Отчет по лабораторной работе №5

Дисциплина: архитектура компьютера

Ермакова Анастасия Алексеевна

Содержание

Список иллюстраций

Список таблиц

# 1 Цель работы

Цель данной лабораторной работы - приобрести практические навыки работы в Midnight Commander, освоить инструкции языка ассемблера mov и int.

# 2 Задание

1. Основы работы с mc
2. Структура программы на языке ассемблера NASM
3. Подключение внешнего файла
4. 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. Для активации оболочки Midnight Commander достаточно ввести в командной строке mc и нажать клавишу Enter. В Midnight Commander используются функциональные клавиши F1 — F10 , к которым привязаны часто выполняемые операции.

# 4 Выполнение лабораторной работы

## 4.1 Основы работы с mc

Открываю Midnight Commander. Для этого прописываю в терминале mc. (рис. 1).

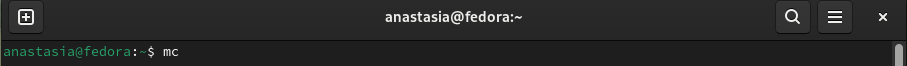


Рис. 1: Команда mc

Пользуясь курсорными клавишами и Enter перехожу в каталог ~/work/arcg-pc. (рис. 2).

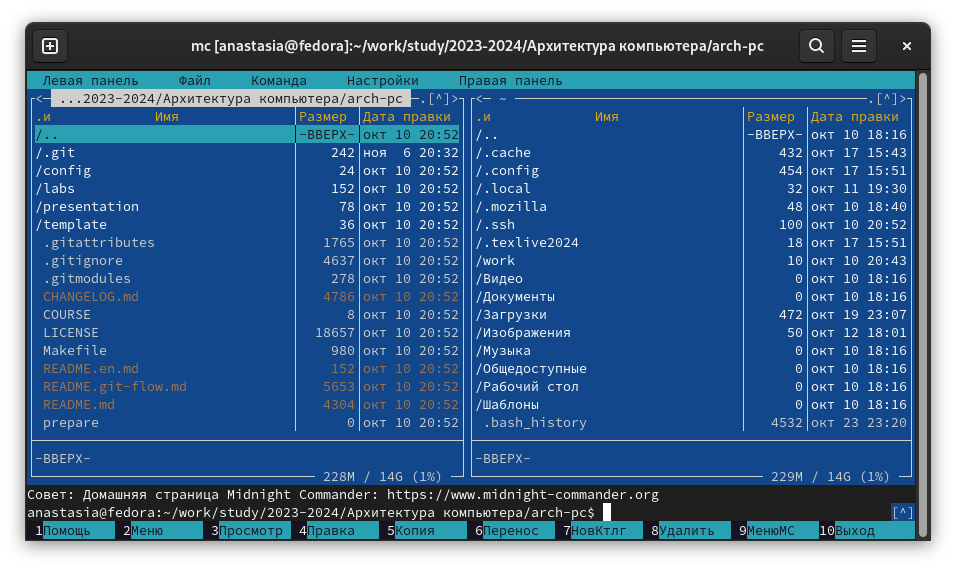


Рис. 2: Интерфейс Midnight Commander

С помощью функциональной клавиши F7 создаю папку lab05 и перехожу в созданный каталог. (рис. 3).

