Отчет по лабораторной работе №6

Смирнов Дмитрий Романович, НММбд-03-22

Содержание

# 1 Цель работы

Приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера mov и int.

# 2 Выполнение лабораторной работы

Открою Midnight Commander с помощью команды mc, перейду в каталог /work/study/2022-2023/Архитектура компьютера/study\_2022-2023\_arh-pc/labs/lab06 и создам папку lab06, а в ней файл lab6.asm

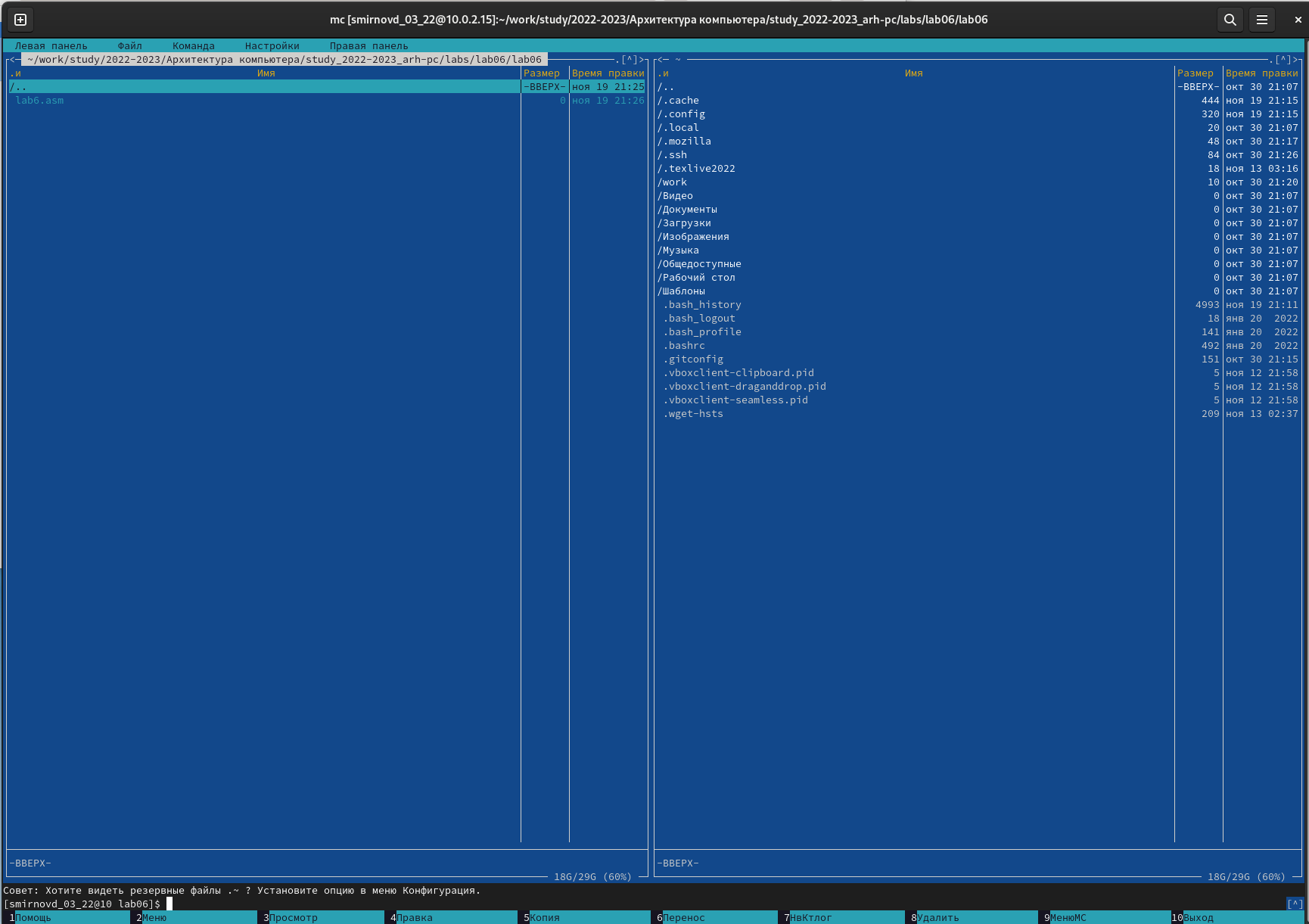


Рис. 1: Рис1

С помощью функциональной клавиши F4 открою файл lab6.asm для редактирования во встроенном редакторе.

