Лабораторная работа №5

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

Борисенкова София Павловна

Содержание

Список иллюстраций

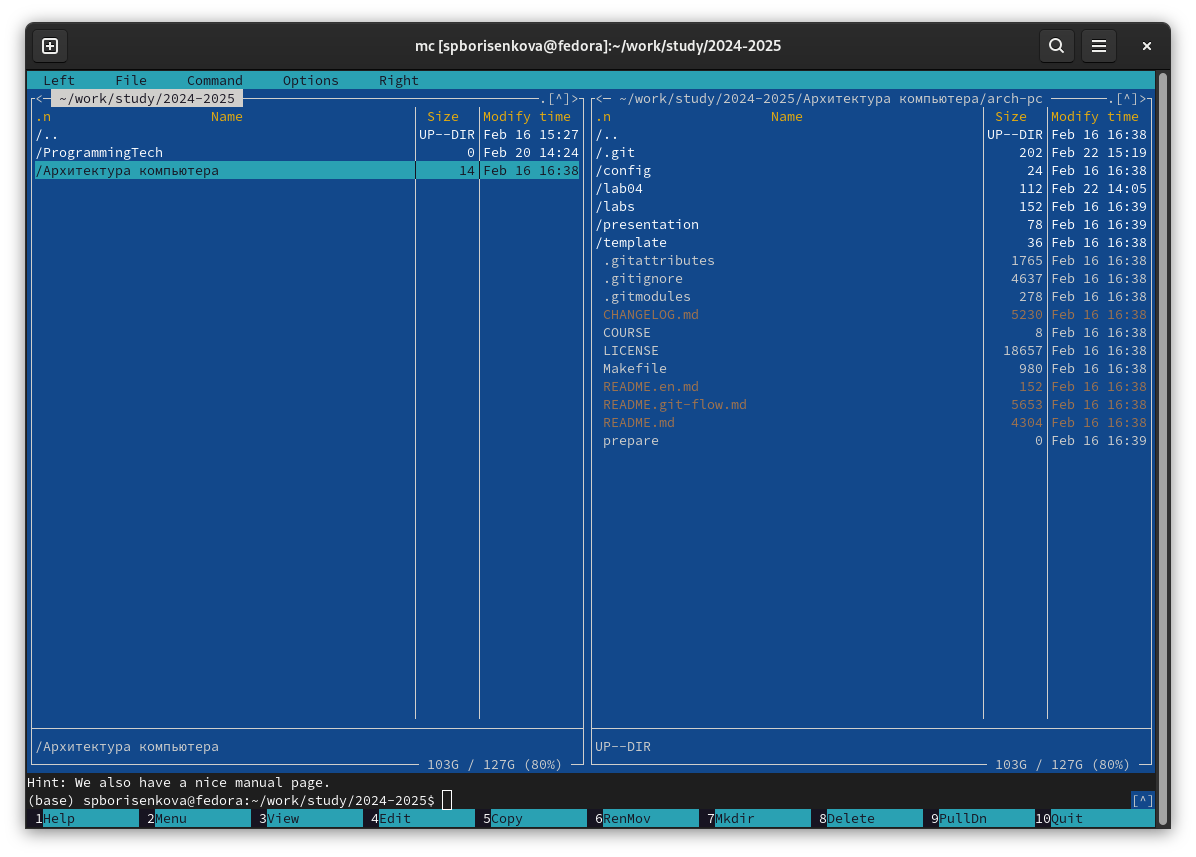
Список таблиц

# 1 Цель работы

Ознакомиться с программой Midnight commander и освоить написание программ на языке ассемблера с помощью инструкций mov и int

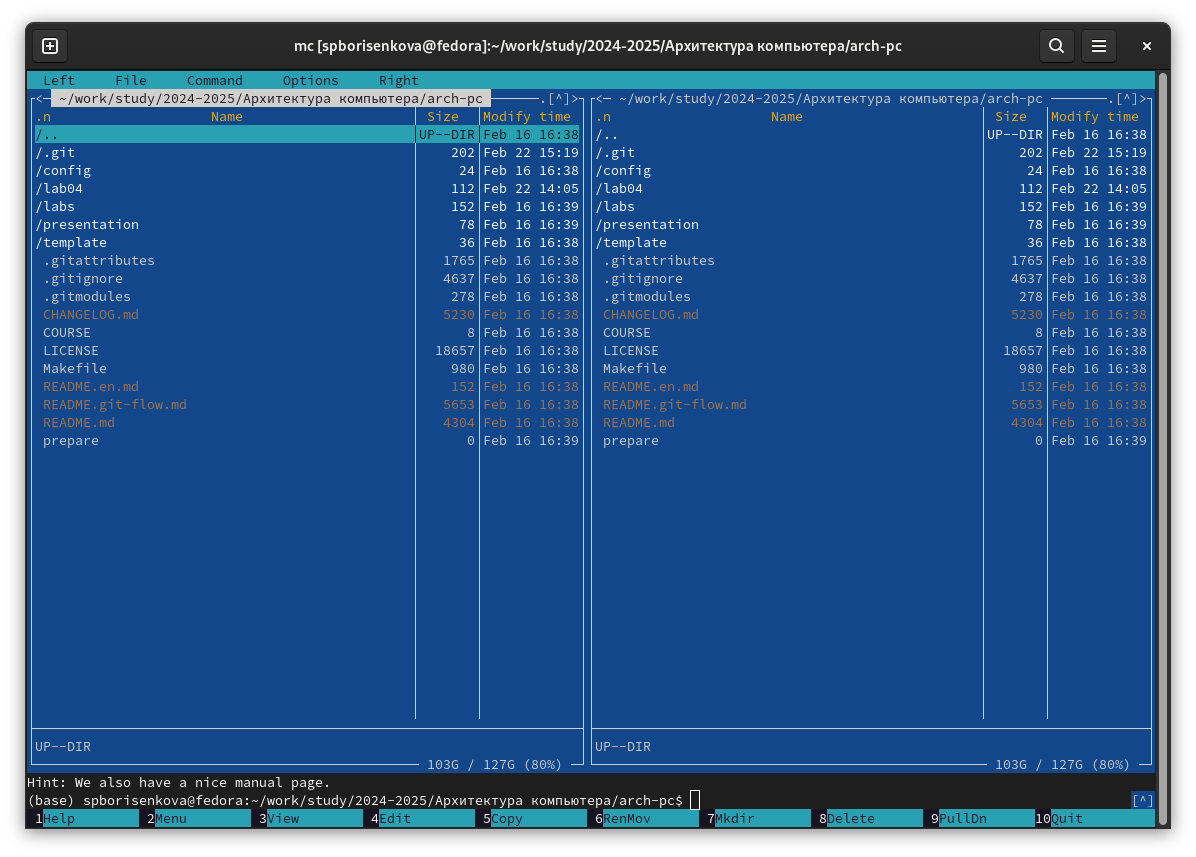
# 2 Выполнение лабораторной работы

Для начала выполнения лабораторной работы нам необходимо открыть Midnight commander с помощью команды mc. После ввода команды мы увидим такой интерфейс (Рис. 2.1):