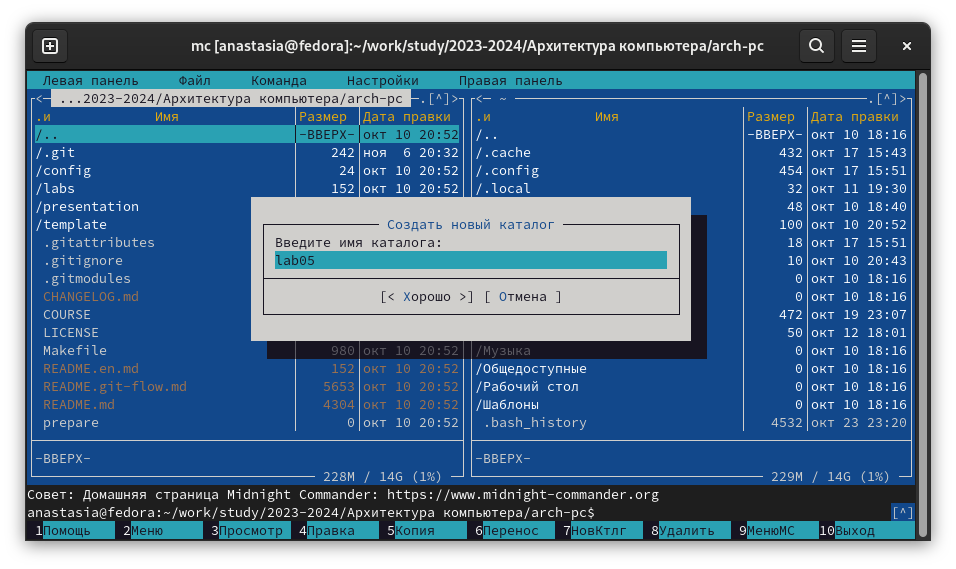


Рис. 3: Создание папки в каталоге

Пользуясь строкой ввода и командой touch создаю файл lab5-1.asm. (рис. 4).

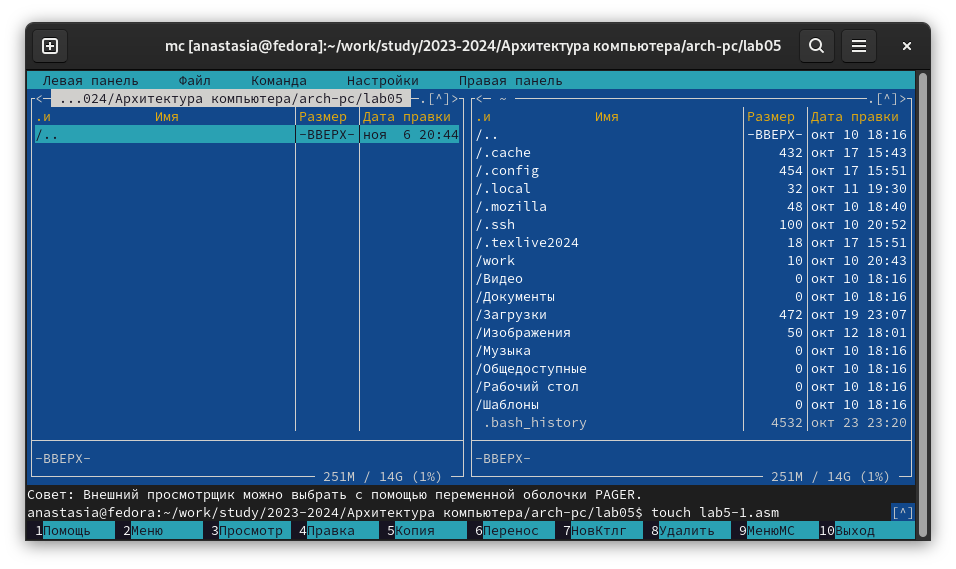


Рис. 4: Создание файла с помощью команды touch

## 4.2 Структура программы на языке ассемблера NASM

С помощью функциональной клавиши F4 открываю файл lab5-1.asm для редактирования во встроенном редакторе mcedit. Ввожу текст программы из листинга 5.1, сохраняю изменения и закрывю файл. (рис. 5).

