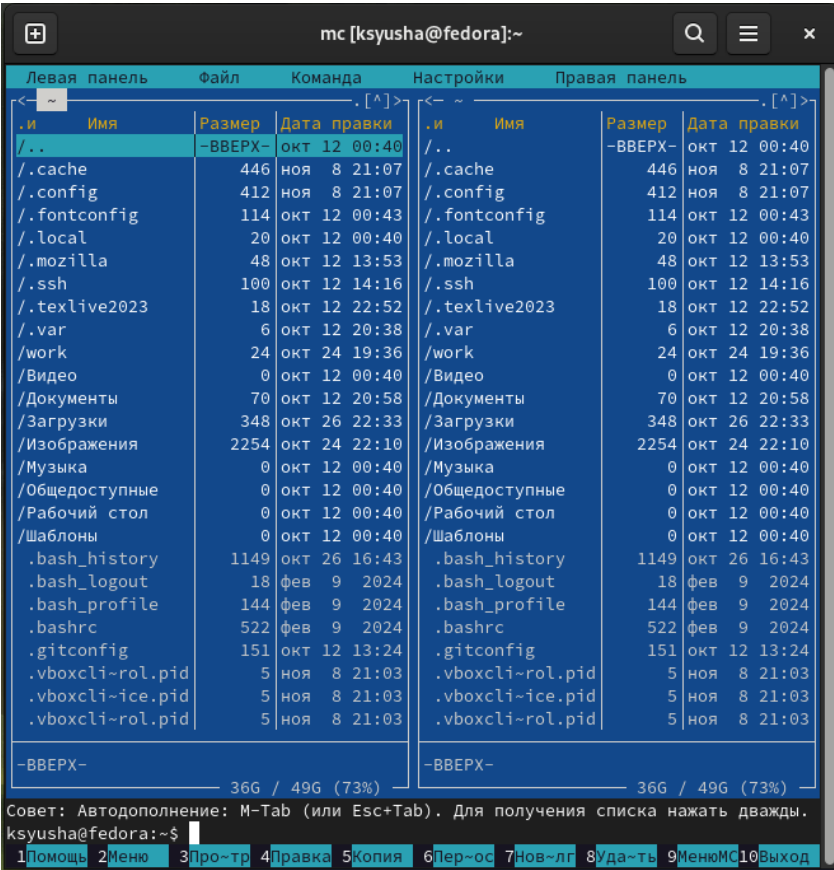


Рис. 2.1: Открываем Midnight Commander с помощью команды mc

Пользуясь клавишами  $\uparrow$ ,  $\downarrow$  и Enter переходим в каталог ~/work/arch-pc созданный при выполнении лабораторной работы №4 (рис. 2.2).

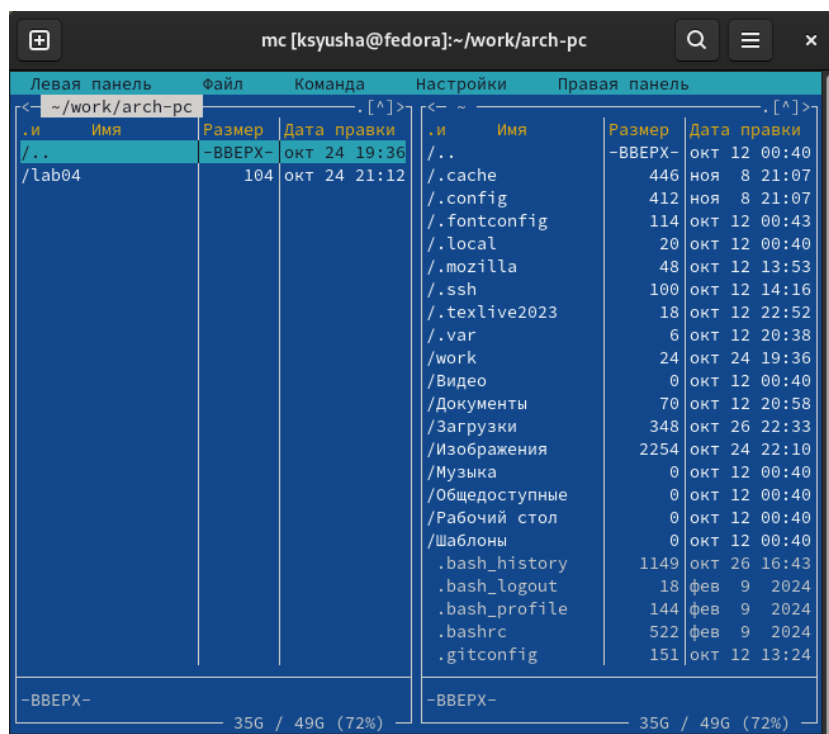


Рис. 2.2: Перешли в каталог

С помощью функциональной клавиши F7 создаём папку lab05 и переходим в созданный каталог (рис. 2.3).