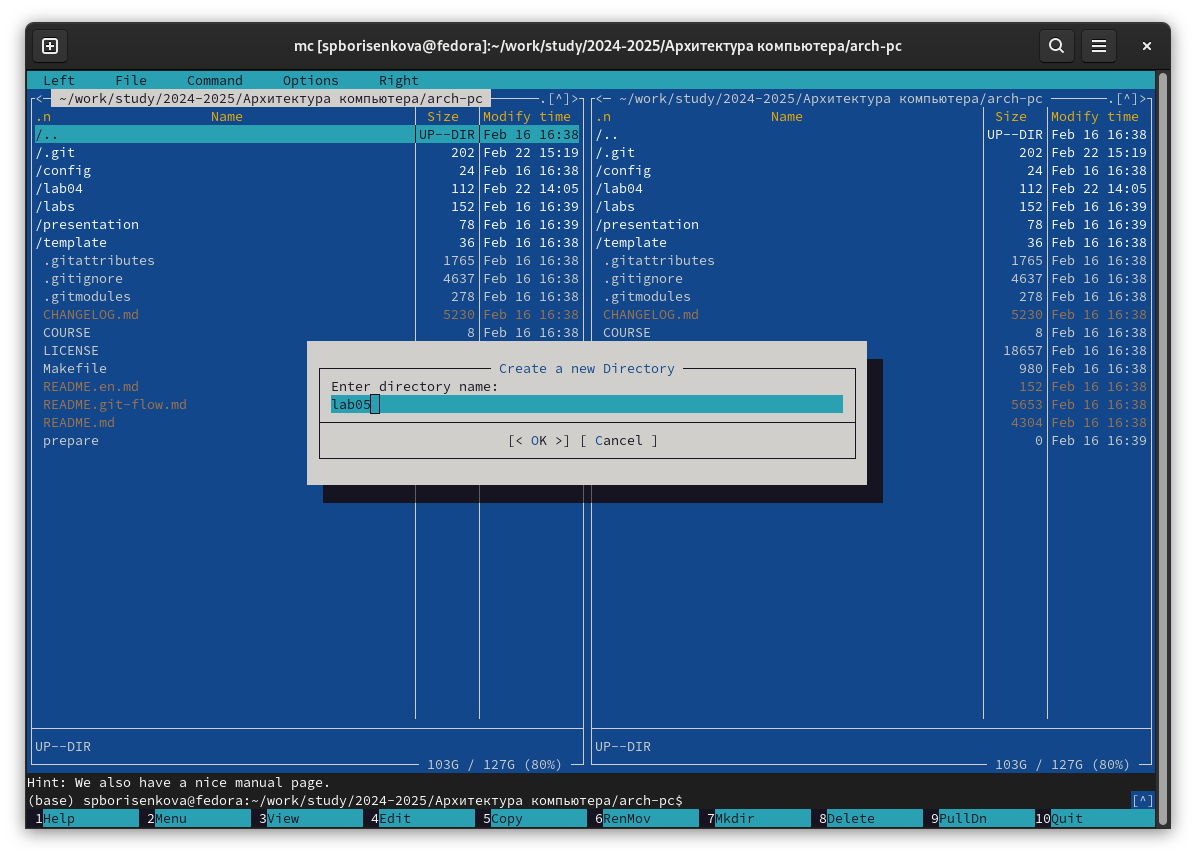
Интерфейс midnight commander

Перейдём в каталог ~/work/arch-pc (Рис. 2.2):



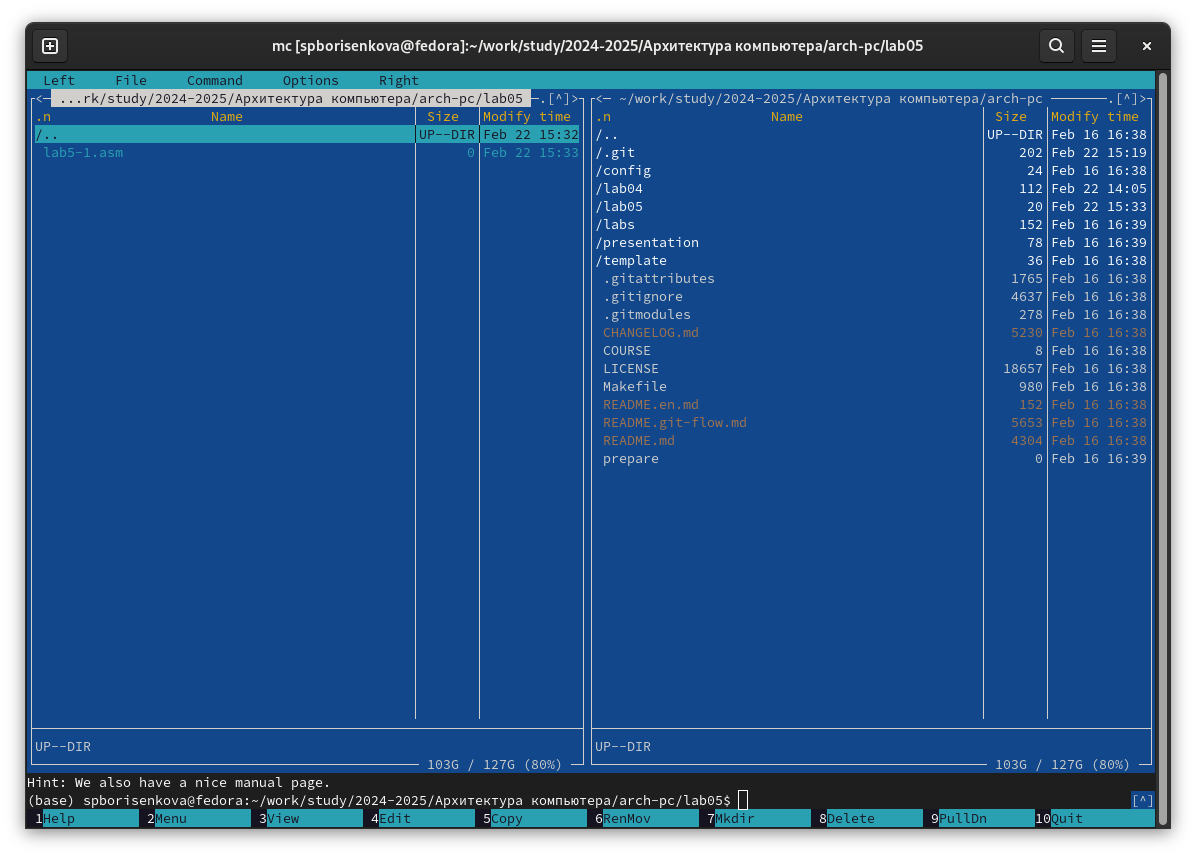
Переход в нужный каталог

Создадим папку lab05 с помощью клавиши F7 (Рис. 2.3):