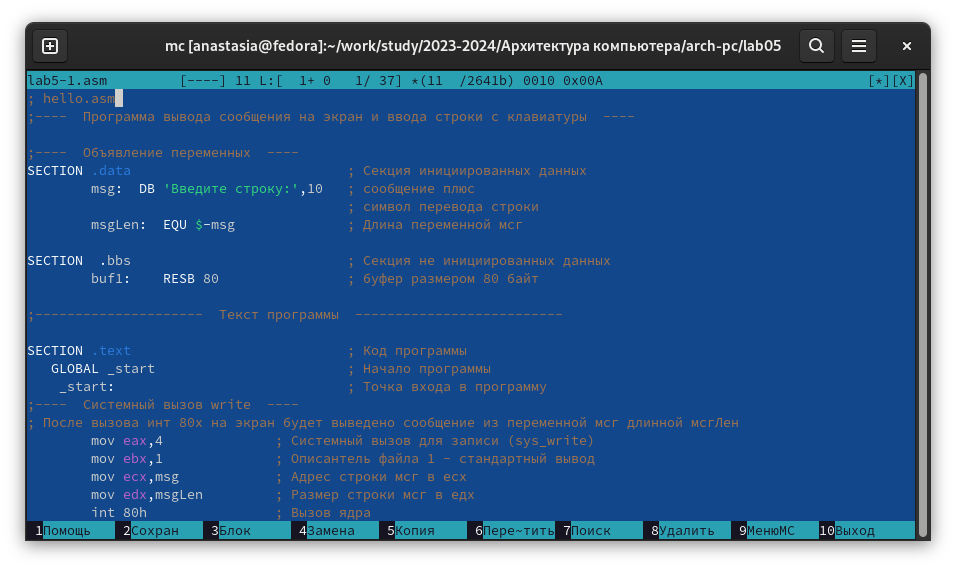


Рис. 5: Окно Midnught Commander. Редактор mcedit

С помощью функциональной клавиши F3 открываю файл lab5-1.asm для просмотра и убеждаюсь, что файл содержит текст программы. (рис. 6).

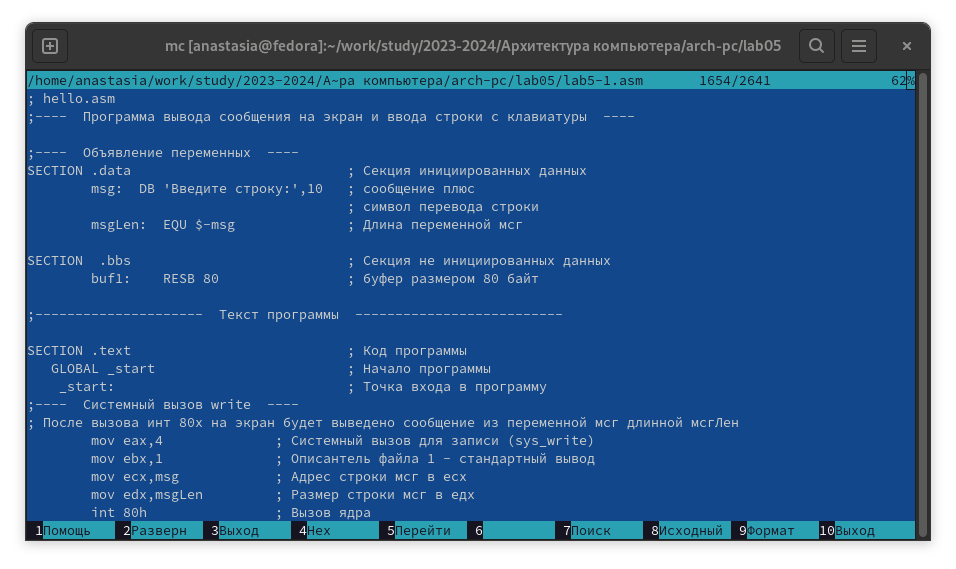


Рис. 6: Файл lab5-1.asm

Оттранслирываю текст программы lab5-1.asm в объектный файл, для этого прописываю nasm -f elf lab5-1.asm (рис. 7). Затем выполняю компановку объектного файла (ld -m elf\_i386 -o lab5-1 lab5-1.o) (рис. 8), создался исполняемый файл \*lab5-1. Запускаю получившийся файл. (рис. 9).