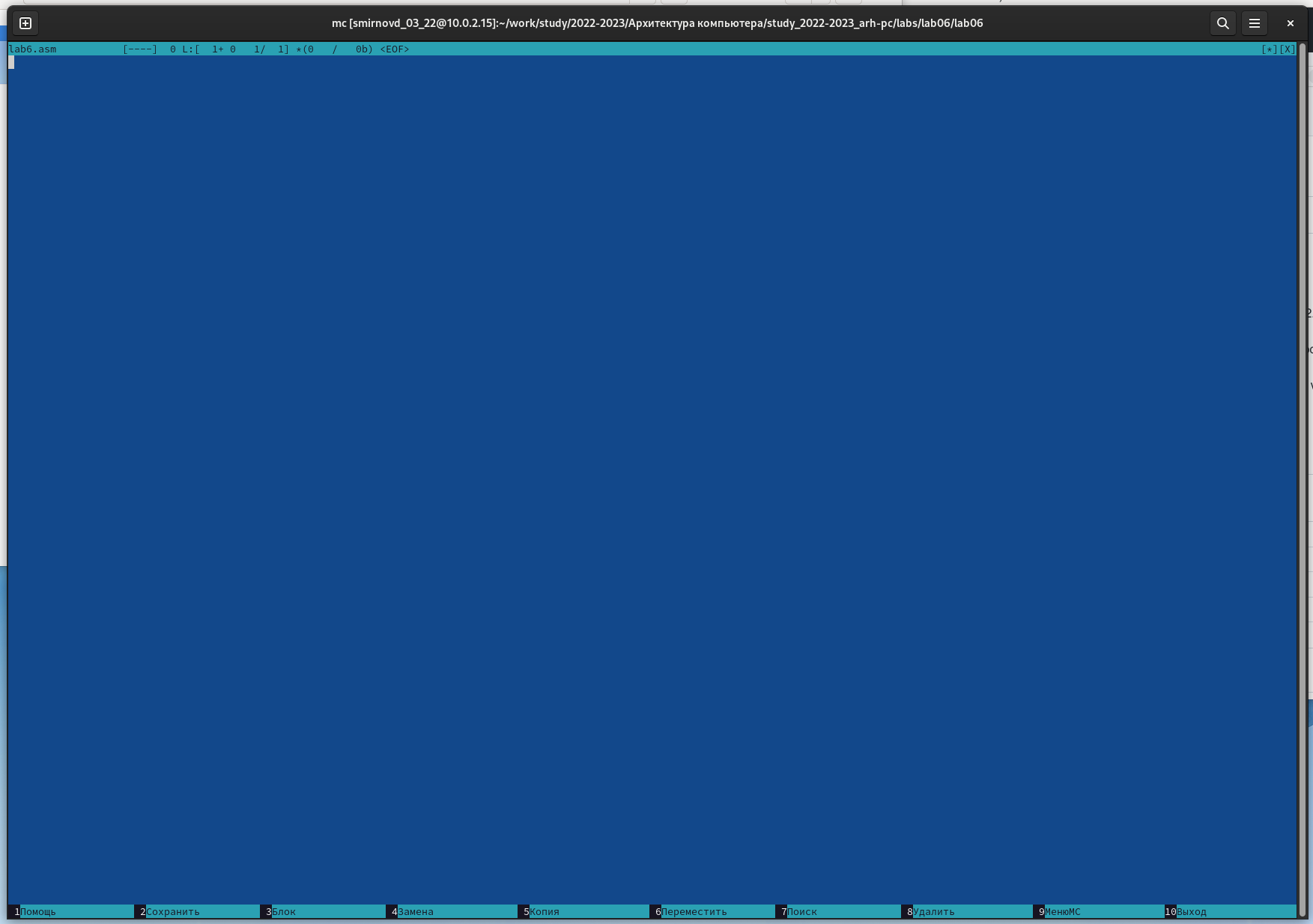


Рис. 2: Рис2

Введу текст программы из листинга 6.1, сохраню изменения и закрою файл.