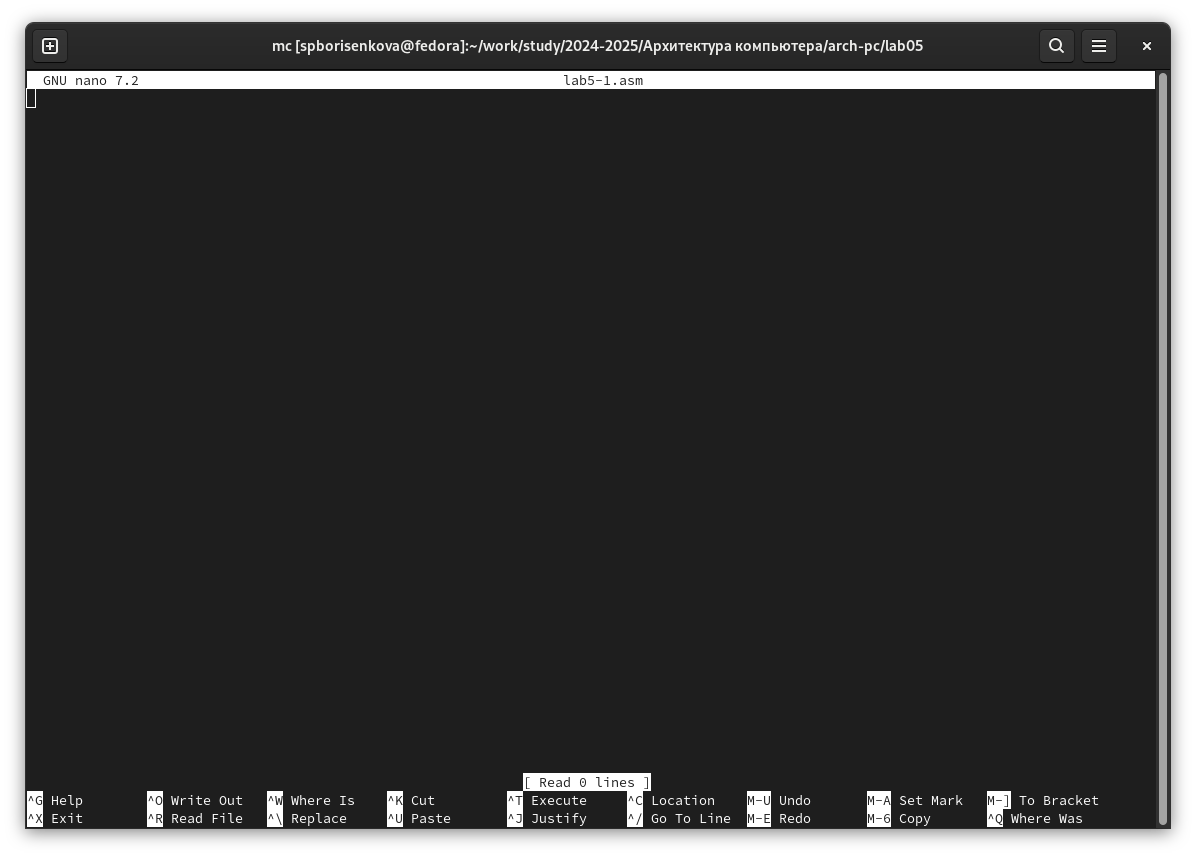
Создание папки

Теперь с помощью команды touch создадим файл lab5-1.asm (Рис. 2.4):



Создание файла lab5-1.asm с помощью команды touch

Теперь откроем только что созданный файл с помощью редактора nano (Рис. 2.5):



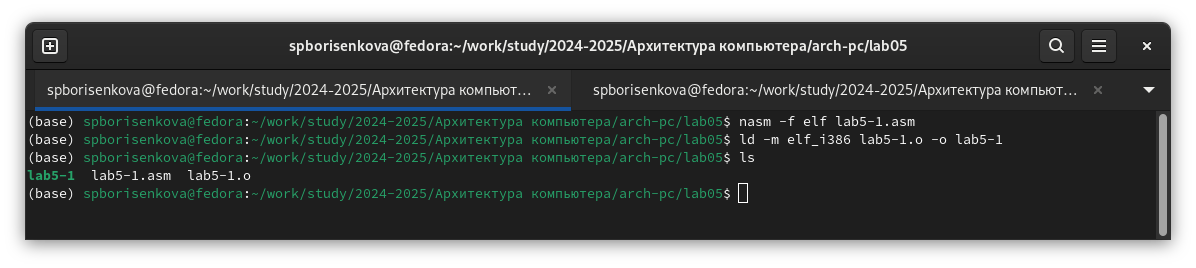
Выбор текстового редактора

Теперь отредактируем файл и поместим в него следующий код (Рис. 2.6):



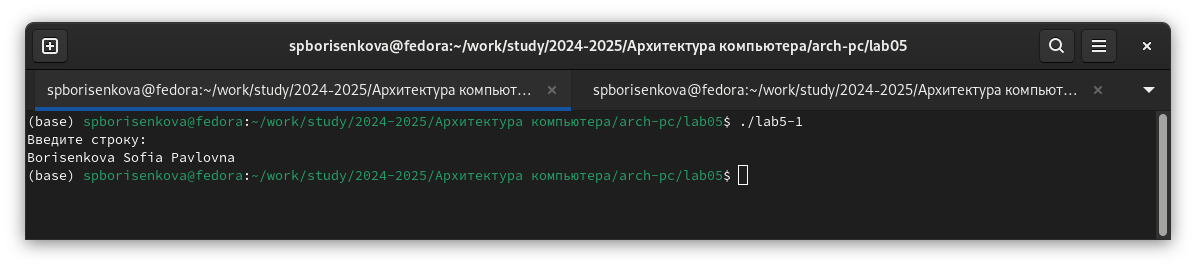
Редактирование файла lab5-1.asm

Теперь сохраним его,скомпилируем и соберём (Рис. 2.7):