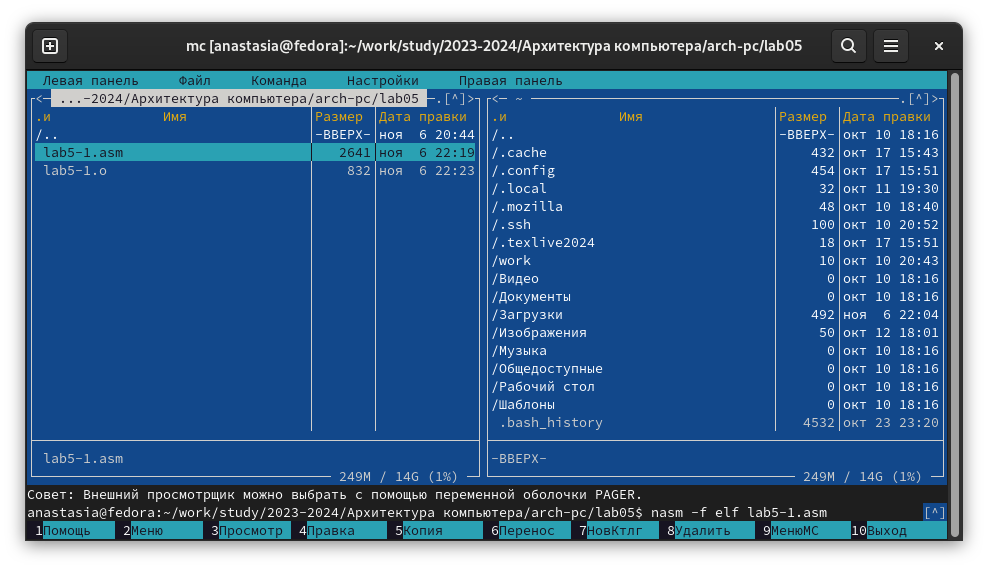


Рис. 7: Создание объектного файла

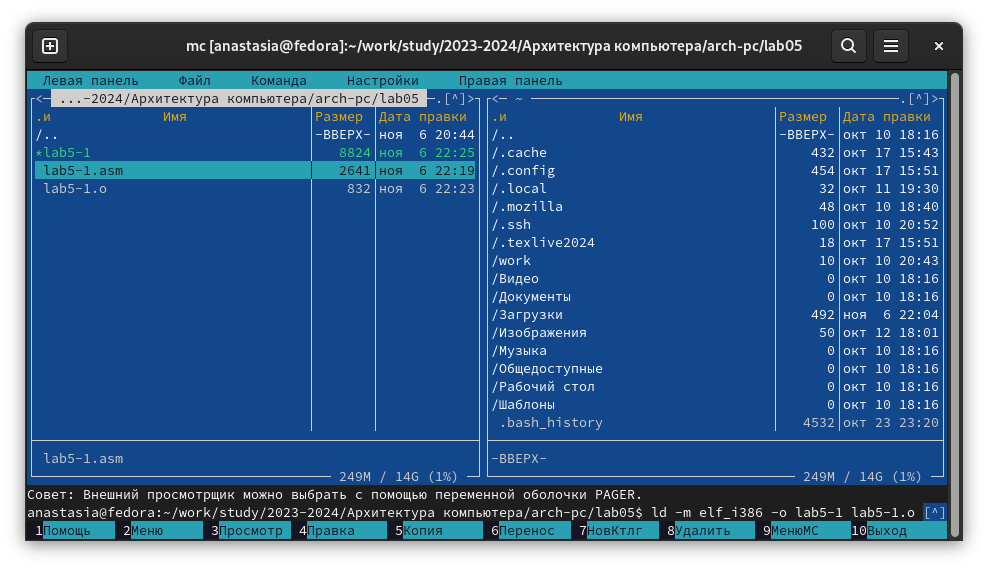


Рис. 8: Компановка объектного файла в исполняемый

Рис. 9: Запуск исполняемого файла

Рис. 9: Запуск исполняемого файла

Программа вывела строку ‘Введите строку:’, ввожу свои ФИО. На этом работа программы заканчивается. (рис. 10).

Рис. 10: Работа программы исполняемого файла

Рис. 10: Работа программы исполняемого файла

## 4.3 Подключение внешнего файла

Скачиваю файл in\_out.asm со страницы курса в ТУИС. Он автоматически сохраняется в каталог Загрузки. (рис. 11).

