

# **Отчёт по лабораторной работе №5**

**Основы работы с Midnight Commander**

Скобеева Алиса Алексеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выполнение самостоятельной работы</b>	<b>15</b>
<b>4</b>	<b>Выводы</b>	<b>18</b>

## Список иллюстраций

2.1	Вводим команду ms . . . . .	6
2.2	Переходим в каталог . . . . .	7
2.3	Используем функциональную клавишу F7 . . . . .	8
2.4	Заполняем файл по листингу . . . . .	9
2.5	Проверяем, как работает данная программа . . . . .	9
2.6	Проверяем наличие файла . . . . .	10
2.7	Создаем копию файла клавишей F6 . . . . .	10
2.8	Проверяем скопированный файл . . . . .	11
2.9	Открываем и заполняем файл . . . . .	12
2.10	Смотрим, как сработала программа . . . . .	13
2.11	Редактируем файл . . . . .	13
2.12	Смотрим, как сработала программа и сравниваем ее с прошлой . .	14
3.1	Создание копии файла . . . . .	15
3.2	Редактируем файл . . . . .	16
3.3	Проверяем правильность написания программы . . . . .	16
3.4	Редактируем файл . . . . .	17
3.5	Проверяем правильность написания программы . . . . .	17

## **Список таблиц**

# 1 Цель работы

Освоить инструкции языка ассемблера mov. Приобрести знания использования Midnight Commander.

# 2 Выполнение лабораторной работы

Открываем Midnight Commander:

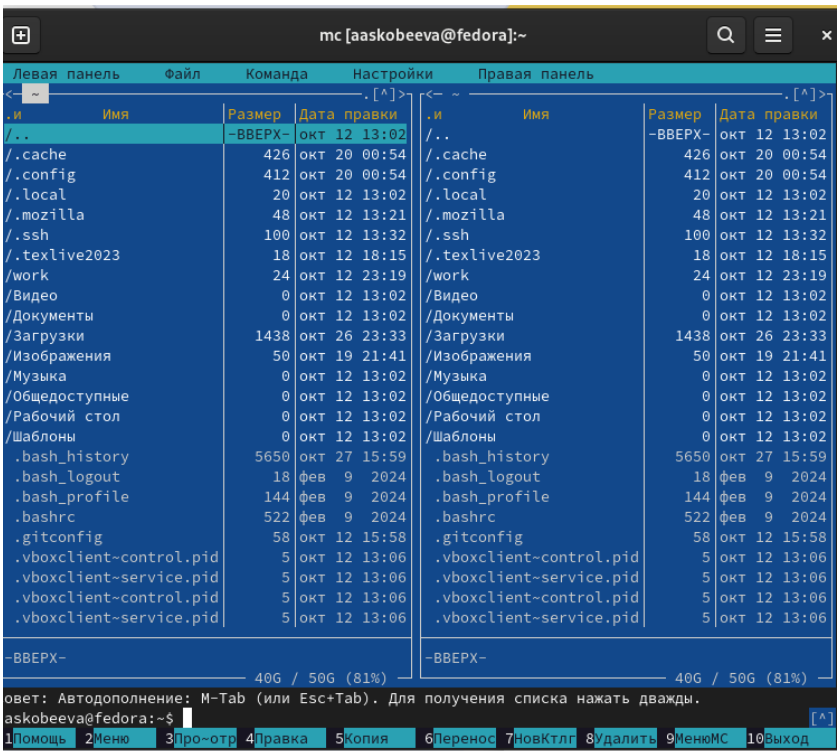


Рис. 2.1: Вводим команду mc

Переходим в каталог, созданный при выполнении 4 лабораторной работы:

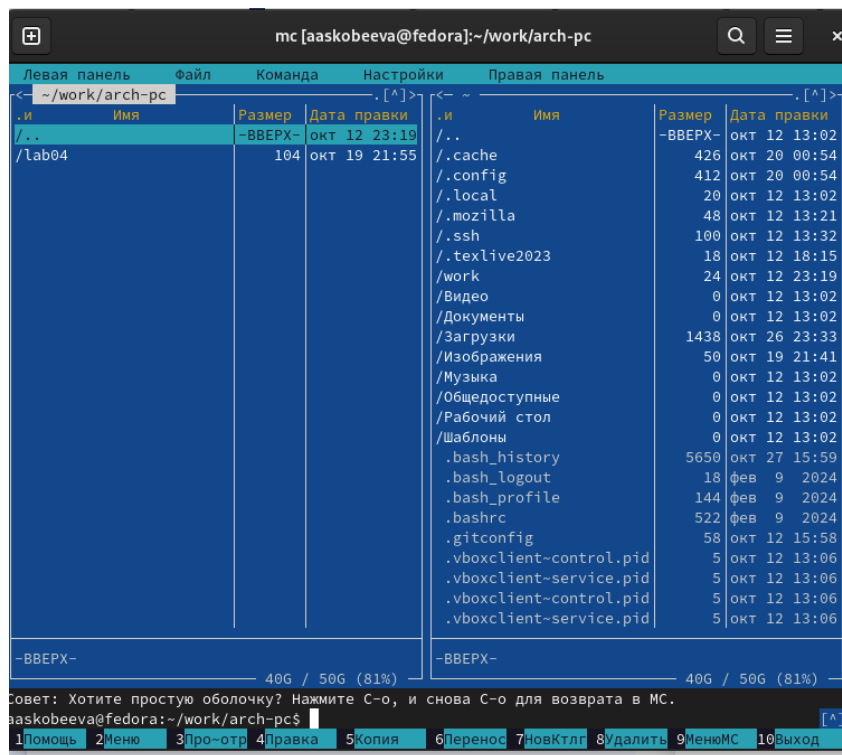


Рис. 2.2: Переходим в каталог

Создаем каталог lab05:

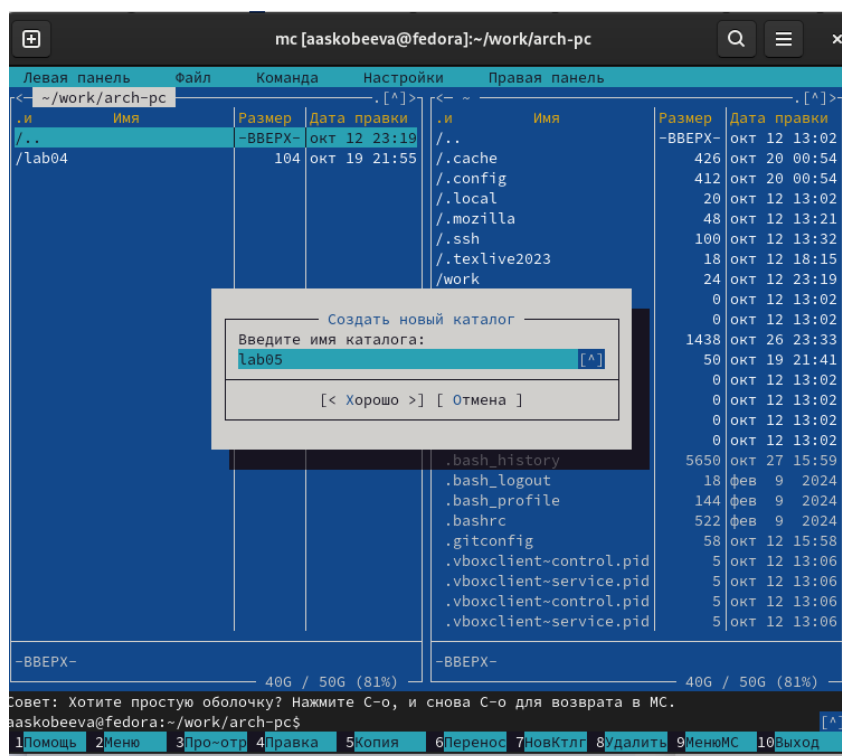
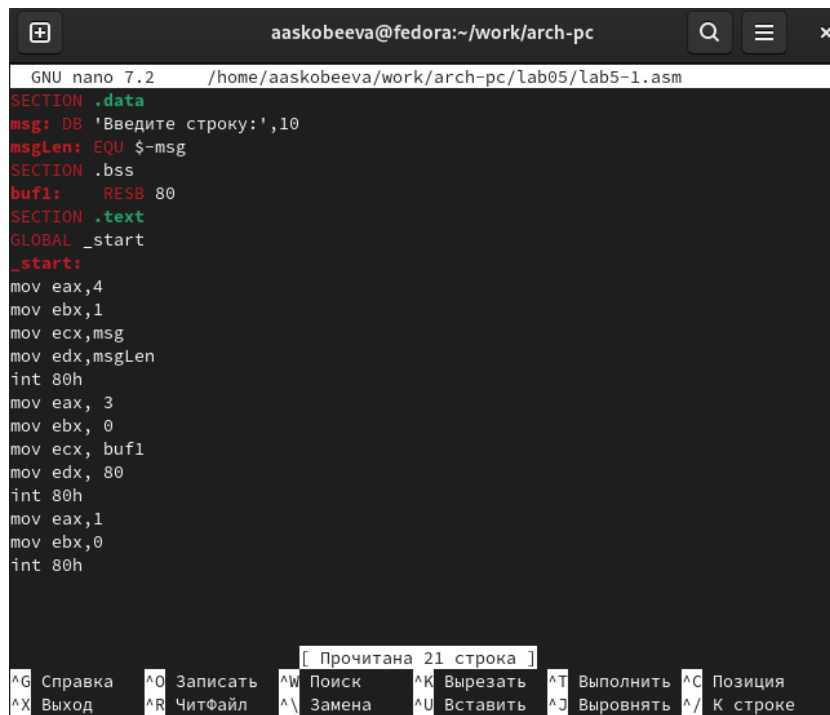