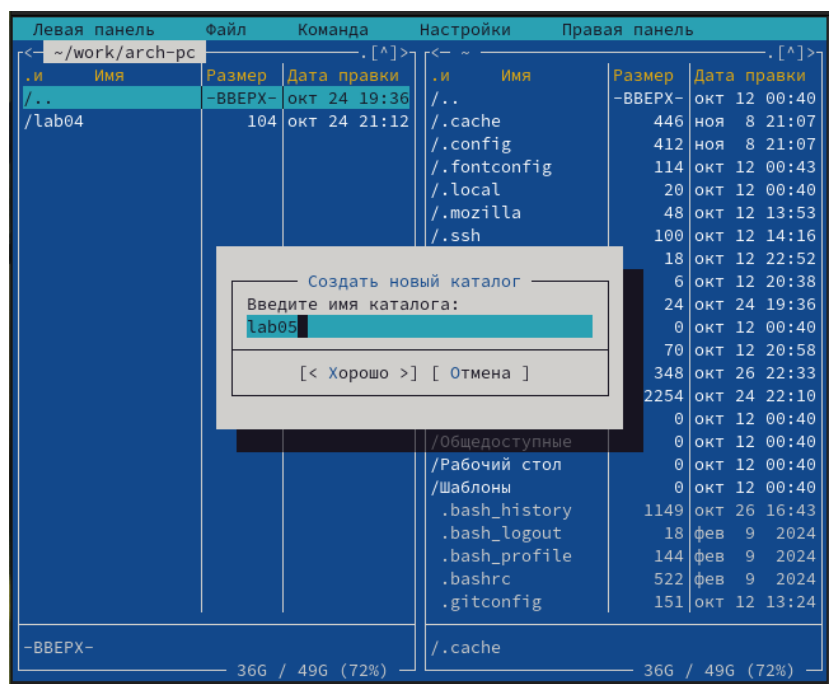


Рис. 2.3: Создали текстовый папку lab05

Пользуясь строкой ввода и командой `touch` создаём файл `lab5-1.asm` (рис. 2.4).

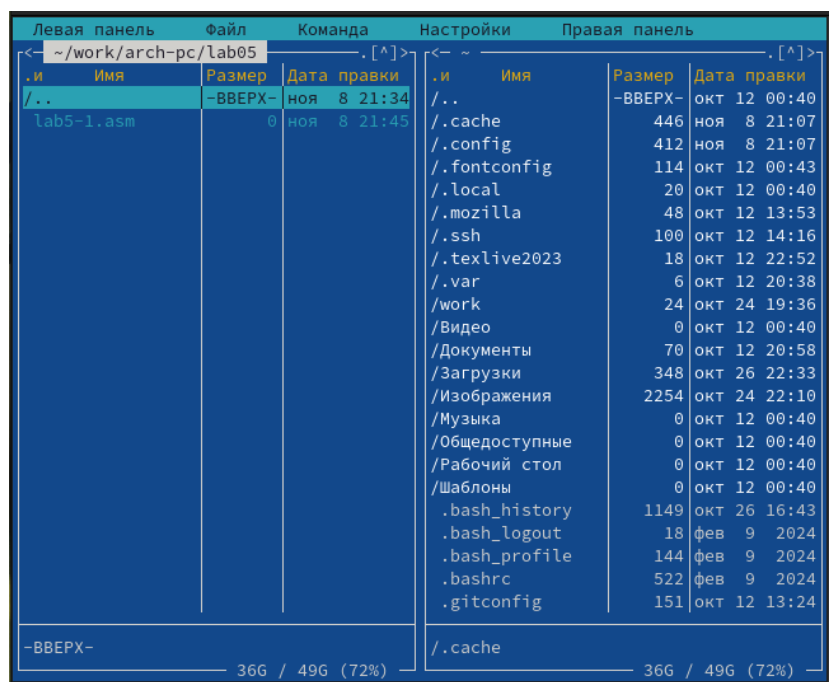
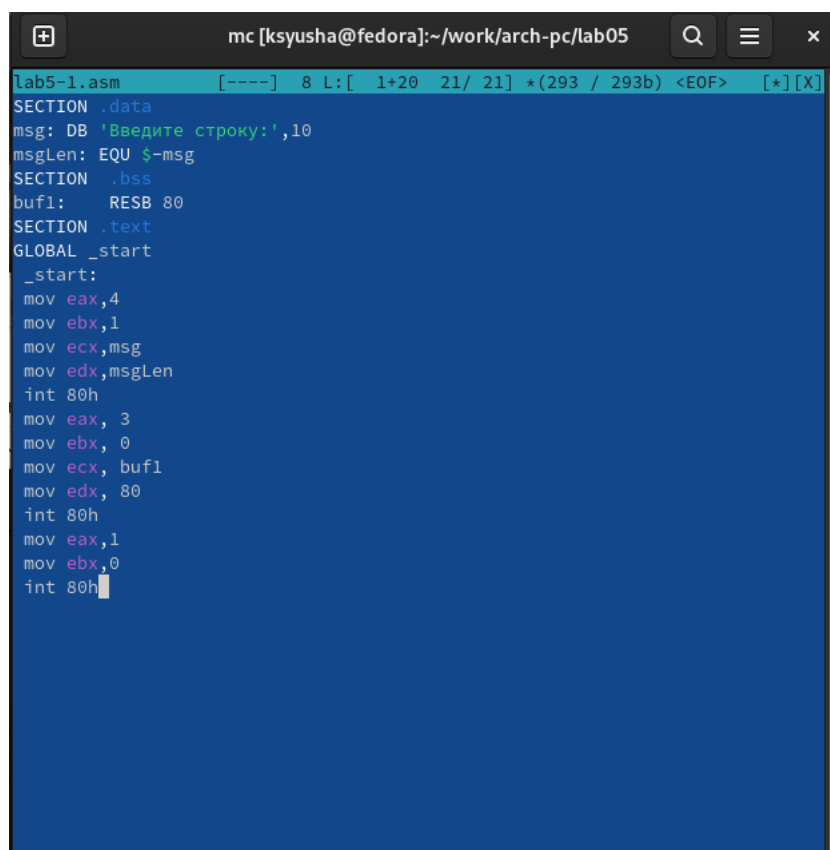