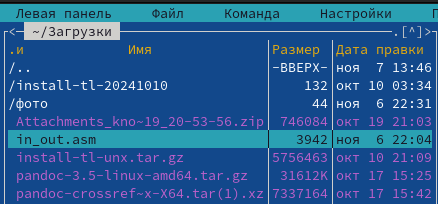


Рис. 11: Файл in\_out.asm в каталоге Загрузки

Копирую его с помощью функциональной клавиши F5 в каталог ~/work/arch-pc/lab05 в котором работаю, и в котором лежит файл lab5-1.asm. (рис. 12 - 13).

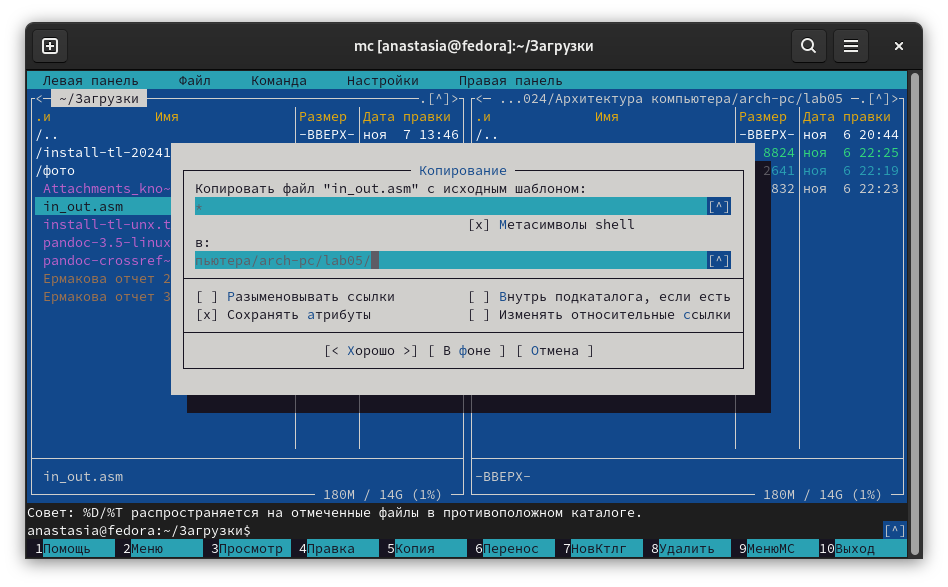


Рис. 12: Копирование файла в нужный каталог

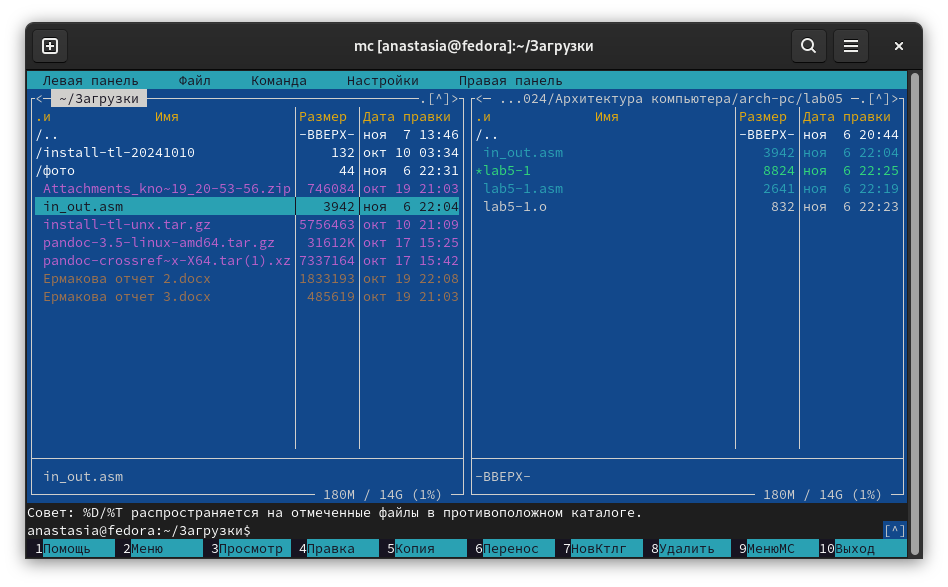


Рис. 13: Файл in\_out.asm в каталоге lab05

С помощью функциональной клавиши F6 создаю копию файла lab5-1.asm с именем lab5-2.asm. Исправляю текст программы в последнем файле с использованием программ из внешнего файла in\_out.asm в соответствии с листингом 5.2. (рис. 14).

