

# **Отчёт по лабораторной работе №5**

**дисциплина: Архитектура компьютера**

Самойлова Софья Дмитриевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
4.1	Основы работы с ms . . . . .	8
4.2	Подключение внешнего файла . . . . .	11
4.3	Выполнение самостоятельной работы . . . . .	14
<b>5</b>	<b>Выводы</b>	<b>17</b>
<b>6</b>	<b>Список литературы</b>	<b>18</b>

# Список иллюстраций

4.1	Midnight Commander . . . . .	8
4.2	Создание lab05 . . . . .	9
4.3	Создание lab05-1.asm . . . . .	9
4.4	Редактирование lab05-1.asm . . . . .	10
4.5	Работа файла . . . . .	10
4.6	Файл на ТУИС . . . . .	11
4.7	Копирование in_out.asm . . . . .	11
4.8	Создание копии . . . . .	12
4.9	Работа нового файла . . . . .	12
4.10	Внесение изменений . . . . .	13
4.11	Работа файла после исправлений . . . . .	13
4.12	Создание копии . . . . .	14
4.13	Работа в копии . . . . .	15
4.14	Работа копии . . . . .	15
4.15	Редактирование копии . . . . .	16
4.16	Работа копии . . . . .	16

# 1 Цель работы

Целью лабораторной работы является приобретение практических навыков работы в Midnight Commander, освоение инструкций языка ассемблера `mov` и `int`.

## 2 Задание

1. Основы работы с mc
2. Подключение внешнего файла in\_out.asm
3. Выполнение заданий для самостоятельной работы

## 3 Теоретическое введение

Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной. Программа на языке ассемблера NASM, как правило, состоит из трёх секций: секция кода программы (SECTION .text), секция инициированных (известных во время компиляции) данных (SECTION .data) и секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение присваивается в ходе выполнения программы) (SECTION .bss). Для объявления инициированных данных в секции .data используются директивы DB, DW, DD, DQ и DT, которые резервируют память и указывают, какие значения должны храниться в этой памяти:

- DB (define byte) — определяет переменную размером в 1 байт;
- DW (define word) — определяет переменную размером в 2 байта (слово);
- DD (define double word) — определяет переменную размером в 4 байта (двойное слово);
- DQ (define quad word) — определяет переменную размером в 8 байт (четверённое слово);
- DT (define ten bytes) — определяет переменную размером в 10 байт.

Директивы используются для объявления простых переменных и для объявления массивов. Для определения строк принято использовать директиву DB

в связи с особенностями хранения данных в оперативной памяти. Инструкция языка ассемблера `mov` предназначена для дублирования данных источника в приёмнике.

```
mov dst,src
```

Здесь операнд `dst` — приёмник, а `src` — источник. В качестве операнда могут выступать регистры (`register`), ячейки памяти (`memory`) и непосредственные значения (`const`). Инструкция языка ассемблера `int` предназначена для вызова прерывания с указанным номером.

```
int n
```

Здесь `n` — номер прерывания, принадлежащий диапазону 0–255. При программировании в Linux с использованием вызовов ядра `sys_calls` `n=80h` (принято задавать в шестнадцатеричной системе счисления).

## 4 Выполнение лабораторной работы

### 4.1 Основы работы с mc

Открываю *Midnight Commander* и перехожу в каталог `~/work/arch-rc` созданный при выполнении лабораторной работы №4 (рис. 4.1).