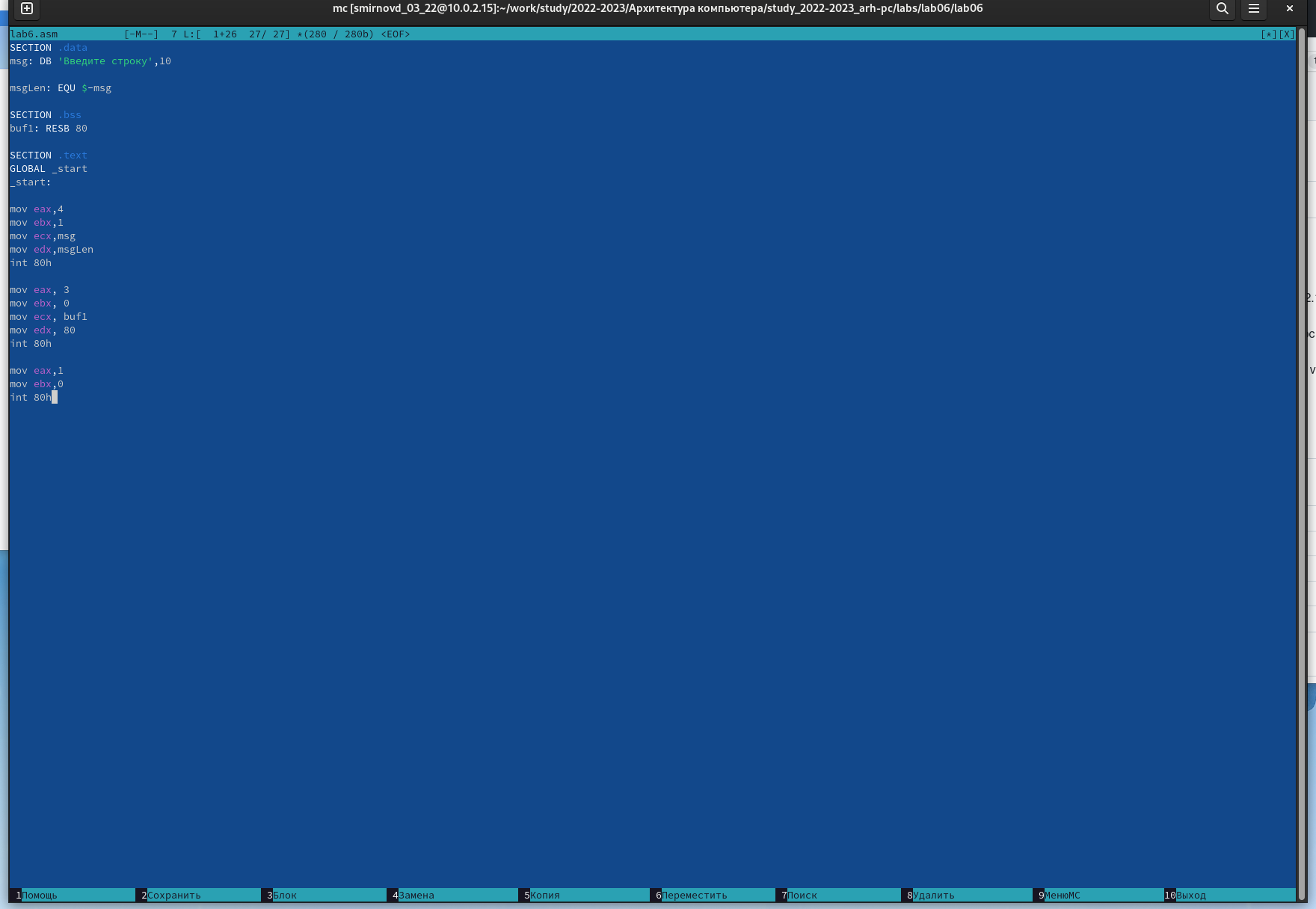


Рис. 3: Рис3

С помощью функциональной клавиши F3 открою файл lab6.asm для просмотра

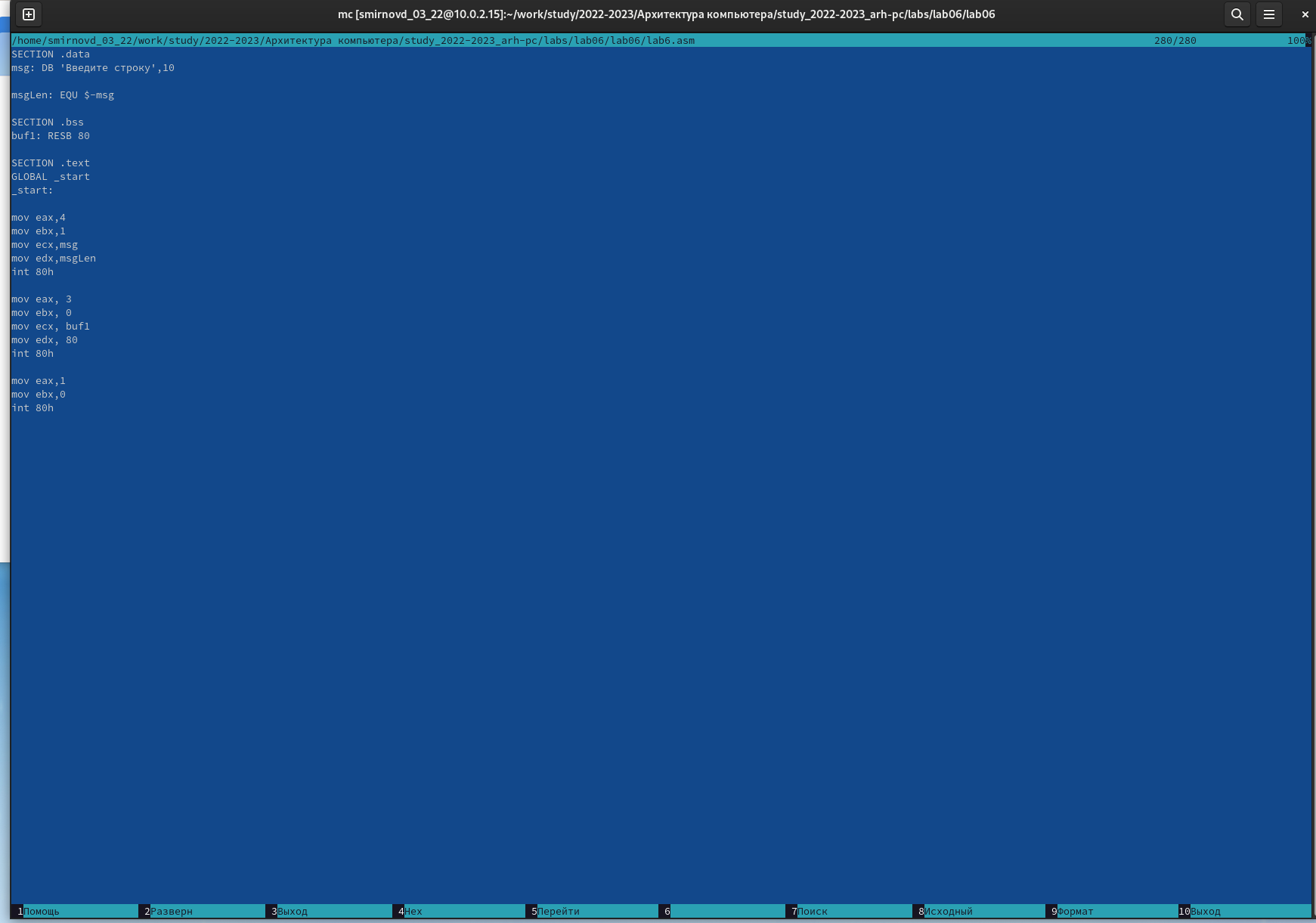


Рис. 4: Рис4