Рис. 2.4: Воспользовались командой `touch` и получили файл `lab5-1.asm`



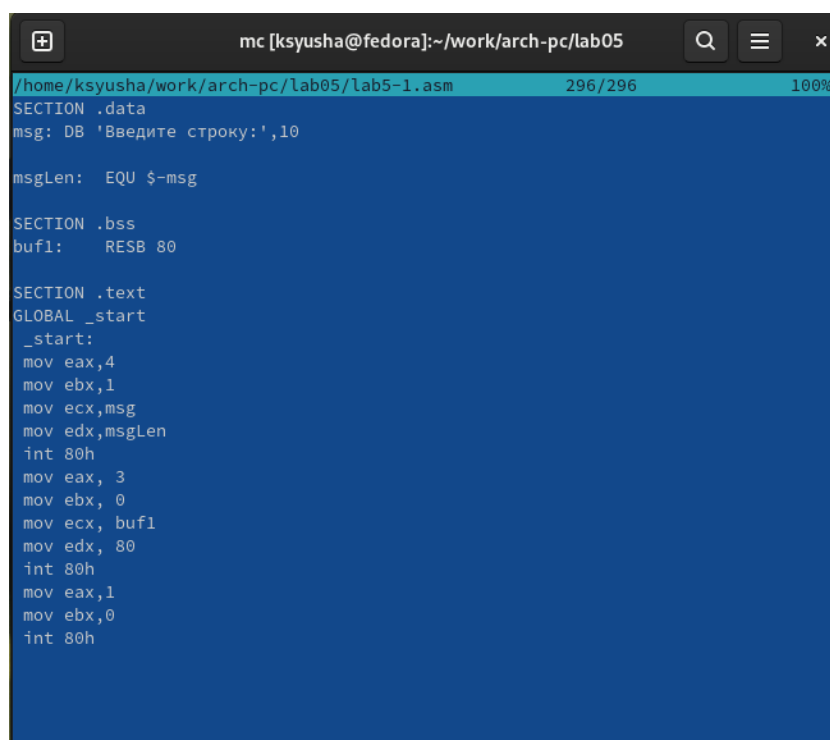
С помощью функциональной клавиши F4 откроем файл lab5-1.asm для редактирования во встроенном редакторе, в качестве встроенного редактора Midnight Commander используем редактор mcedit. Редактируем файл, вводя программы ((рис. 2.5).



```
lab5-1.asm [----] 8 L:[ 1+20 21/ 21] *(293 / 293b) <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. 2.5: Открыли файл функциональной клавишей, заполнили и сохранили его

Открываем файл lab5-1.asm функциональной клавишей F3 для перепроверки(рис. 2.6).

A screenshot of a code editor window titled 'mc [ksyusha@fedora]:~/work/arch-pc/lab05'. The editor displays the assembly code for 'lab5-1.asm'. The code is as follows:

```
/home/ksyusha/work/arch-pc/lab05/lab5-1.asm 296/296 100%
SECTION .data
msg: DB 'Введите строку:',10

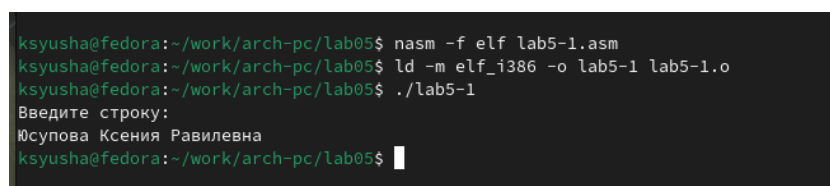
msgLen: EQU $-msg

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: Проверили, что файл содержит текст программы

Оттранслируем текст программы lab5-1.asm в объектный файл. Выполняем компоновку объектного файла и запускаем получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос вводим ФИО(рис. 2.7).