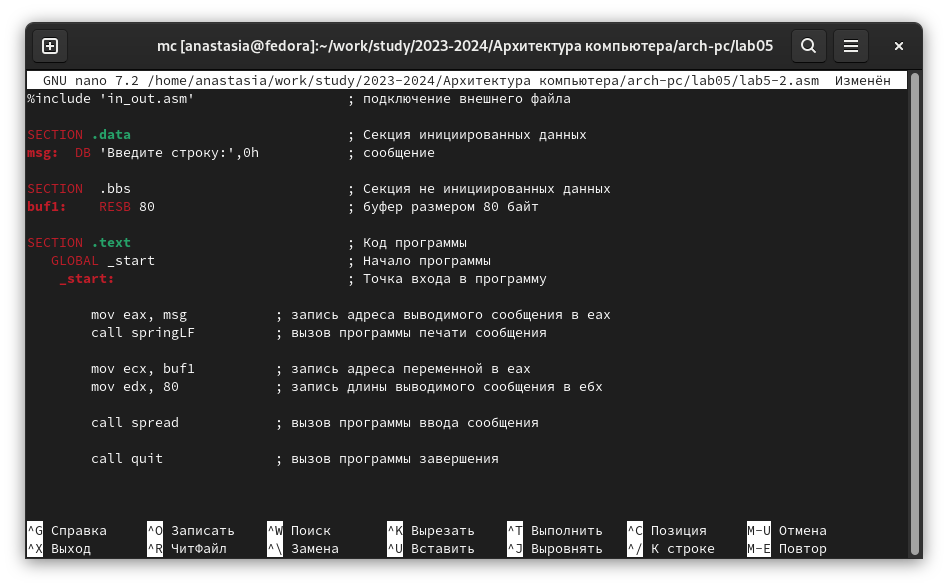


Рис. 14: Изменение текста файла lab5-2.asm

Создаю исполняемый файл и проверяю его работу. (рис. 15).

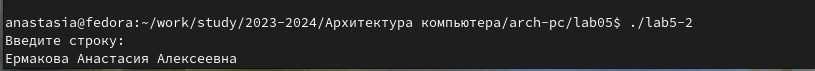


Рис. 15: Работа программы исполняемого файла

Заменяю в файле lab5-2.asm подпрограмму sprintLF на sprint. Создаю исполняемый файл и проверяю его работу. (рис. 16).

Рис. 16: Работа измененной программы исполняемого файла

Рис. 16: Работа измененной программы исполняемого файла

Разница между ними в том, что первая программа запрашивает ввод с новой строки (sptintLF), а вторая запрашивает ввод без перехода на новую строку(sprint).

## 4.4 Выполнение заданий для самостоятельной работы

Создаю копию файла lab5-1.asm с именем lab5-1-1.asm. (рис. 17).