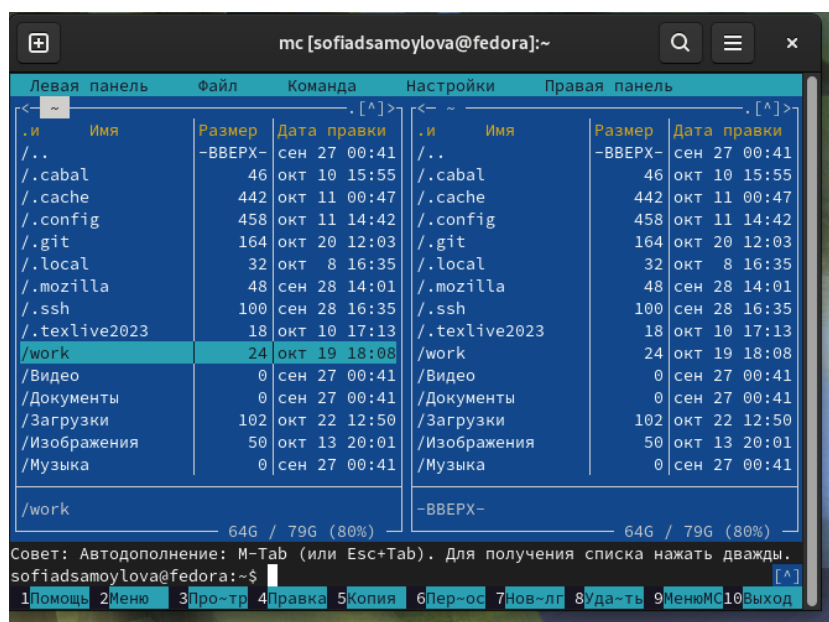


Рис. 4.1: Midnight Commander

С помощью функциональной клавиши F7 создаю папку `lab05` и перехожу в созданный каталог (рис. 4.2).



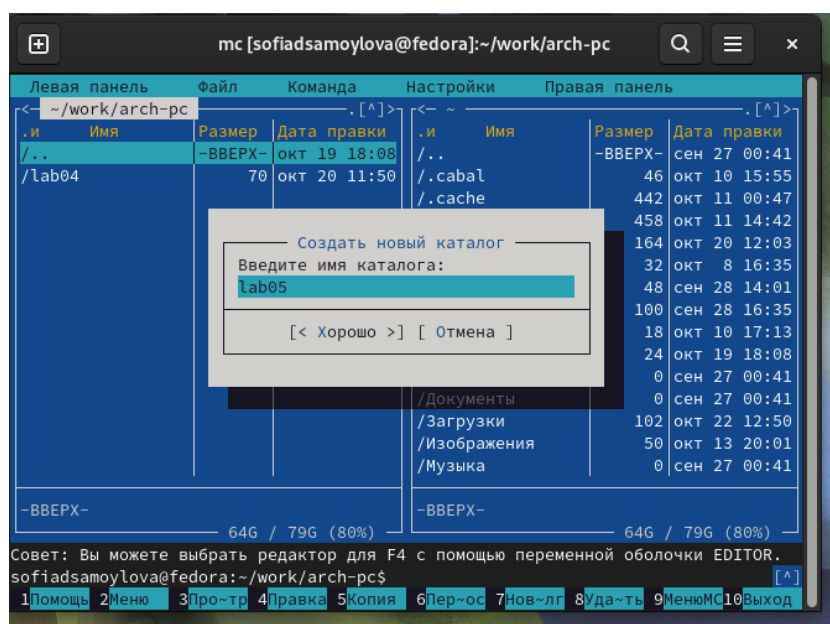


Рис. 4.2: Создание lab05

Пользуясь строкой ввода и командой touch создаю файл lab5-1.asm (рис. 4.3).

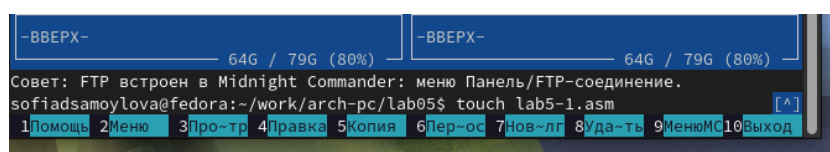


Рис. 4.3: Создание lab05-1.asm

С помощью функциональной клавиши F4 открываю файл lab5-1.asm для редактирования во встроенном редакторе. В качестве встроенного редактора *Midnight Commander* используется редактор *mcedit*. Ввожу текст программы из листинга, сохраняю изменения. (рис. 4.4).