A screenshot of a terminal window showing the compilation and execution of the assembly program. The commands and output are as follows:

```
ksyusha@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
ksyusha@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
ksyusha@fedora:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Юсупова Ксения Равиловна
ksyusha@fedora:~/work/arch-pc/lab05$
```

Рис. 2.7: Проверили, что данная программа работает

Скачиваем файл со страницы курса (рис. 2.8).

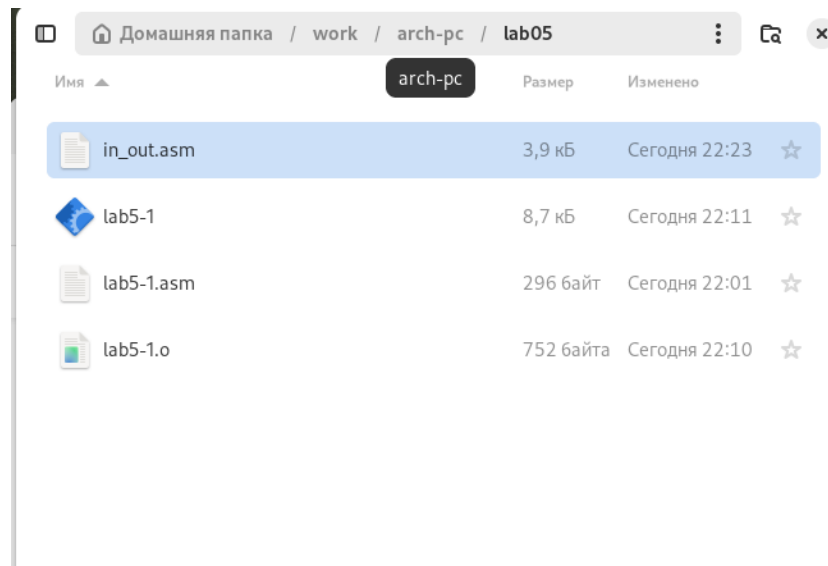


Рис. 2.8: Скачали файл со страницы курса

Копируем файл в каталог с программой (рис. 2.9).

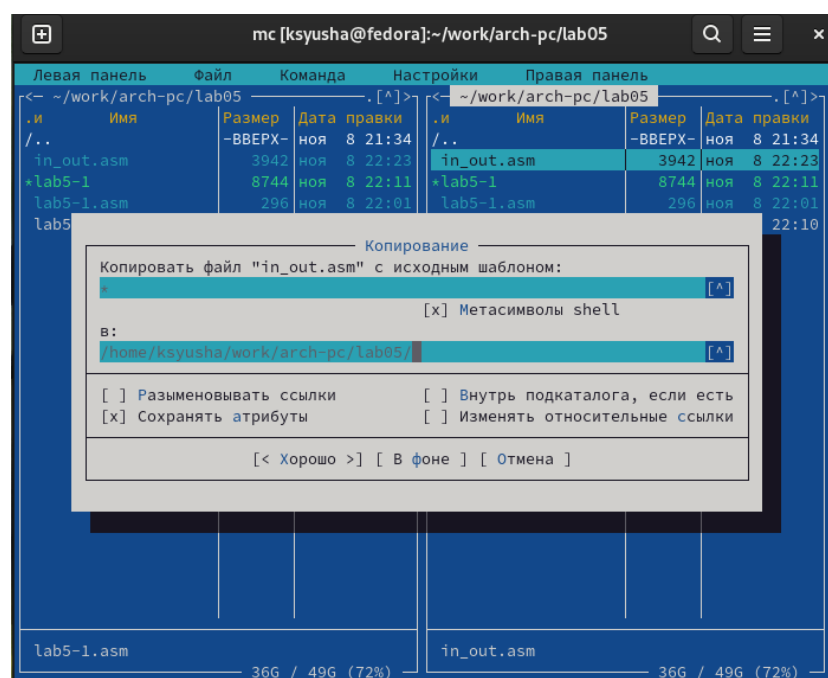


Рис. 2.9: Скопировали файл в каталог

С помощью функциональной клавиши F6 создаём копию файла lab5-1.asm именем lab5-2.asm. Выделяем файл lab5-1.asm, нажимаем клавишу F6, вводим имя файла lab5-2.asm и нажимаем клавишу Enter. (рис. 2.10).