Оттранслирую текст программы lab6.asm в объектный файл. Выполню компоновку объектного файла и запущу получившийся исполняемый файл. Программа выводит строку ‘Введите строку:’ и ожидает ввода с клавиатуры. На запрос введу свои ФИО.

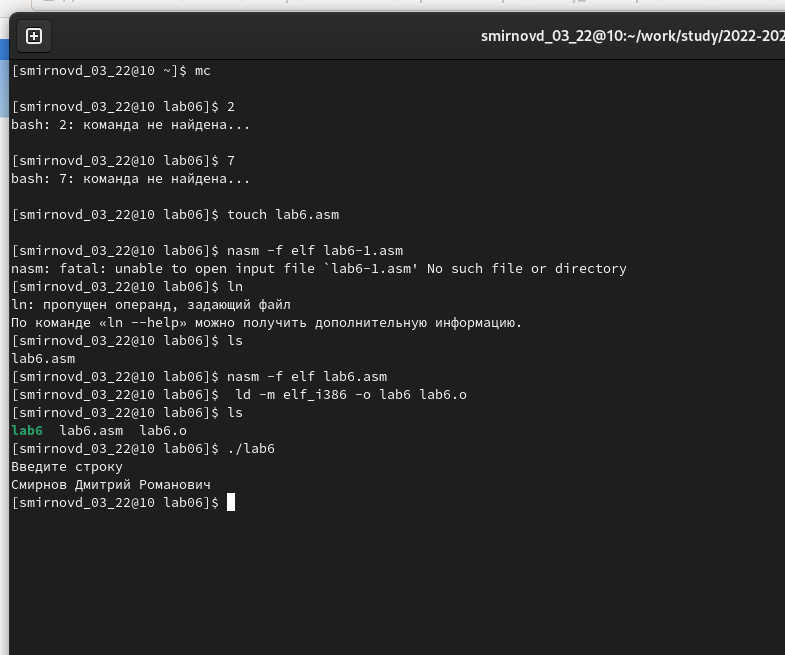


Рис. 5: Рис5

Скачаю файл in\_out.asm со страницы курса в ТУИС. И скопирую в папку /work/study/2022-2023/Архитектура компьютера/study\_2022-2023\_arh-pc/labs/lab06/lab06