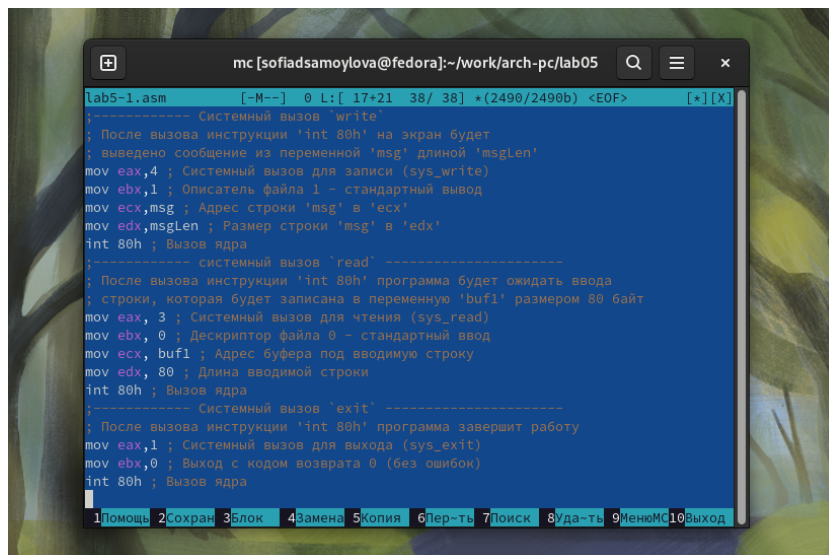


Рис. 4.4: Редактирование lab05-1.asm

Оттранслирую текст программы lab5-1.asm в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос ввожу свои ФИО (рис. 4.5).

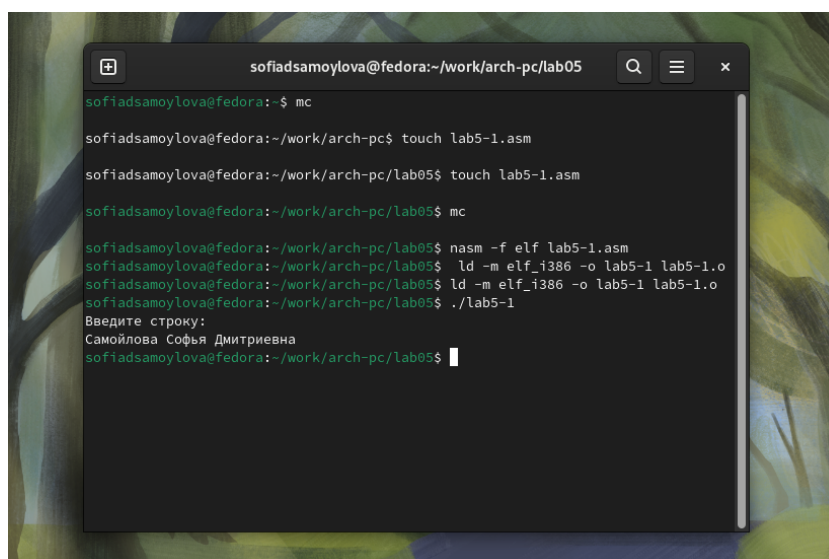


Рис. 4.5: Работа файла

## 4.2 Подключение внешнего файла

Скачиваю файл `in_out.asm` со страницы курса в ТУИС (рис. 4.6).

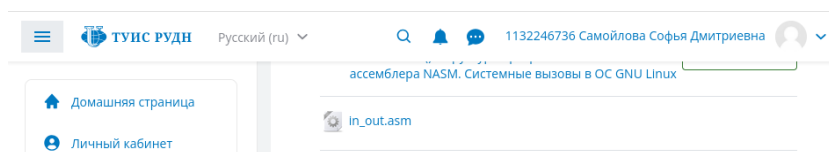


Рис. 4.6: Файл на ТУИС

Копирую файл `in_out.asm` в каталог с файлом `lab5-1.asm` с помощью функциональной клавиши F5 (рис. 4.7).