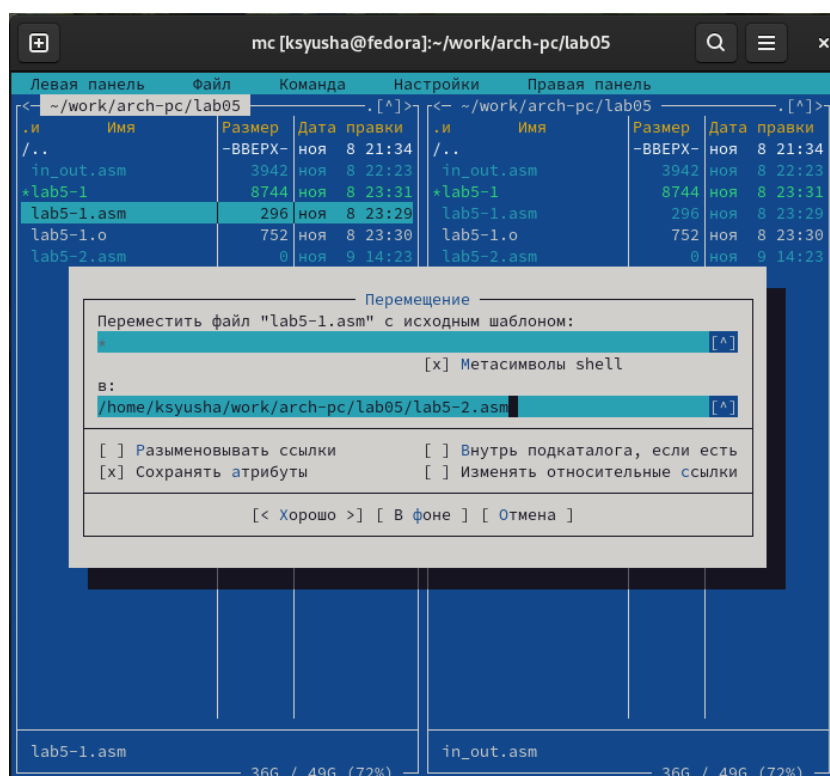
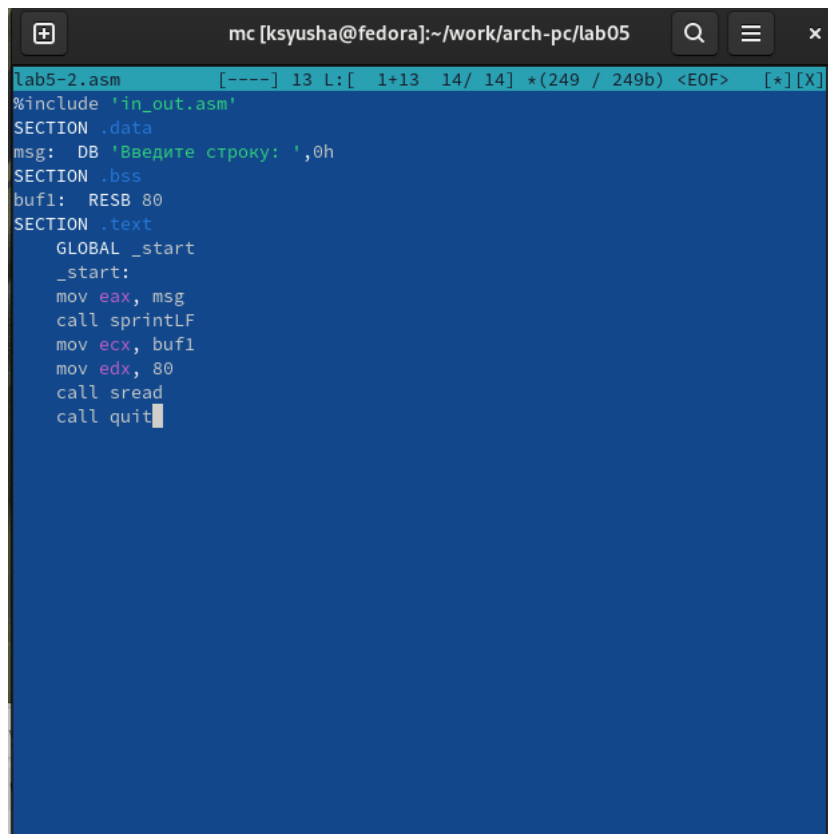