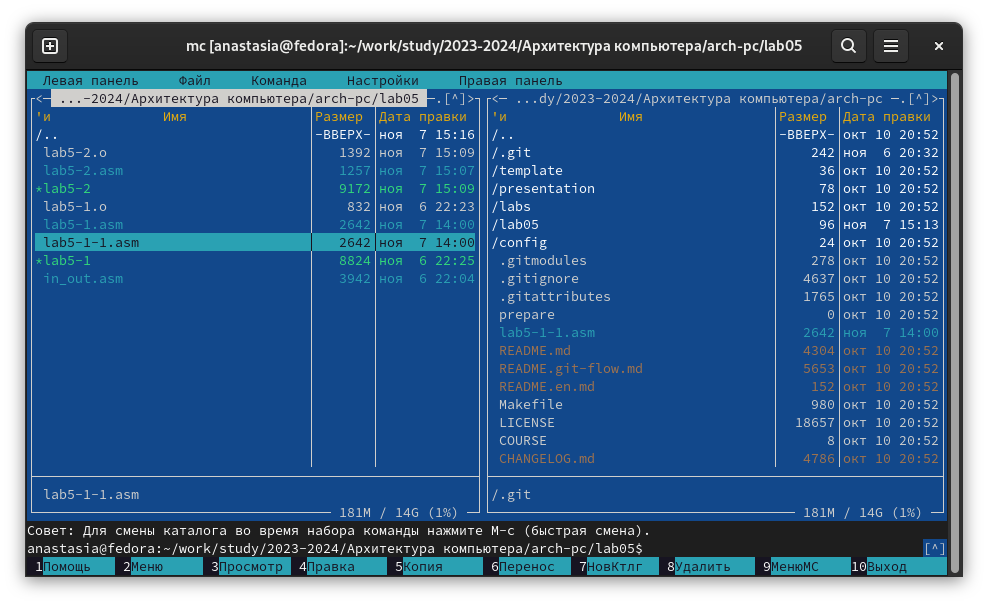


Рис. 17: Копия файла lab5-1.asm

С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу (без использования внешнего файла) так, чтобы после того, как я введу строку с клавиатуры, она выводила ее на экран. (рис. 18).

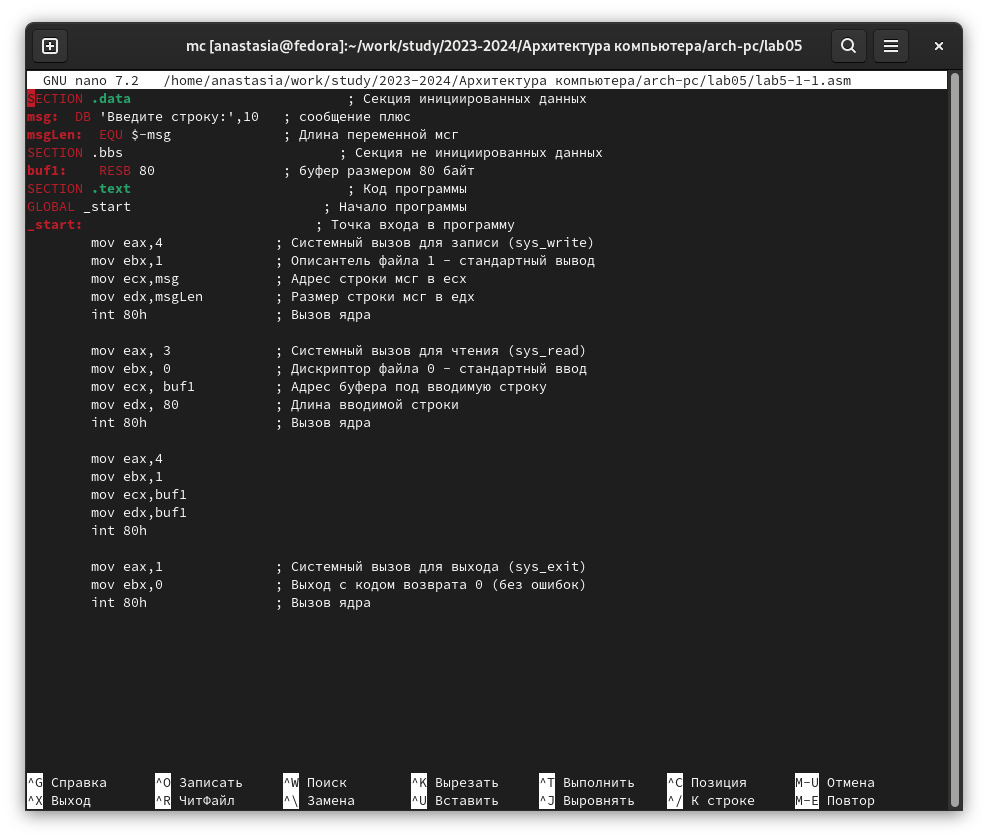


Рис. 18: Текст программы файла lab5-1-1.asm

Получаю исполняемый файл и проверяю его работу: на приглашение ввести строку ввожу свои ФИО. (рис. 19).