Рис. 2.3: Используем функциональную клавишу F7

Создаем файл lab5-1.asm с помощью команды touch и открываем его:





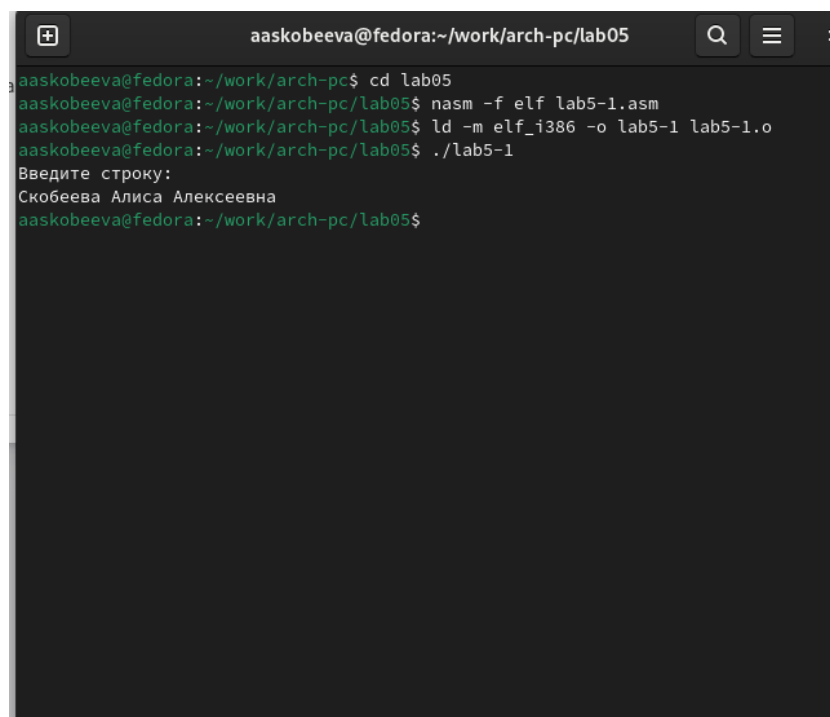
```
GNU nano 7.2 /home/aaskobeeva/work/arch-pc/lab05/lab5-1.asm
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h
mov eax,3
mov ebx,0
mov ecx,buf1
mov edx,80
int 80h
mov eax,1
mov ebx,0
int 80h
```

[ Прочитана 21 строка ]

<b>^G</b> Справка	<b>^O</b> Записать	<b>^W</b> Поиск	<b>^K</b> Вырезать	<b>^T</b> Выполнить	<b>^C</b> Позиция
<b>^X</b> Выход	<b>^R</b> ЧитФайл	<b>^I</b> Замена	<b>^U</b> Вставить	<b>^J</b> Вывод	<b>^_</b> К строке

Рис. 2.4: Заполняем файл по листингу

Транслируем текст программы и запускаем исполняемый файл:



```
aaskobeeva@fedora:~/work/arch-pc/lab05
aaskobeeva@fedora:~/work/arch-pc$ cd lab05
aaskobeeva@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
aaskobeeva@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
aaskobeeva@fedora:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Скобеева Алиса Алексеевна
aaskobeeva@fedora:~/work/arch-pc/lab05$
```

Рис. 2.5: Проверяем, как работает данная программа

Скачиваем файл со страницы курса и перемещаем его в нужный каталог:

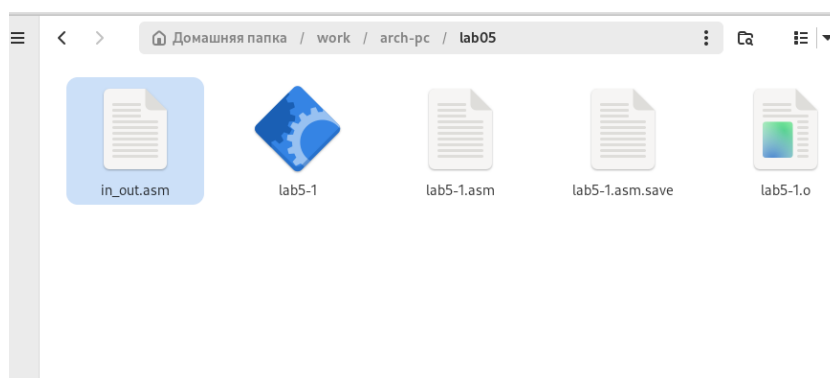


Рис. 2.6: Проверям наличие файла

Создаем копию файла lab5-1.asm:

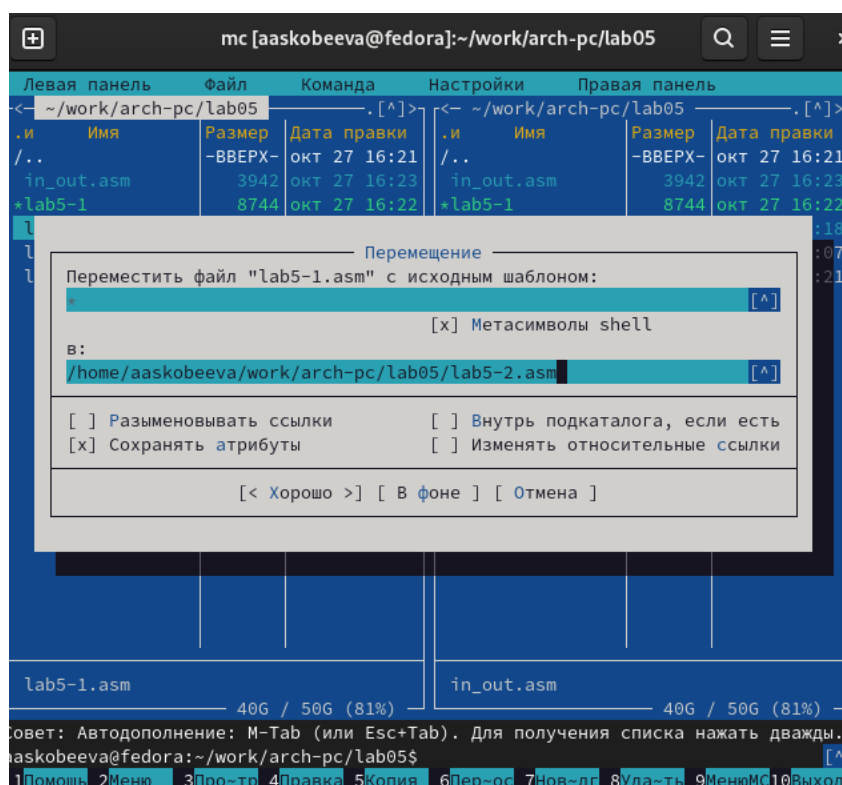


Рис. 2.7: Создаем копию файла клавишей F6

Проверяем созданный файл:

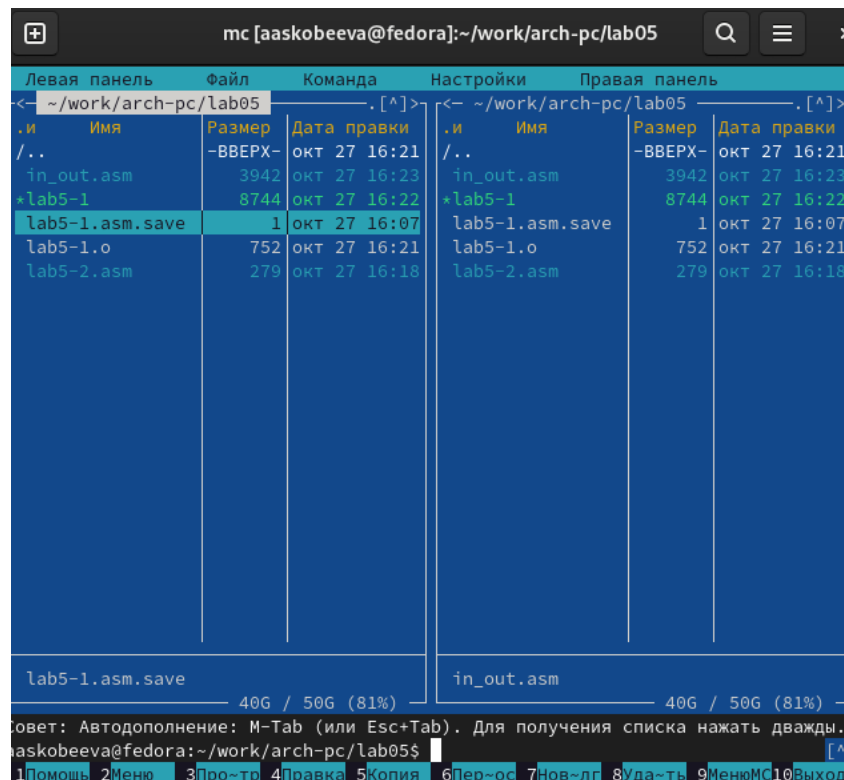
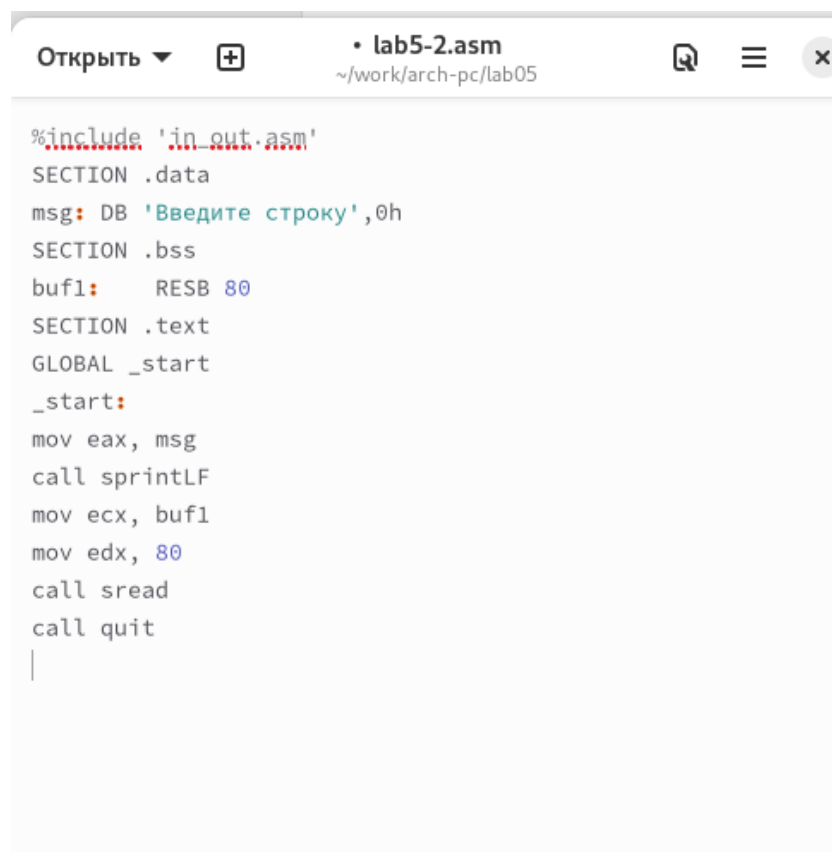