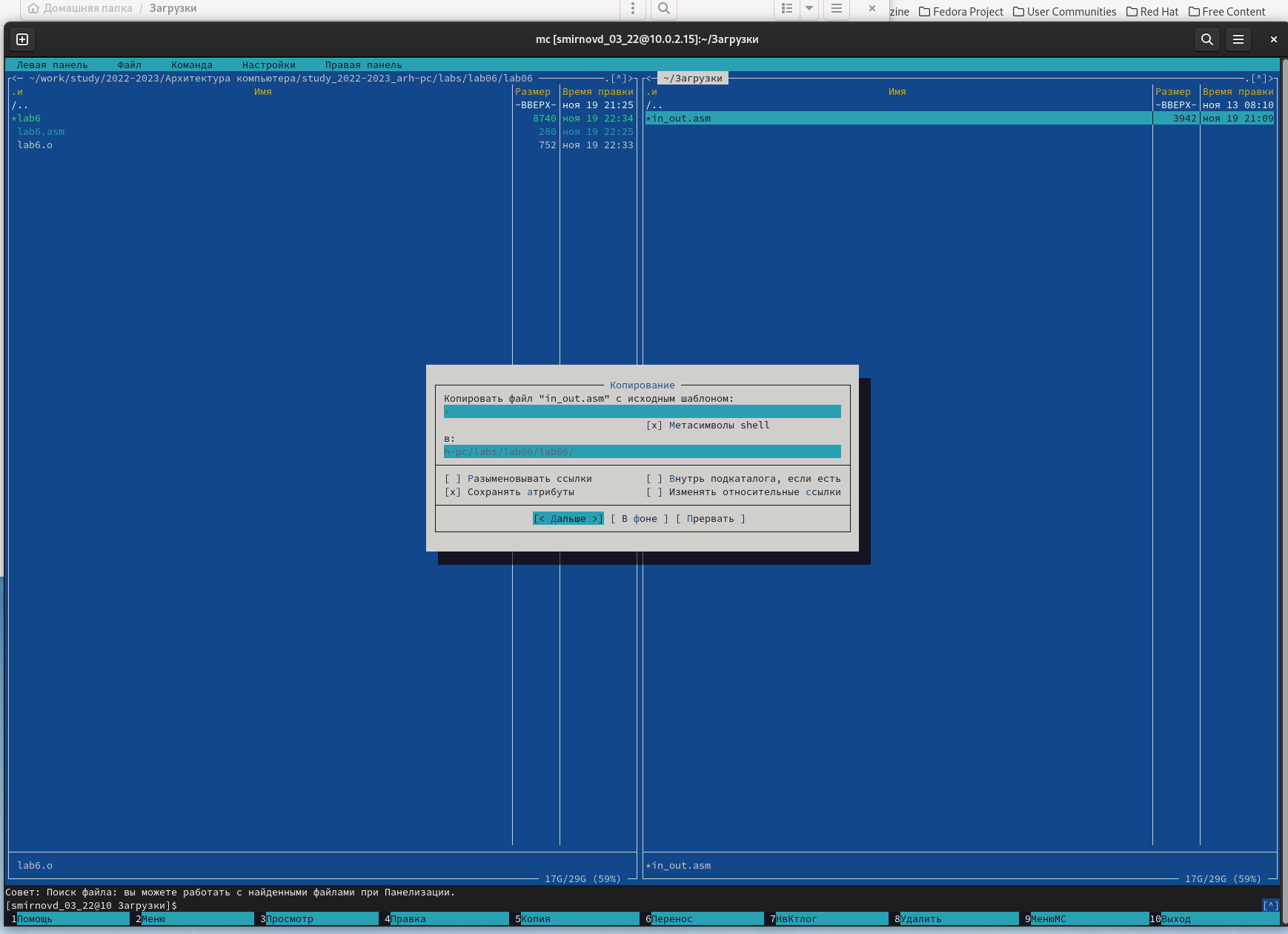


Рис. 6: Рис6

С помощью функциональной клавиши F6 создам копию файла lab6.asm с именем lab6-2.asm. Выделите файл lab6.asm, нажмите клавишу F6 , введите имя файла lab6-2.asm и нажму клавишу Enter