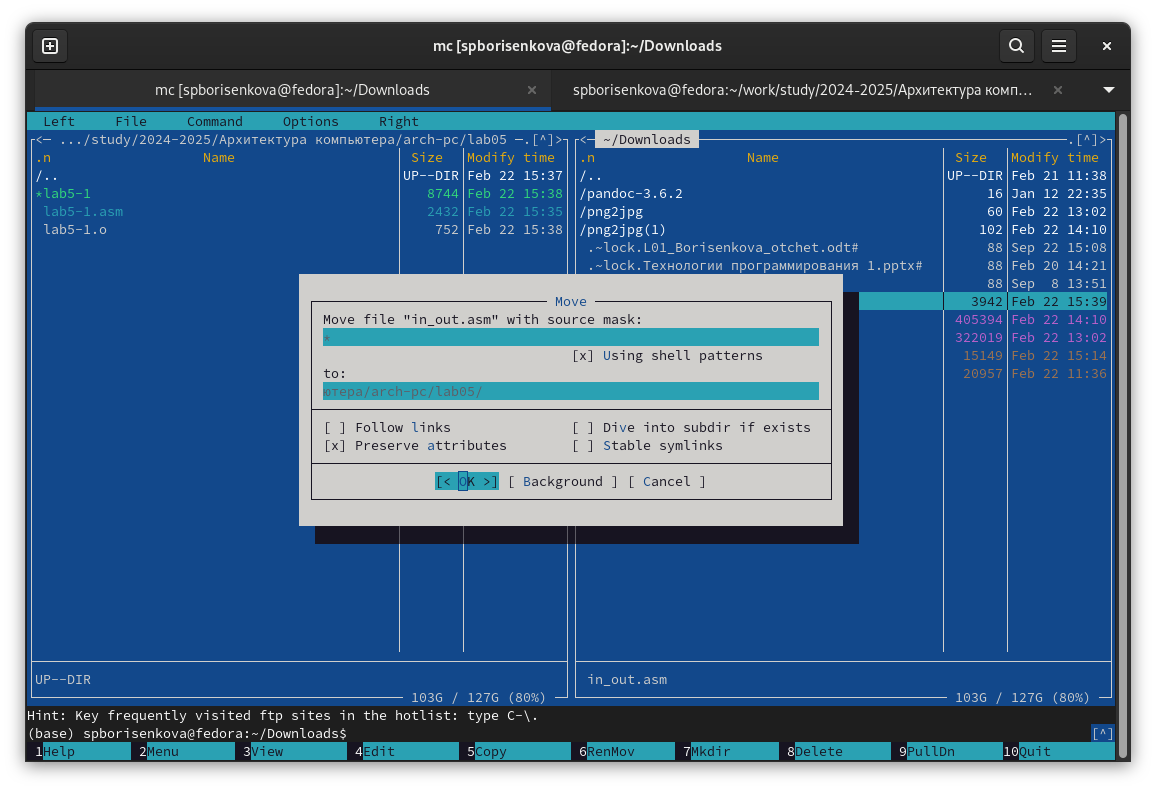
Сборка исполняемого файла

После этого запустим получившийся исполняемый файл и введём ФИО (Рис. 2.8):



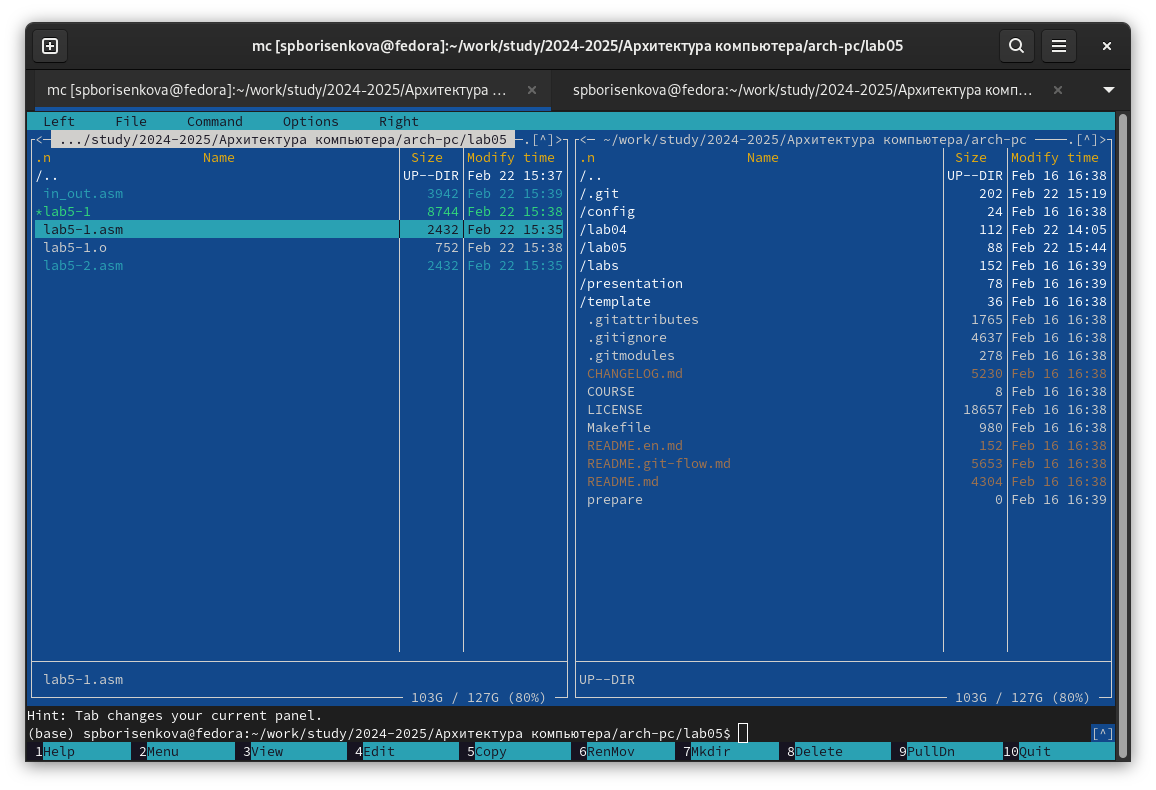
Взаимодействие с программой

После нажатия Enter программа завершится и ничего не произойдёт. Теперь скачаем файл in\_out.asm, откроем папку с ним в правой панели и скопируем его в нашу рабочую папку с помощью F6 (Рис. 2.9):