Рис. 2.10: Создали копию файла lab5-1.asm

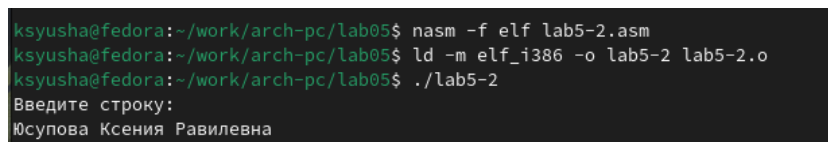
Исправляем текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm(рис. 2.11).



```
lab5-2.asm [----] 13 L: [ 1+13 14/ 14] *(249 / 249b) <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. 2.11: Открывали и заполнили файл

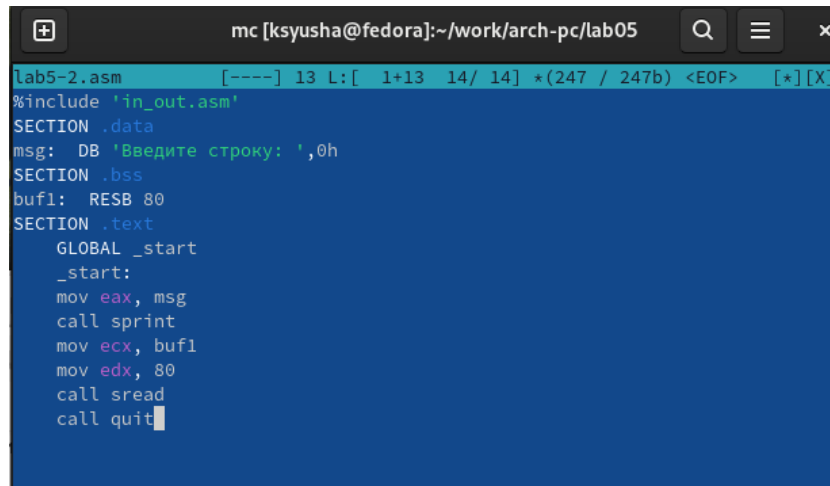
Запускаем программу и проверяем её на работу(рис. 2.12).



```
ksyusha@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
ksyusha@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
ksyusha@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Юсупова Ксения Равиловна
```

Рис. 2.12: Проверили работу программы

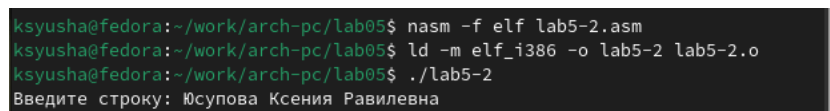
В файле lab5-2.asm заменяем подпрограмму sprintLF на sprint. Создаём исполняемый файл и проверяем его работу.(рис. 2.13).



```
lab5-2.asm [----] 13 L: [ 1+13 14/ 14] *(247 / 247b) <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
call quit
```

Рис. 2.13: Отредактировали файл

Запускаем программу и проверяем её на работу(рис. 2.14).