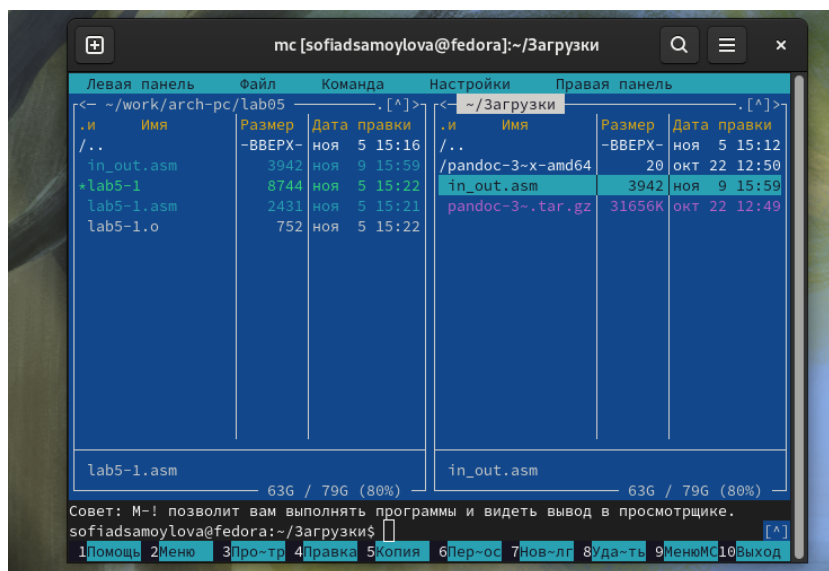


Рис. 4.7: Копирование `in_out.asm`

С помощью функциональной клавиши F6 создаю копию файла `lab5-1.asm` с именем `lab5-2.asm`. Выделяю файл `lab5-1.asm`, нажимаю клавишу F6, ввожу имя файла `lab5-2.asm` и нажимаю клавишу Enter (рис. 4.8).

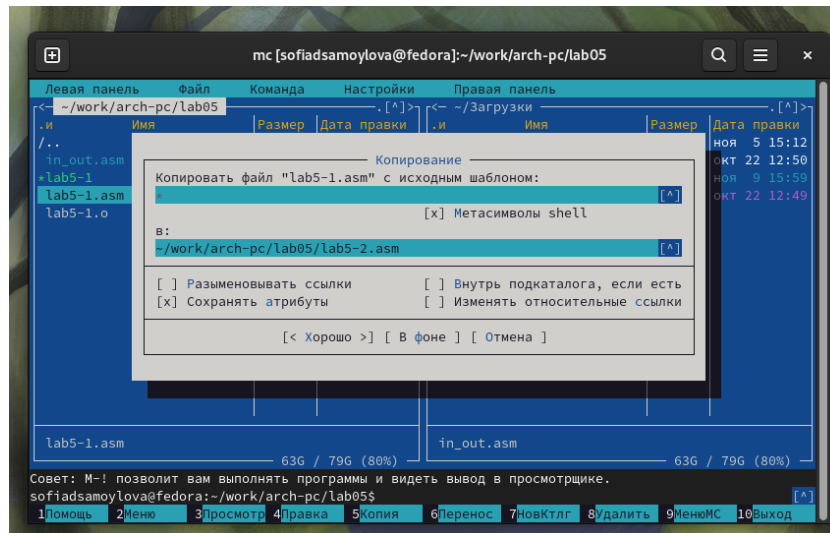


Рис. 4.8: Создание копии

Исправляю текст программы в файле `lab5-2.asm` с использованием подпрограмм из внешнего файла `in_out.asm`. Создаю исполняемый файл и проверяю его работу (рис. 4.9).