Рис. 2.8: Проверяем скопированный файл

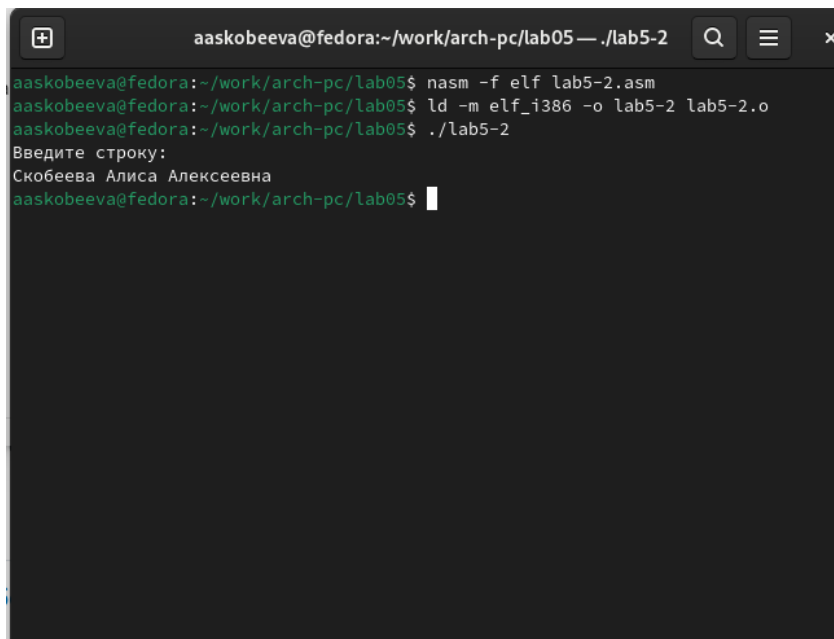
Открываем созданный файл и заполняем его по листингу:

A screenshot of a code editor window. The title bar shows "lab5-2.asm" and the file path "~/work/arch-pc/lab05". The editor contains assembly code for a program that prints a message and reads input. The code is as follows:

```
%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: Открываем и заполняем файл

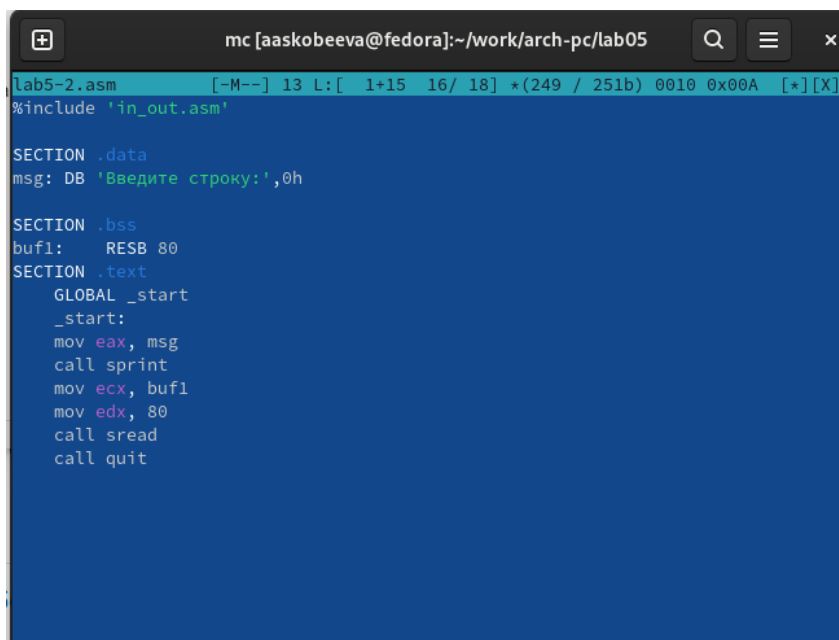
Транслируем и запускаем новый файл:

A terminal window titled 'aaskobeeva@fedora:~/work/arch-pc/lab05 — ./lab5-2'. The user runs 'nasm -f elf lab5-2.asm', then 'ld -m elf\_i386 -o lab5-2 lab5-2.o', and finally './lab5-2'. The output shows 'Введите строку:' followed by 'Скобеева Алиса Алексеевна' and a prompt character.

```
aaskobeeva@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
aaskobeeva@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
aaskobeeva@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Скобеева Алиса Алексеевна
aaskobeeva@fedora:~/work/arch-pc/lab05$
```

Рис. 2.10: Смотрим, как сработала программа

Снова открываем файл для редактирования и меняем `sprintf` на `sprint`:

An editor window titled 'mc [aaskobeeva@fedora]:~/work/arch-pc/lab05'. It shows the assembly file 'lab5-2.asm' with the following code:

```
lab5-2.asm [-M--] 13 L:[ 1+15 16/ 18] *(249 / 251b) 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. 2.11: Редактируем файл

Транслируем и запускаем файл:

```
aaskobeeva@fedora: ~/work/arch-pc/lab05$ mc
aaskobeeva@fedora: ~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
aaskobeeva@fedora: ~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
aaskobeeva@fedora: ~/work/arch-pc/lab05$ ./lab5-2
Введите строку:Скобеева Алиса Алексеевна
aaskobeeva@fedora: ~/work/arch-pc/lab05$
```

Рис. 2.12: Смотрим, как сработала программа и сравниваем ее с прошлой

Таким образом, можем сделать вывод, что команда `sprint` выводит текст в той же строке, а `sprintLF` переносит текст на новую строку.