```
ksyusha@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
ksyusha@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
ksyusha@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Юсупова Ксения Равиловна
```

Рис. 2.14: Проверили работу программы

После полученных программ, можем сделать вывод, что программа `sprintLF` переносит текст на следующую строку, а команда `sprint` оставляет текст на той же строке.

## 2.2 Задания для самостоятельной работы

Создаем копию файла `lab5-1.asm` и называем его `lab5-3.asm`(рис. 2.15).

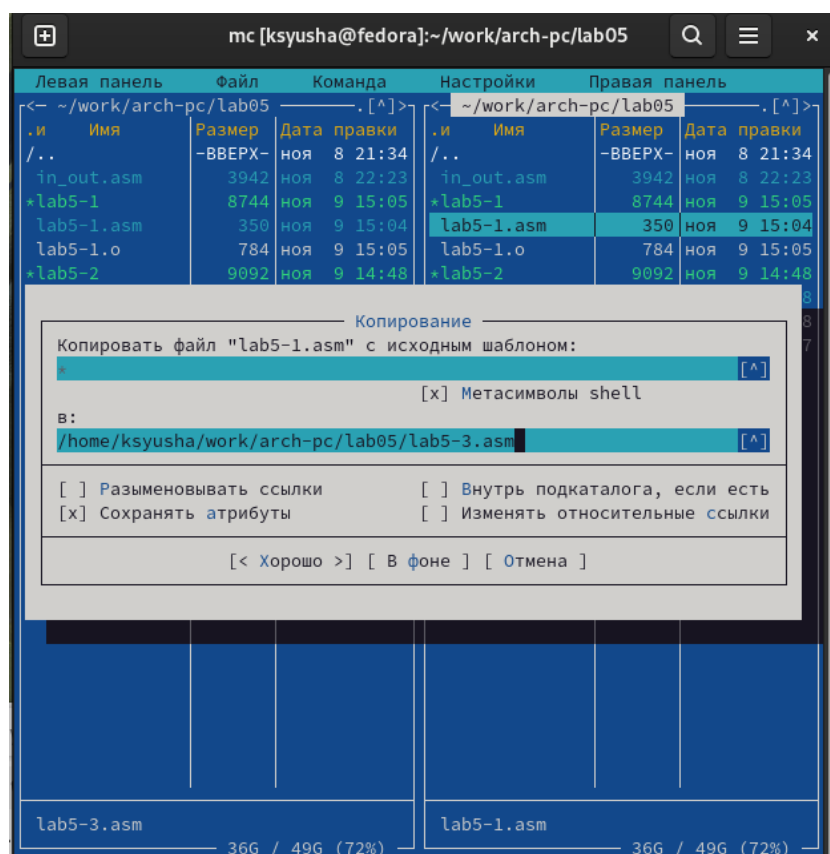
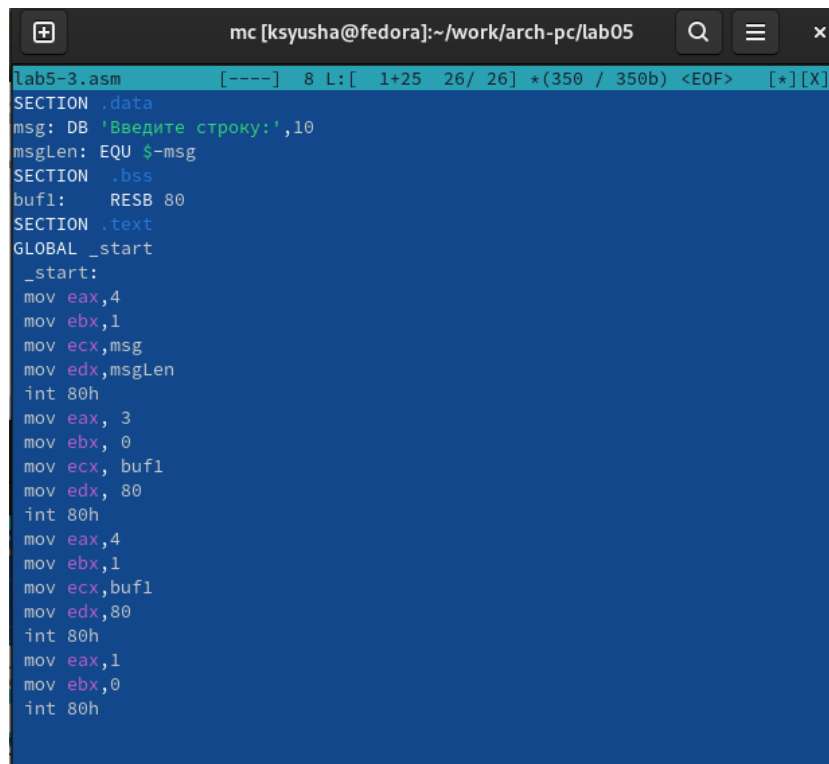


Рис. 2.15: Создали копию файла lab5-1.asm

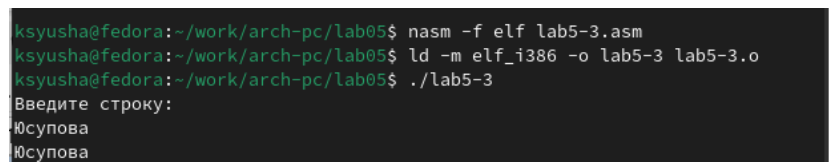
Внесим изменения в программу (без использования внешнего файла in\_out.asm), так чтобы она работала по алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введенную строку на экран. (рис. 2.16).



```
lab5-3.asm [----] 8 L: [ 1+25 26/ 26] *(350 / 350b) <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,4
mov ebx,1
mov ecx,buf1
mov edx,80
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рис. 2.16: Внесли изменения в программу

Запускаем программу и проверяем её работу(рис. 2.17).



```
ksyusha@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-3.asm
ksyusha@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-3 lab5-3.o
ksyusha@fedora:~/work/arch-pc/lab05$ ./lab5-3
Введите строку:
Юсупова
Юсупова
```

Рис. 2.17: Проверили работу программы

Создаем копию файла lab5-2.asm и называем его lab5-4.asm(рис. 2.18).