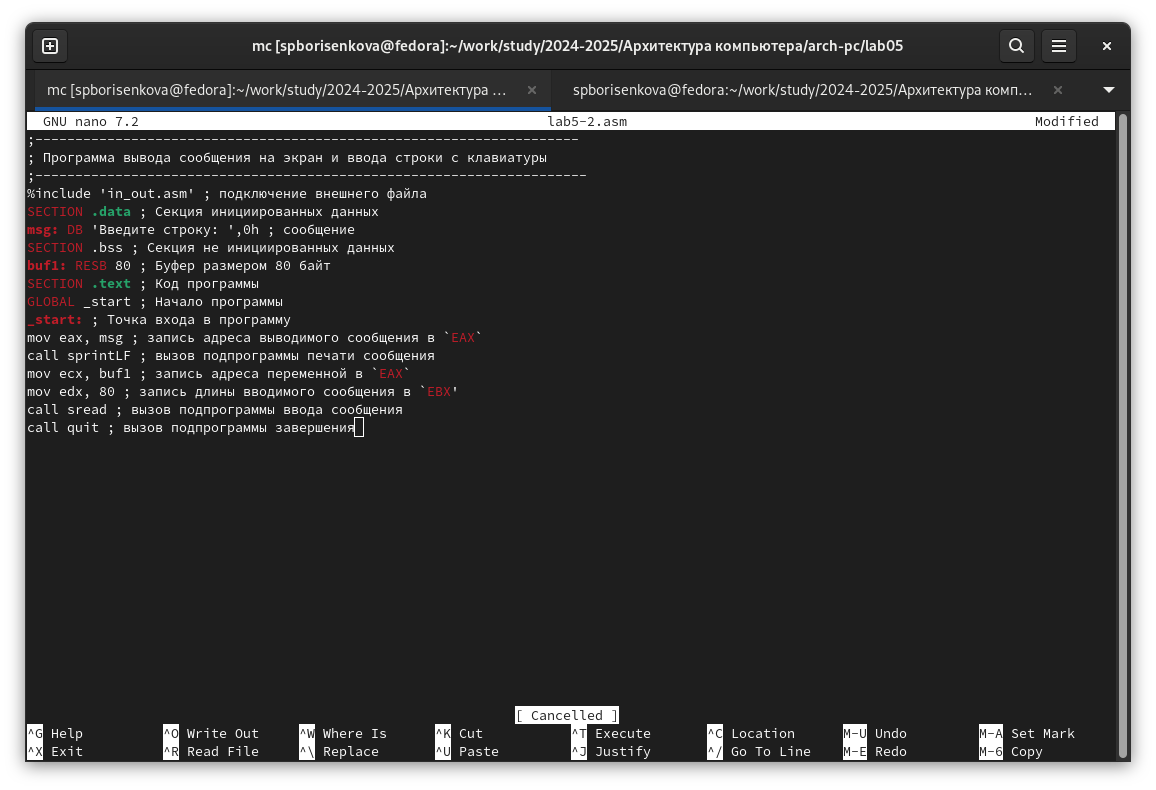
Копирование файла

Теперь сделаем копию файла lab5-1.asm с помощью команды F5. Назовём копию lab5-2.asm (Рис. 2.10):



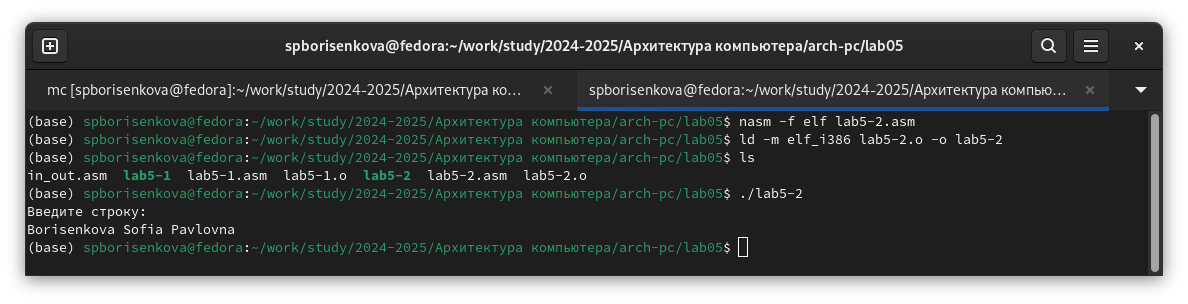
Копирование файла c помощью F5

Откроем в текстовом редакторе файл lab5-2.asm и напишем туда следующий код (Рис. 2.11):



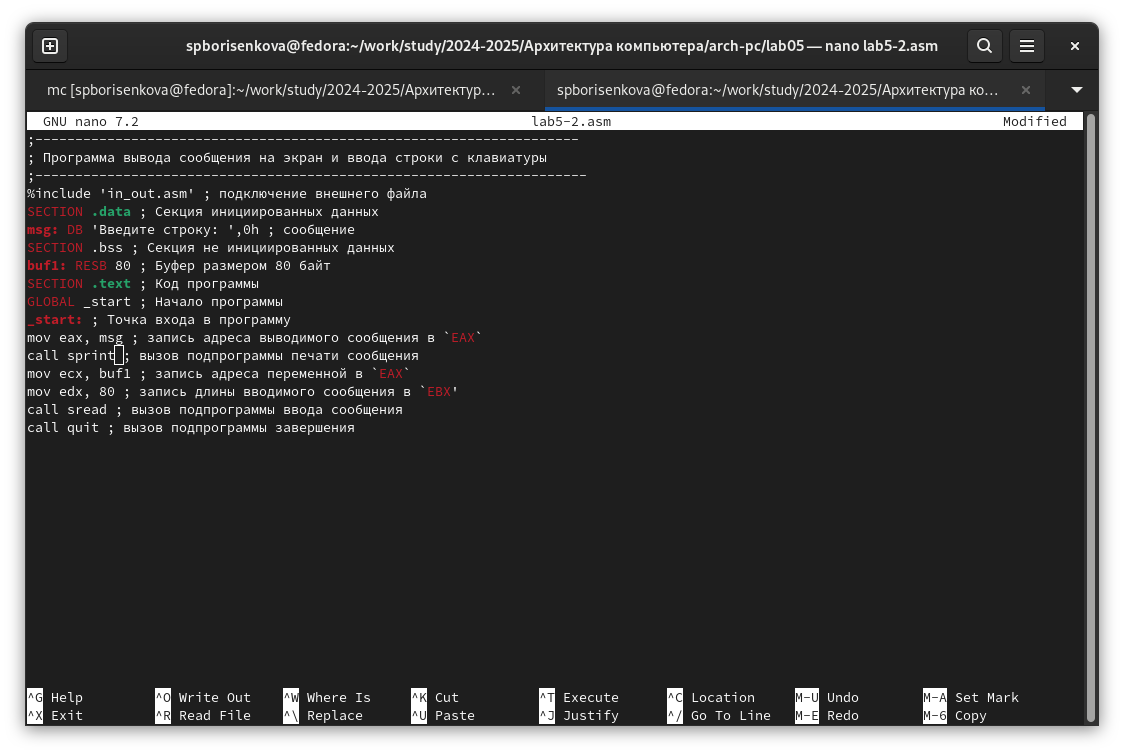
Редактирование файла lab5-2.asm

После чего создадим исполняемый файл с помощью nasm и ld и запустим созданный файл (Рис. 2.12):



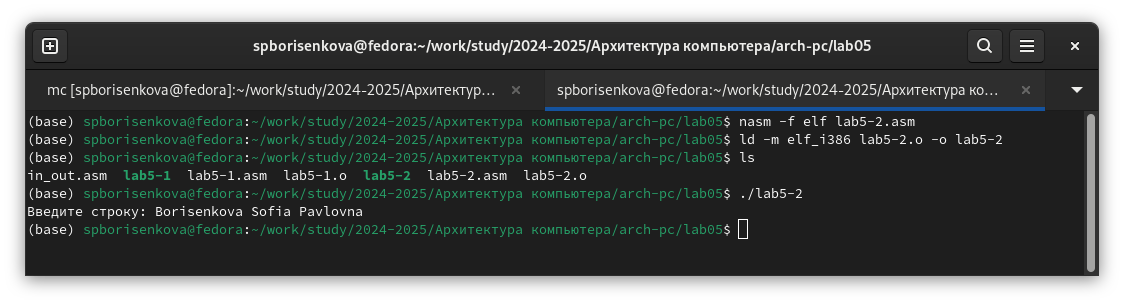
Запуск исполняемого файла

Попробуем теперь вместо команды sprintLF использовать просто команду sprint (Рис. 2.13):



Изменение файла lab5-2.asm

Точно также соберём исполняемый файл и запустим его (Рис. 2.14):