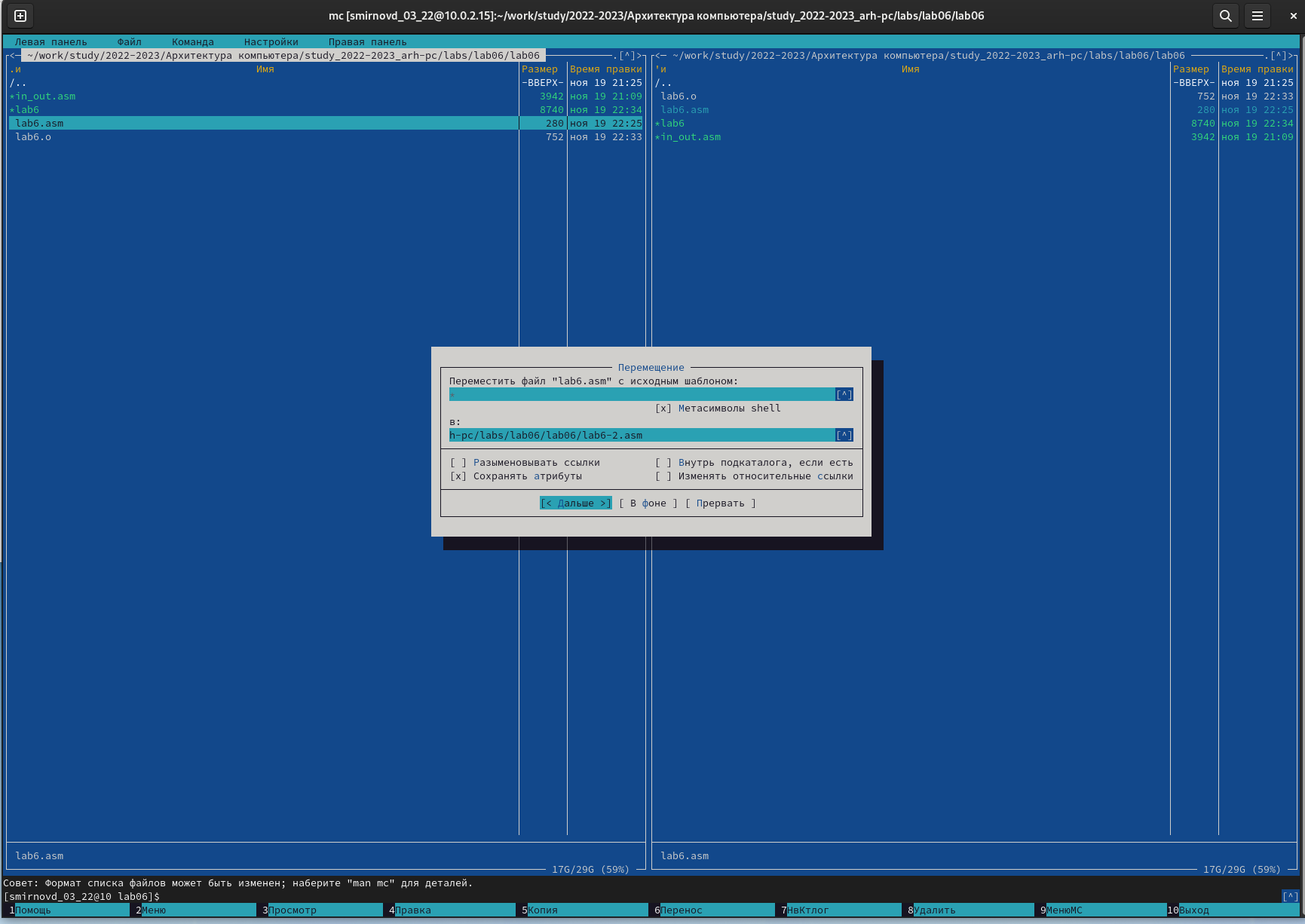


Рис. 7: Рис7

Исправлю текст программы в файле lab6-2.asm с использование подпрограмм из внешнего файла in\_out.asm и запущу программу