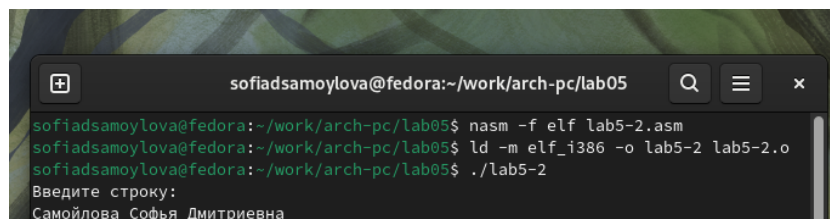
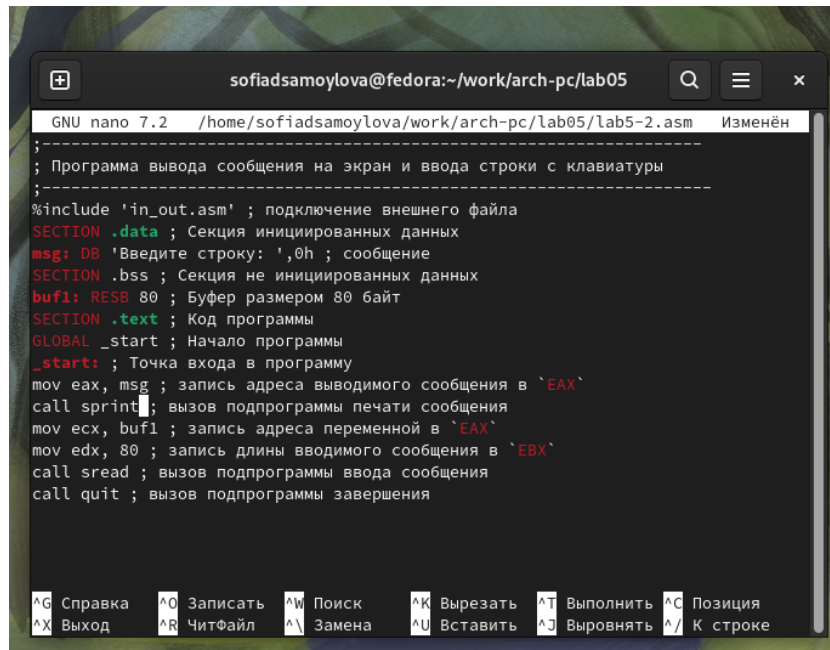


Рис. 4.9: Работа нового файла

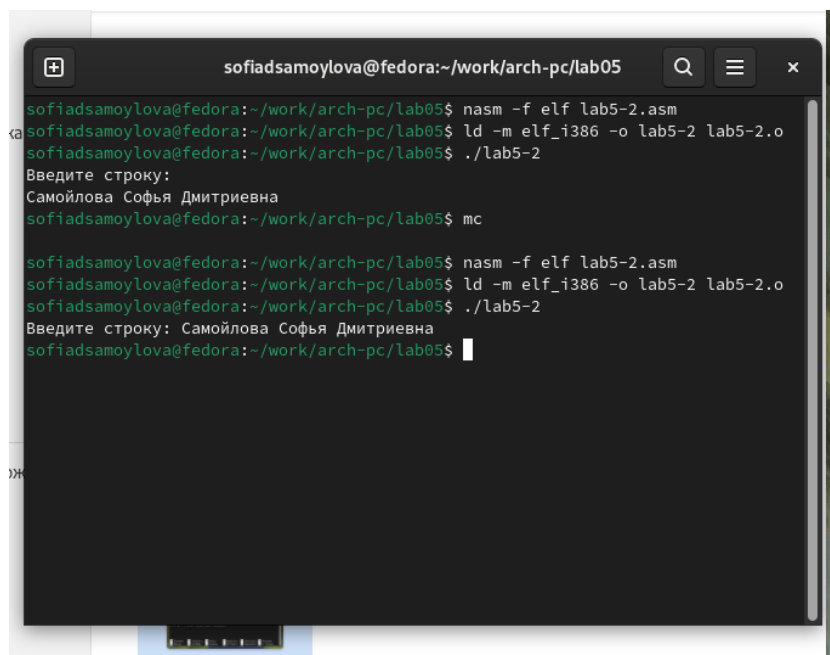
В файле `lab5-2.asm` заменяю подпрограмму `sprintf` на `sprint` (рис. 4.10).



```
GNU nano 7.2 /home/sofiadsamoylova/work/arch-pc/lab05/lab5-2.asm Изменён
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не иницированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
mov eax, msg ; запись адреса выводимого сообщения в `EAX`
call sprint ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 4.10: Внесение изменений

Создаю исполняемый файл и проверяю его работу (рис. 4.11).