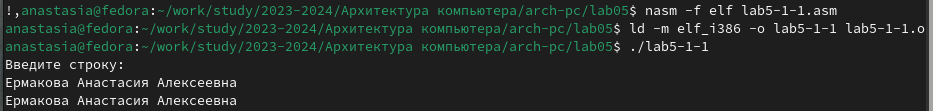


Рис. 19: Работа программы исполняемого файла lab5-1-1

Код программы:

SECTION .data ; Секция инициированных данных  
msg: DB 'Введите строку:',10 ; сообщение плюс  
msgLen: EQU $-msg ; Длина переменной мсг  
SECTION .bss ; Секция не инициированных данных  
buf1: RESB 80 ; буфер размером 80 байт  
SECTION .text ; Код программы  
GLOBAL \_start ; Начало программы  
\_start: ; Точка входа в программу  
  
mov eax,4 ; Системный вызов для записи (sys\_write)  
mov ebx,1 ; Описантель файла 1 - стандартный вывод  
mov ecx,msg ; Адрес строки мсг в есх  
mov edx,msgLen ; Размер строки мсг в едх  
int 80h ; Вызов ядра  
  
mov eax, 3 ; Системный вызов для чтения (sys\_read)  
mov ebx, 0 ; Дискриптор файла 0 - стандартный ввод  
mov ecx, buf1 ; Адрес буфера под вводимую строку  
mov edx, 80 ; Длина вводимой строки  
int 80h ; Вызов ядра  
  
mov eax,4 ; Сиситемный вызов для записи  
mov ebx,1 ; Описатель файла 1 - стандартный вывод  
mov ecx,buf1 ; Адрес строки буф1 в ecx  
mov edx,buf1 ; Размер строки буф1  
int 80h ; Вызов ядра  
  
mov eax,1 ; Системный вызов для выхода (sys\_exit)  
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)  
int 80h ; Вызов ядра

Создаю копию файла lab5-2.asm с именем lab5-2-1.asm. Изменяю эту программу (с использованием подпрограмм внешнего файла in\_out.asm) так, чтобы она работала таким же образом, как и lab5-1-1. Создаю исполняемый файл и проверяю его работу. (рис. 20).

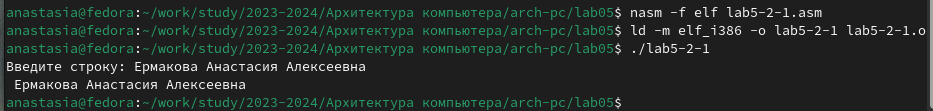


Рис. 20: Работа программы исполняемого файла lab5-2-1

Код программы:

%include 'in\_out.asm' ; подключение внешнего файла  
  
SECTION .data ; Секция инициированных данных  
msg: DB 'Введите строку:',0h ; сообщение  
SECTION .bss ; Секция не инициированных 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 еах  
call sprint ; вызов программы печати сообщения  
  
mov ecx, buf1 ; запись адреса переменной в еах  
mov edx, 80 ; запись длины выводимого сообщения в ебх  
  
call sread ; вызов программы ввода сообщения  
  
mov eax,4 ; системный вызов для записи  
mov ebx,1 ; описатель файла 1 - стандартный вывод  
mov ecx,buf1 ; адрес строки буф1 в есх  
int 80h ; вызов ядра  
  
call quit ; вызов подпрограммы завершения

# 5 Выводы

В ходе выполнения данной лабораторной работы я приобрела практические навыки работы в Midnight Commander и освоила иснтрукции языка ассемблера mov и int.

# 6 Список литературы

1. [Архитектура ЭВМ](https://esystem.rudn.ru/mod/resource/view.php?id=1030553)