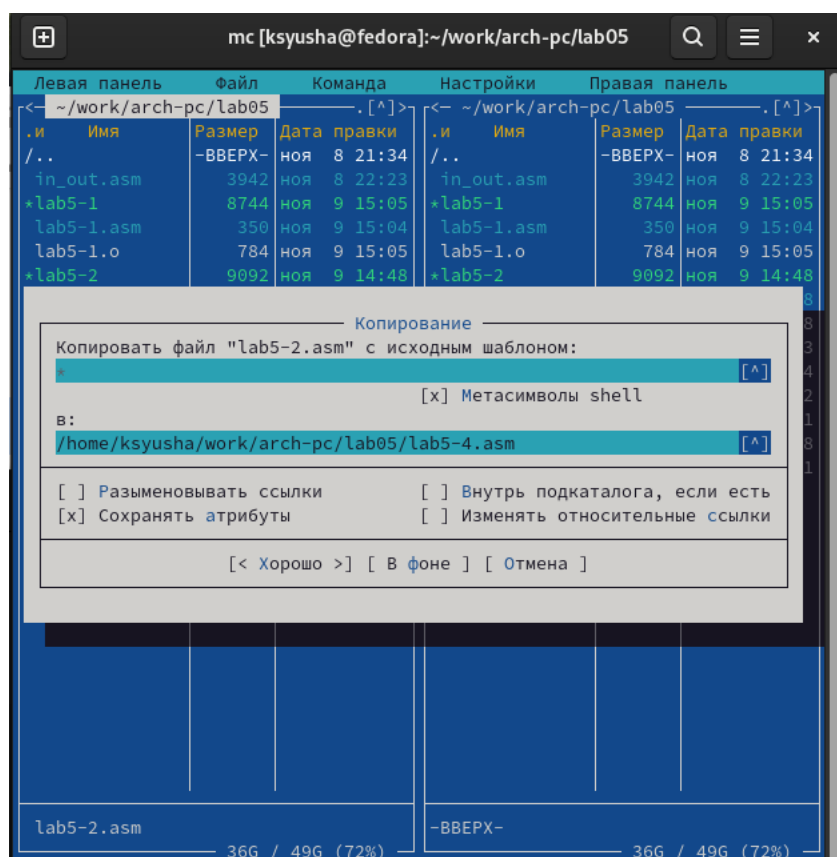
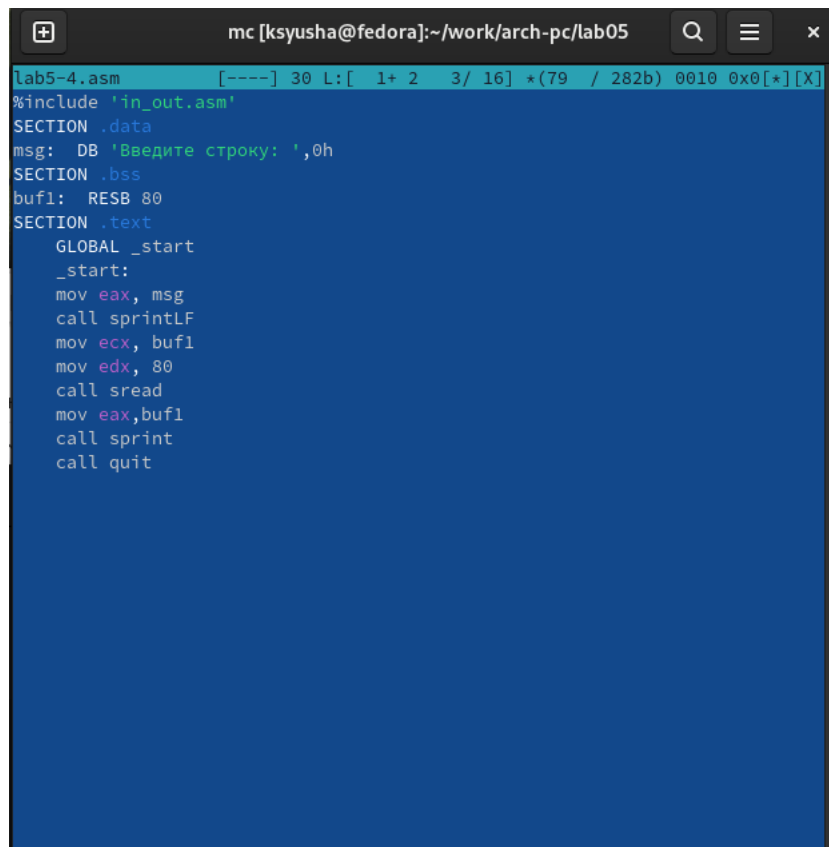


Рис. 2.18: Создали копию файла lab5-2.asm

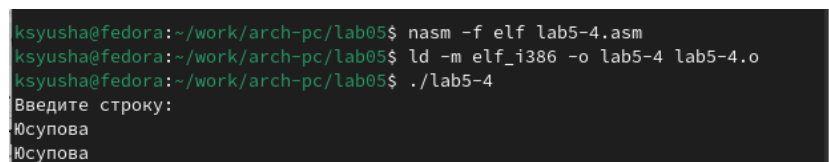
Исправляем текст программы с использованием программ из внешнего файла in\_out.asm, так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введенную строку на экран.(рис. 2.19).



```
lab5-4.asm [----] 30 L: [ 1+ 2 3/ 16] *(79 / 282b) 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 sprintLF
mov ecx, buf1
mov edx, 80
call sread
mov eax, buf1
call sprint
call quit
```

Рис. 2.19: Внесли изменения в программу

Запускаем программу и проверяем её работу (рис. 2.20).



```
ksyusha@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-4.asm
ksyusha@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-4 lab5-4.o
ksyusha@fedora:~/work/arch-pc/lab05$ ./lab5-4
Введите строку:
Юсупова
Юсупова
```

Рис. 2.20: Проверили работу программы

## **3 Выводы**

В ходе лабораторной работы, мы приобрели практические навыки работы в Midnight Commander и освоили инструкции языка ассемблера `mov` и `int`.