```
sofiadsamoylova@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
sofiadsamoylova@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
sofiadsamoylova@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Самойлова Софья Дмитриевна
sofiadsamoylova@fedora:~/work/arch-pc/lab05$ mc

sofiadsamoylova@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
sofiadsamoylova@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
sofiadsamoylova@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Самойлова Софья Дмитриевна
sofiadsamoylova@fedora:~/work/arch-pc/lab05$
```

Рис. 4.11: Работа файла после исправлений

Разница между исполняемыми файлами в том, что запуск первого запрашивает

ввод с новой строки, а программа, которая выполняется при запуске второго, запрашивает ввод без переноса на новую строку, в этом и заключается само различие между подпрограммами `sprintLF` и `sprint`.

## 4.3 Выполнение самостоятельной работы

Создаю копию файла `lab5-1.asm`(рис. 4.12).

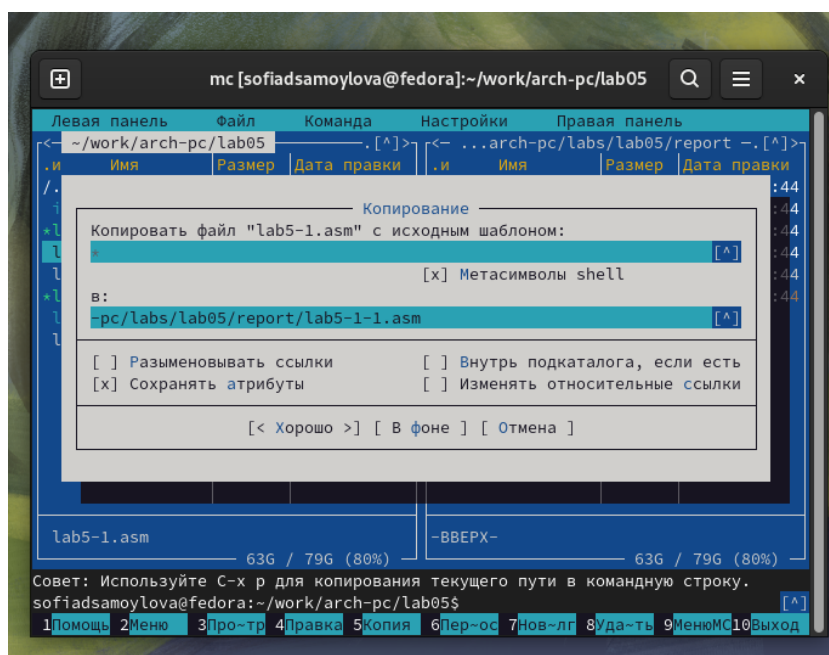
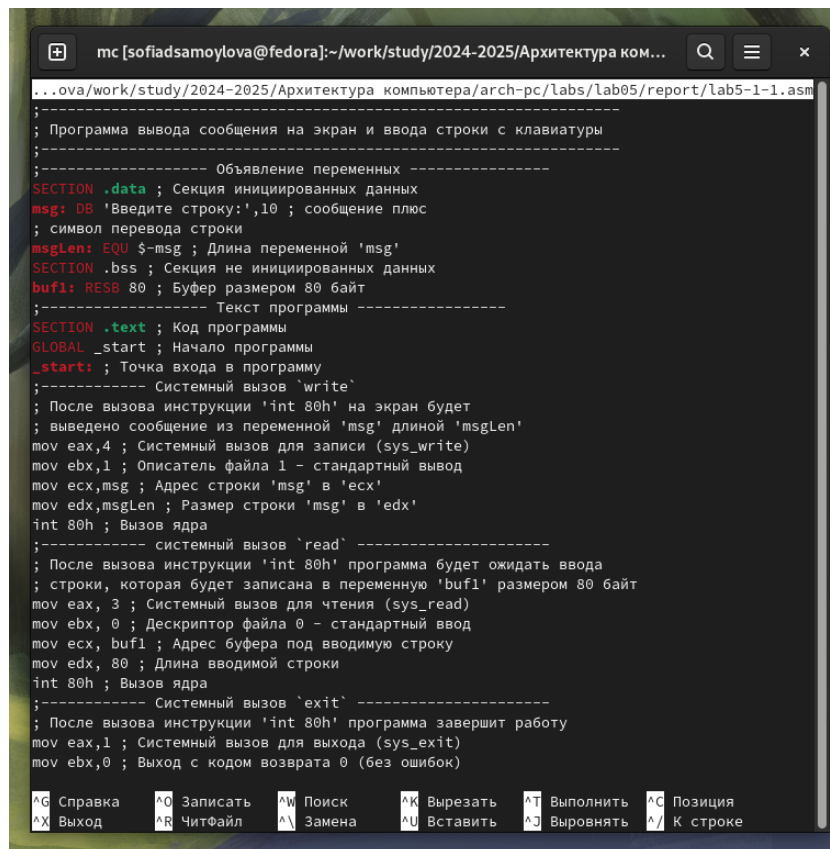


Рис. 4.12: Создание копии

Вношу изменения в программу, так чтобы она работала по следующему алгоритму (рис. 4.13):

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

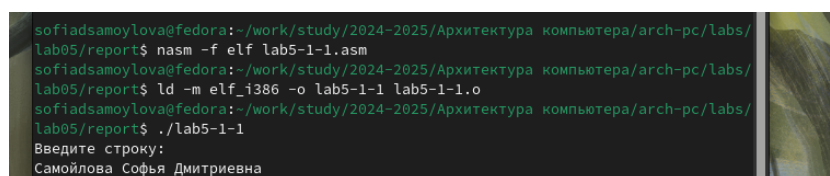