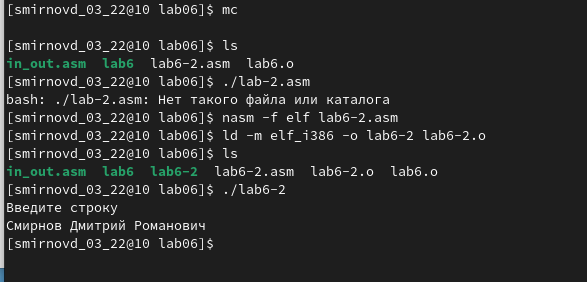


Рис. 8: Рис8

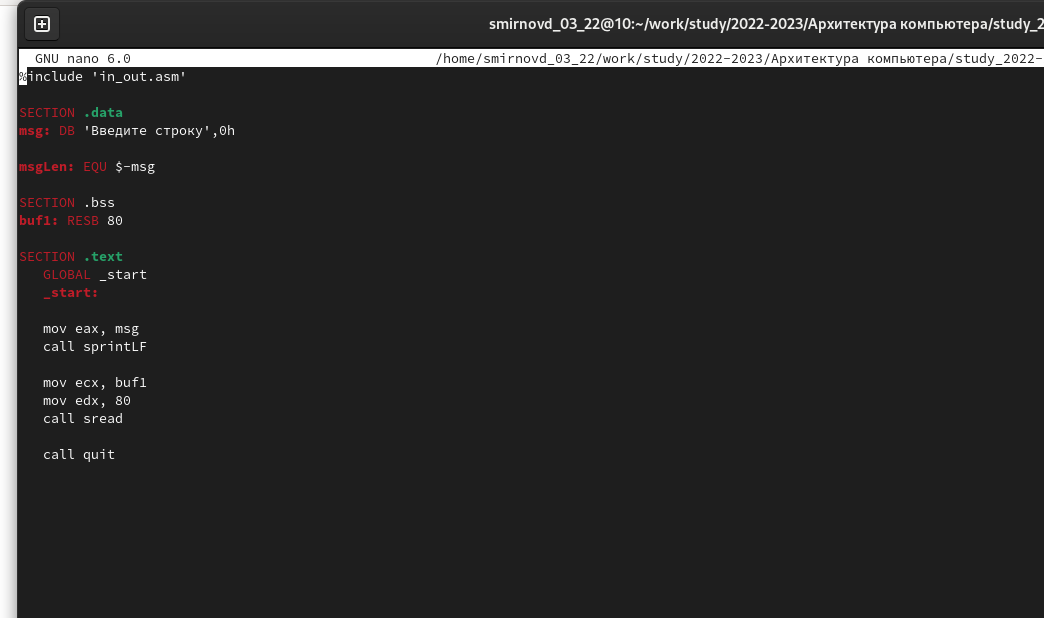


Рис. 9: Рис9

В файле lab6-2.asm заменю подпрограмму sprintLF на sprint. Создам исполняемый файл и проверю его работу

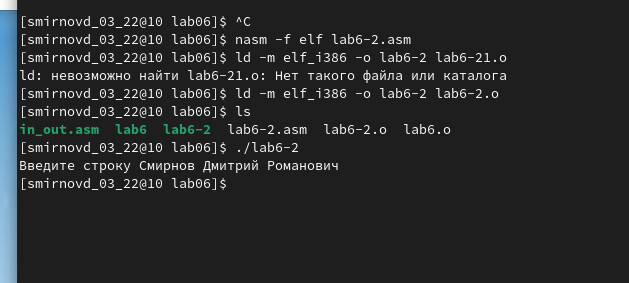


Рис. 10: Рис10

Разница в работе программ видна, вторая программа не переводит пользователя на новую строку.

# 3 Вопросы для самопроверки

1. Каково назначение mc? Основное назначение данной программы - упростить и сделать более наглядной работу с файлами в системе Linux.
2. Какие операции с файлами можно выполнить как с помощью команд bash, так и с помощью меню (комбинаций клавиш) mc? Приведите несколько примеров.
3. Какова структура программы на языке ассемблера NASM? 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).
4. Для описания каких данных используются секции bss и data в языке ассемблера NASM? Эти секции используются для определения кода запуска
5. Для чего используются компоненты db, dw, dd, dq и dt языка ассемблера NASM? DB (define byte) — определяет переменную размером в 1 байт; • DW (define word) — определяет переменную размеров 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 (учетверённое слово); • DT (define ten bytes) — определяет переменную размером в 10 байт.
6. Какое произойдёт действие при выполнении инструкции mov eax, esi?
7. Для чего используется инструкция int 80h? Системный вызов exit является обязательным в конце любой программы на языке ассемблер. Для обозначения конца программы перед вызовом инструкции int 80h необходимо поместить в регистр еах значение 1, а в регистр ebx код завершения 0.

# 4 Выводы

Я Приобрел практические навыки работы в Midnight Commander и освоил инструкций языка ассемблера mov и int.