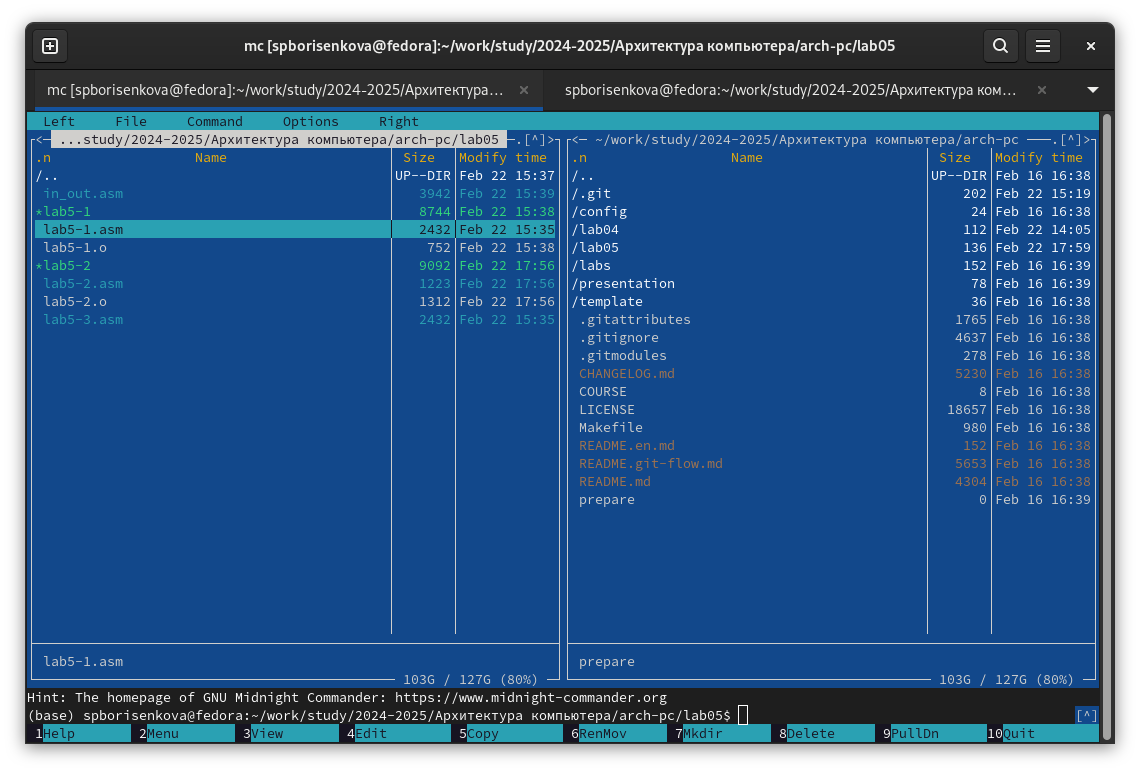


Запуск изменённого файла

Теперь нет переноса на следующую строку. Этим и отличаются команды sprintLF от sprint. Первая добавляет перенос после текста, а вторая нет

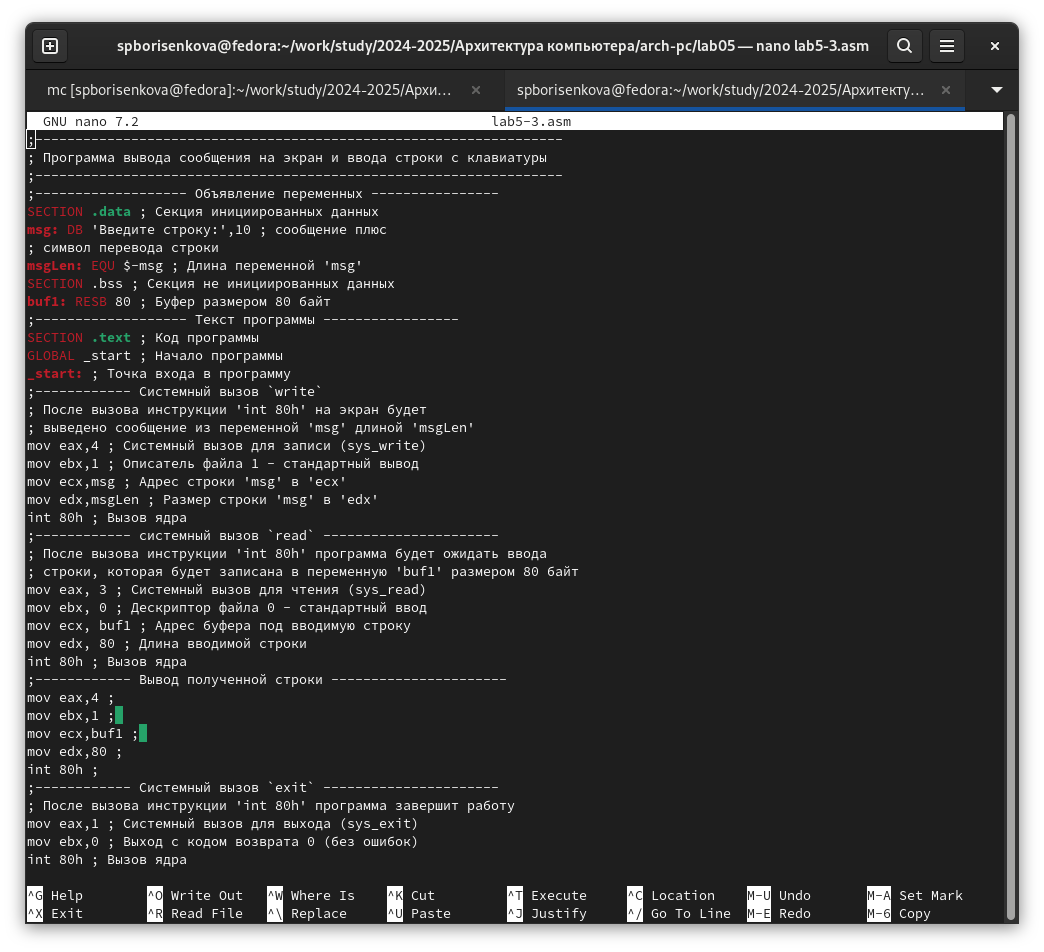
# 3 Выполнение задания для самостоятельной работы

Теперь создадим с помощью копию файла lab5-1.asm (Рис. 3.1):