```
mc [sofiadsamoylova@fedora]:~/work/study/2024-2025/Архитектура ком...
...ova/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05/report/lab5-1-1.asm
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
;----- Текст программы -----
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
;----- Системный вызов 'write' -----
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'msg' длиной 'msgLen'
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра
;----- системный вызов 'read' -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт
mov eax,3 ; Системный вызов для чтения (sys_read)
mov ebx,0 ; Дескриптор файла 0 - стандартный ввод
mov ecx,buf1 ; Адрес буфера под вводимую строку
mov edx,80 ; Длина вводимой строки
int 80h ; Вызов ядра
;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)

^C Справка      ^O Записать    ^W Поиск       ^K Вырезать    ^T Выполнить   ^S Позиция
^X Выход        ^R ЧитФайл    ^\ Замена      ^U Вставить    ^J Выводить    ^/_ К строке
```

Рис. 4.13: Работа в копии

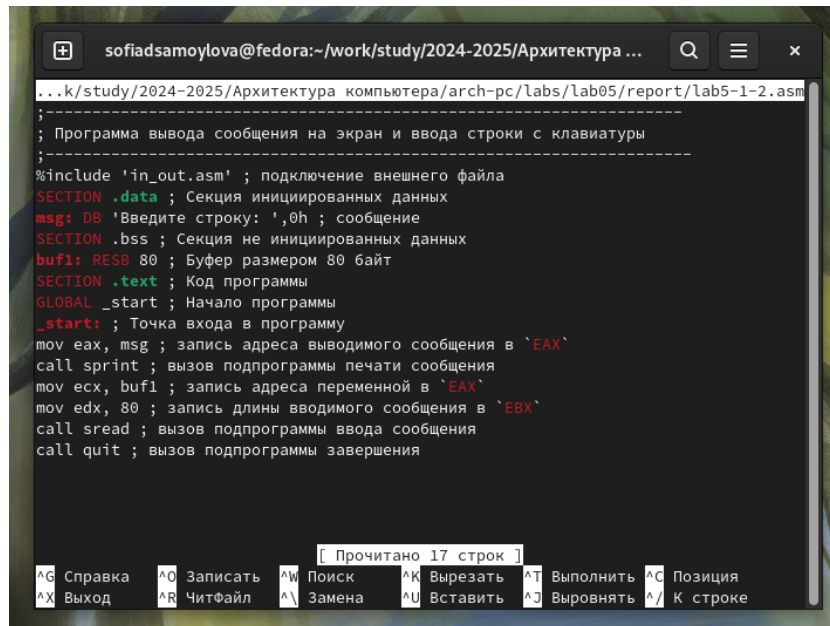
Получаю исполняемый файл и проверяю его работу. На приглашение ввести строку ввожу свои ФИО (рис. 4.14).



```
sofiadsamoylova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/
lab05/report$ nasm -f elf lab5-1-1.asm
sofiadsamoylova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/
lab05/report$ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
sofiadsamoylova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/
lab05/report$ ./lab5-1-1
Введите строку:
Самойлова Софья Дмитриевна
```

Рис. 4.14: Работа копии

Создаю копию файла lab5-2.asm. Исправляю текст программы с использованием подпрограмм из внешнего файла in\_out.asm (рис. 4.15), так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введенную строку на экран.



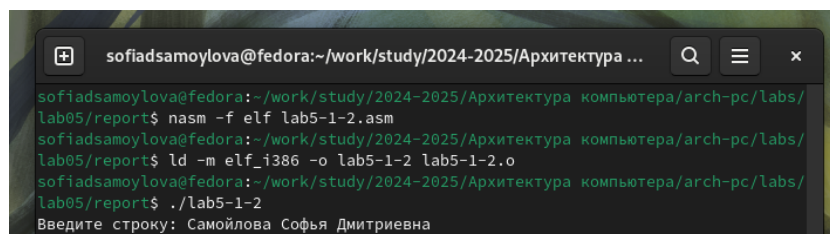
```
.../study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05/report/lab5-1-2.asm
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
mov eax, msg ; запись адреса выводимого сообщения в `EAX`
call sprint ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

[ Прочитано 17 строк ]

⌘G Справка ⌘O Записать ⌘W Поиск ⌘K Вырезать ⌘T Выполнить ⌘C Позиция  
⌘X Выход ⌘R ЧитФайл ⌘\ Замена ⌘U Вставить ⌘J Выводить ⌘/ К строке

Рис. 4.15: Редактирование копии

Создаю исполняемый файл и проверяю его работу (рис. 4.16).



```
sofiadsamoylova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/
lab05/report$ nasm -f elf lab5-1-2.asm
sofiadsamoylova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/
lab05/report$ ld -m elf_i386 -o lab5-1-2 lab5-1-2.o
sofiadsamoylova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/
lab05/report$ ./lab5-1-2
Введите строку: Самойлова Софья Дмитриевна
```

Рис. 4.16: Работа копии



## 5 Выводы

При выполнении данной лабораторной работы я приобрела практические навыки работы в Midnight Commander, а также освоила инструкции языка ассемблера `mov` и `int`.

## **6 Список литературы**

1. Лабораторная работа №5