# 3 Выполнение самостоятельной работы

Создаем копию файла lab5-1:

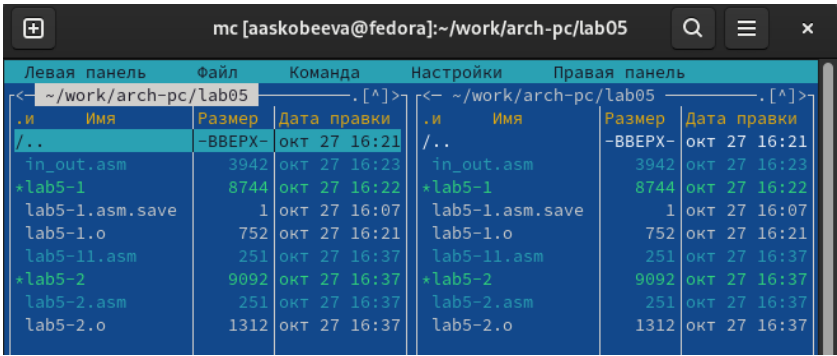
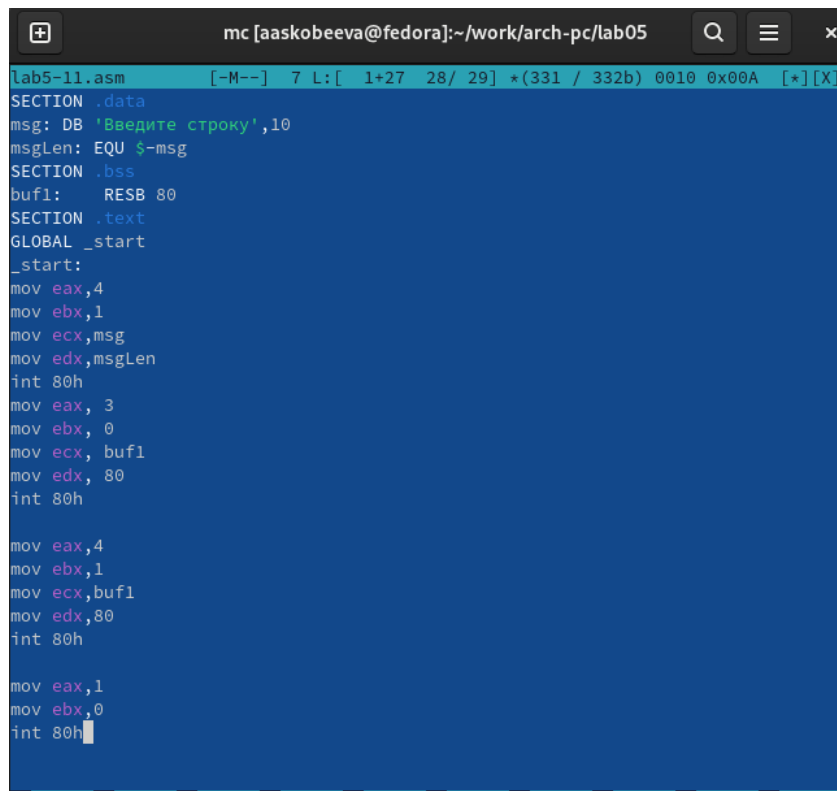


Рис. 3.1: Создание копии файла

Редактируем файл так, чтобы введенный текст с клавиатуры выводился в консоль:



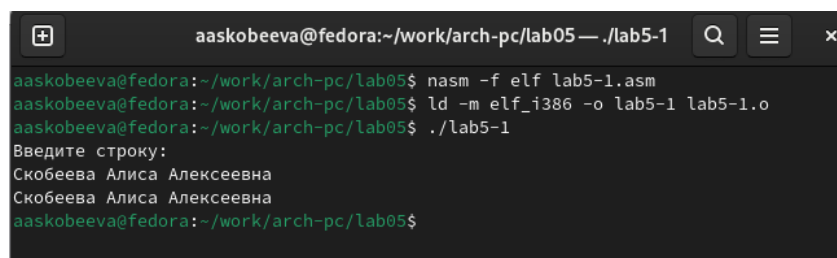
```
lab5-1.asm [-M--] 7 L: [ 1+27 28/ 29] *(331 / 332b) 0010 0x00A [*] [X]
SECTION .data
msg: DB 'Введите строку',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h
mov eax,3
mov ebx,0
mov ecx,buf1
mov edx,80
int 80h

mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,80
int 80h

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

Рис. 3.2: Редактируем файл

Транслируем файл и запускаем программу:

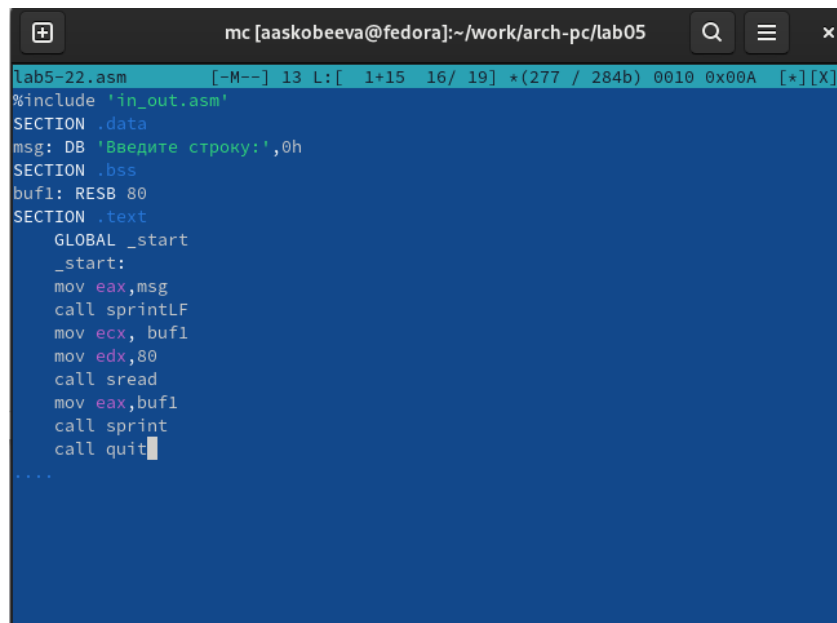


```
aaskobeeva@fedora:~/work/arch-pc/lab05 — ./lab5-1
aaskobeeva@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
aaskobeeva@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
aaskobeeva@fedora:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Скобеева Алиса Алексеевна
Скобеева Алиса Алексеевна
aaskobeeva@fedora:~/work/arch-pc/lab05$
```

Рис. 3.3: Проверяем правильность написания программы

Создаем копию файла lab5-2, открываем его и редактируем так, чтобы введенный текст с клавиатуры выводился в консоль:

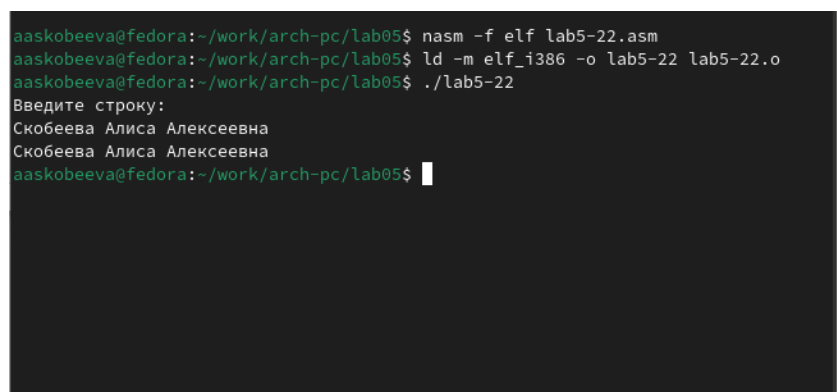




```
lab5-22.asm [-M--] 13 L: [ 1+15 16/ 19] *(277 / 284b) 0010 0x00A [*][X]
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку:',0h
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,msg
call sprintf
mov ecx, buf1
mov edx,80
call sread
mov eax,buf1
call sprintf
call quit
....
```

Рис. 3.4: Редактируем файл

Транслируем файл и запускаем программу:



```
aaskobeeva@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-22.asm
aaskobeeva@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-22 lab5-22.o
aaskobeeva@fedora:~/work/arch-pc/lab05$ ./lab5-22
Введите строку:
Скобеева Алиса Алексеевна
Скобеева Алиса Алексеевна
aaskobeeva@fedora:~/work/arch-pc/lab05$
```

Рис. 3.5: Проверяем правильность написания программы

## 4 Выводы

Я приобрела навыки работы с Midnight Commander и освоила инструкцию mov.