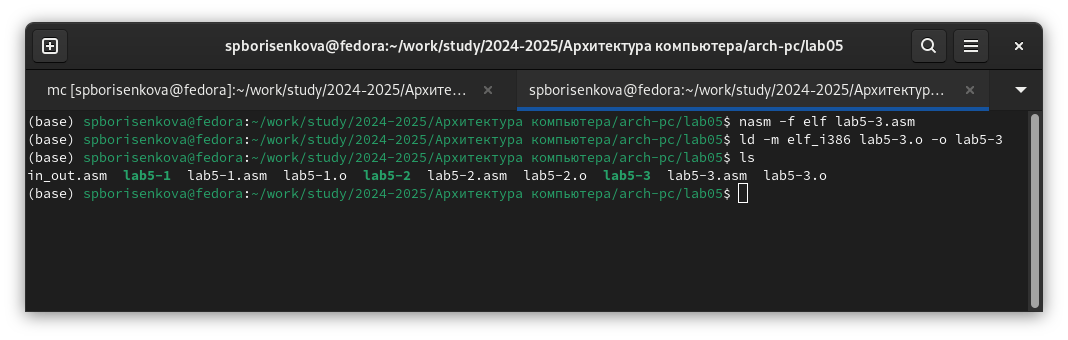
Создание файла lab5-3.asm

Изменим копию так, чтобы она выводила тот текст, который получила на ввод. Для этого перед системным вызовом exit вставим текст с системным вызовом write. Он очень похож на системный вызов write, который уже был в коде, но есть несколько отличий. Так, мы перемещаем адрес строки buf1 в ecx и размер строки buf1 (80) в edx (Рис. 3.2):



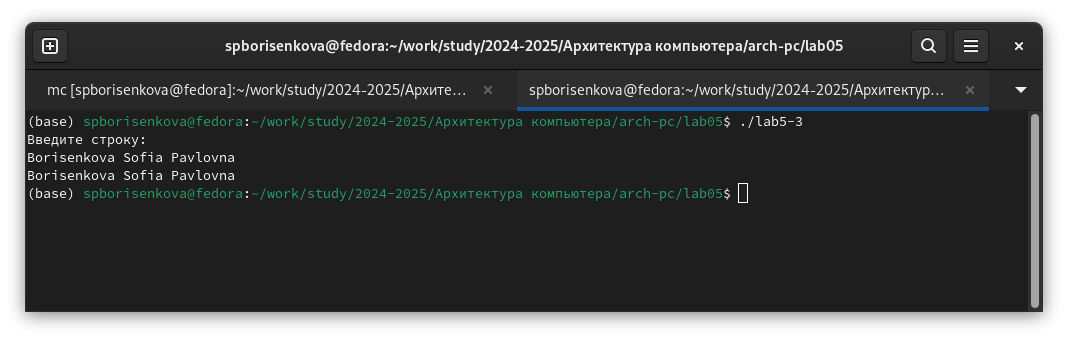
Изменение файла lab5-3.asm

Сохраним изменения и создадим исполняемый файл (Рис. 3.3):



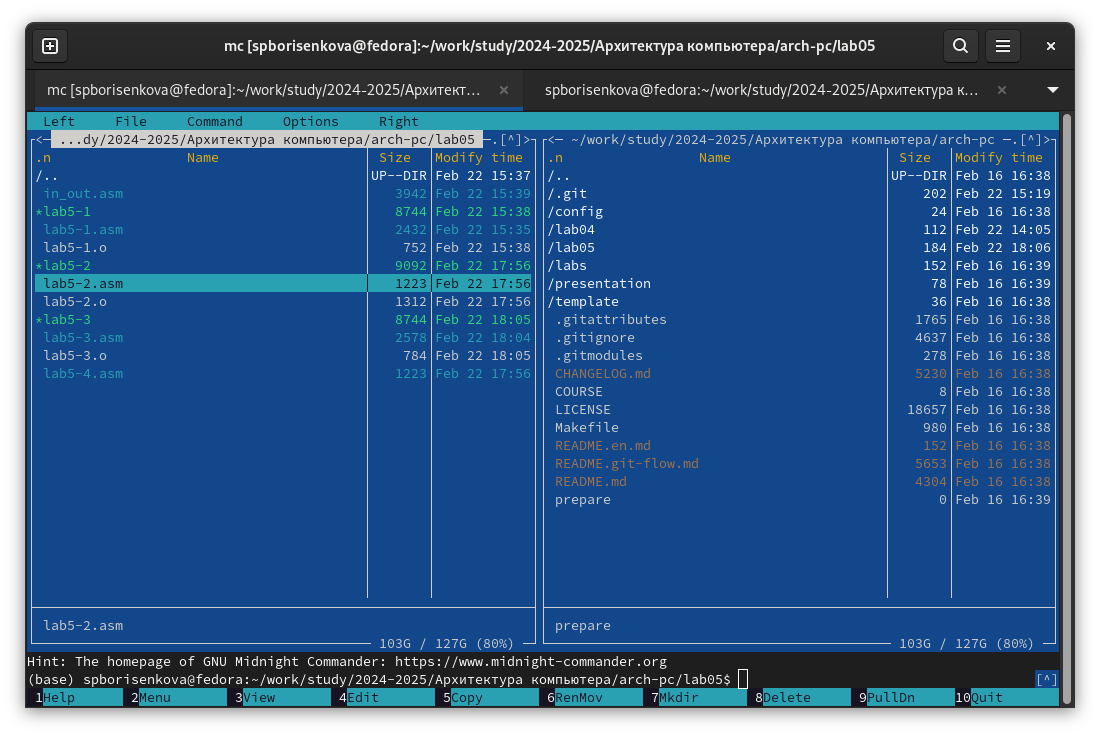
Создание исполняемого файла

Запустим его и проверим, что всё работает (Рис. 3.4):



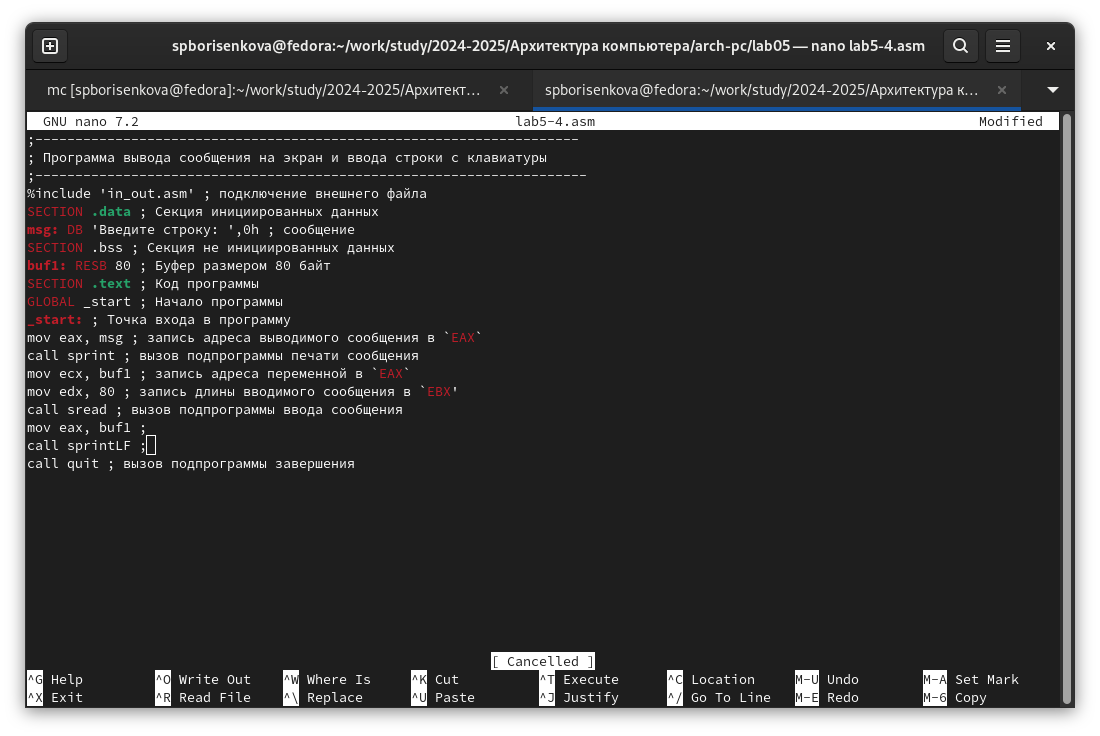
Проверка работы программы

Теперь создадим с помощью F5 копию файла lab5-2.asm (Рис. 3.5):



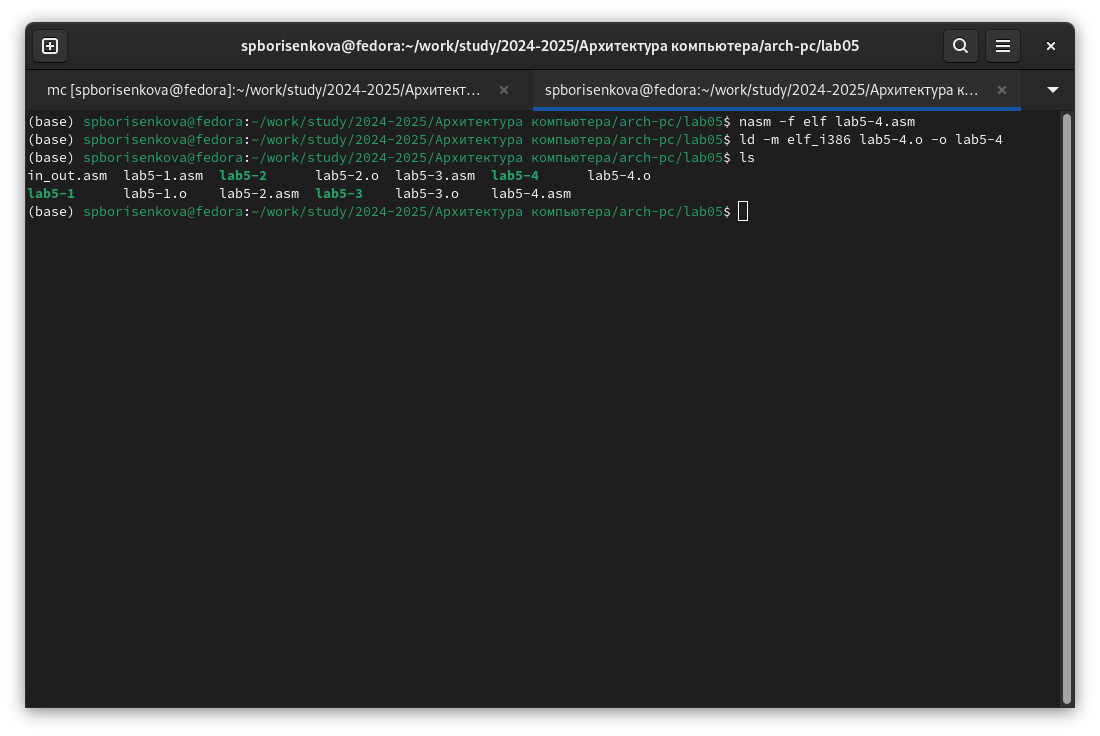
Создание файла lab5-4.asm

Сделаем так, чтобы этот код также выводил тот текст, что получит на ввод. Для этого перед последней строкой добавим строчку, которая записывает в eax адрес buf1, а также строчку, которая вызывает подпрограмму sprintLF (Рис. 3.6):



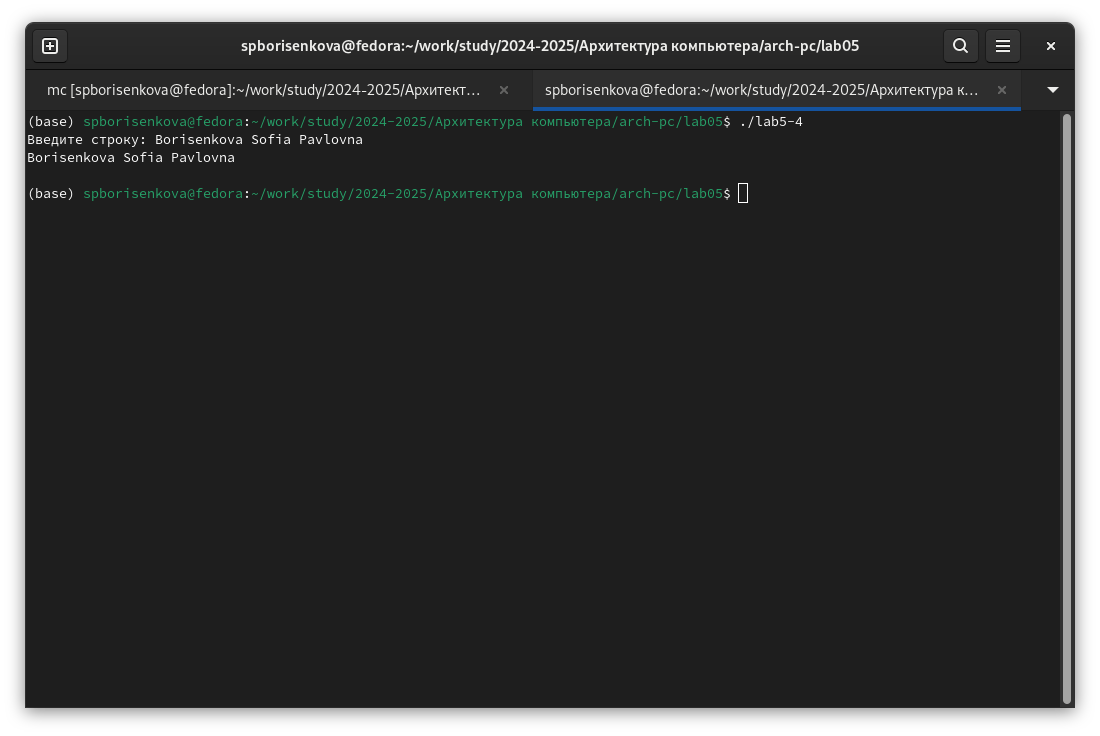
Изменение файла lab5-4.asm

Теперь создадим исполняемый файл (Рис. 3.7):



Создание исполняемого файла

Теперь запустим программу и убедимся, что она работает (Рис. 3.8):



Проверка работы программы

# 4 Выводы

В результате выполнения работы были получены навыки работы с Midnight commander, а также навыки написания простых программ ввода-вывода на языке ассемблера