Лабораторная работа №5

дисциплина: Архитектура компьютера

Савенкова Алиса Евгеньевна

Содержание

# 1 Цель работы

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

# 2 Задание

1. Основы работы с mc
2. Подключение внешнего файла in\_out.asm
3. Задания 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. 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). Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике. В общем виде эта инструкция записывается в виде: mov dst,src Здесь операнд dst — приёмник, а src — источник. В качестве операнда могут выступать регистры (register), ячейки памяти (memory) и непосредственные значения (const). Простейший диалог с пользователем требует наличия двух функций — вывода текста на экран и ввода текста с клавиатуры. Простейший способ вывести строку на экран — использовать системный вызов write. Этот системный вызов имеет номер 4, поэтому перед вызовом инструкции int необходимо поместить значение 4 в регистр eax. Первым аргументом write, помещаемым в регистр ebx, задаётся дескриптор файла. Для вывода на экран в качестве дескриптора файла нужно указать 1 (это означает «стандартный вывод», т. е. вывод на экран). Вторым аргументом задаётся адрес выводимой строки (помещаем его в регистр ecx, например, инструкцией mov ecx, msg). Строка может иметь любую длину. Последним аргументом (т.е. в регистре edx) должна задаваться максимальная длина выводимой строки. Для ввода строки с клавиатуры можно использовать аналогичный системный вызов read. Его аргументы – такие же, как у вызова write, только для «чтения» с клавиатуры используется файловый дескриптор 0 (стандартный ввод). 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 Выполнение лабораторной работы

Первым делом открываю Midnight Commander (рис. 1).

Рис. 1: Открытие mc

Рис. 1: Открытие mc

Перехожу в каталог ~/work/arch-pc созданный, при выполнении лабораторной работы №4 (рис. 2).

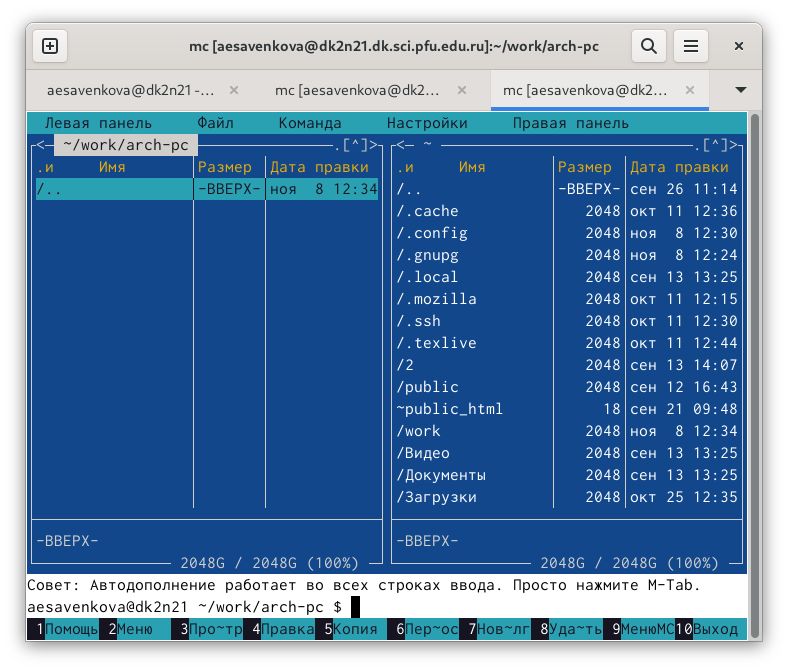


Рис. 2: mc

С помощью F7 создаю папку lab05 и перейдите в созданный каталог (рис. 3 - 4).

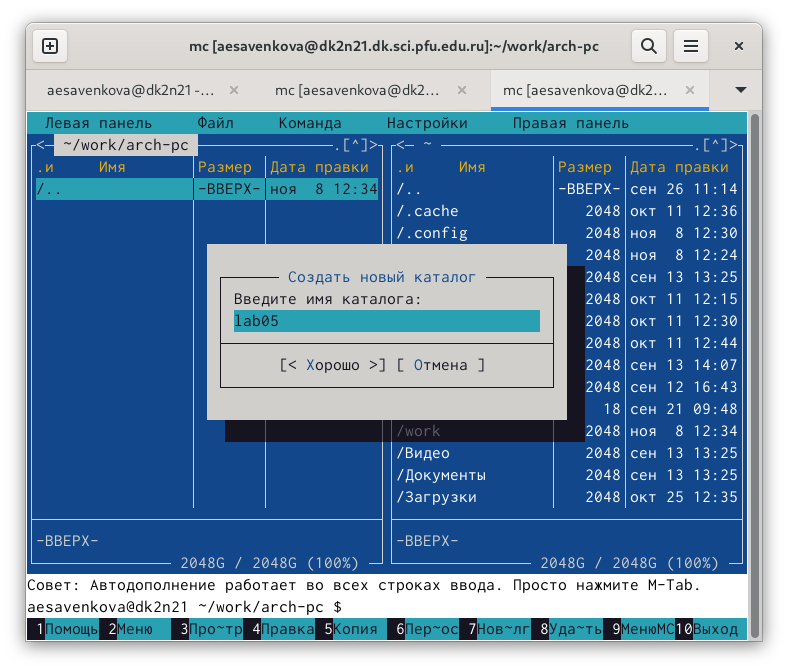


Рис. 3: Создание нового каталога

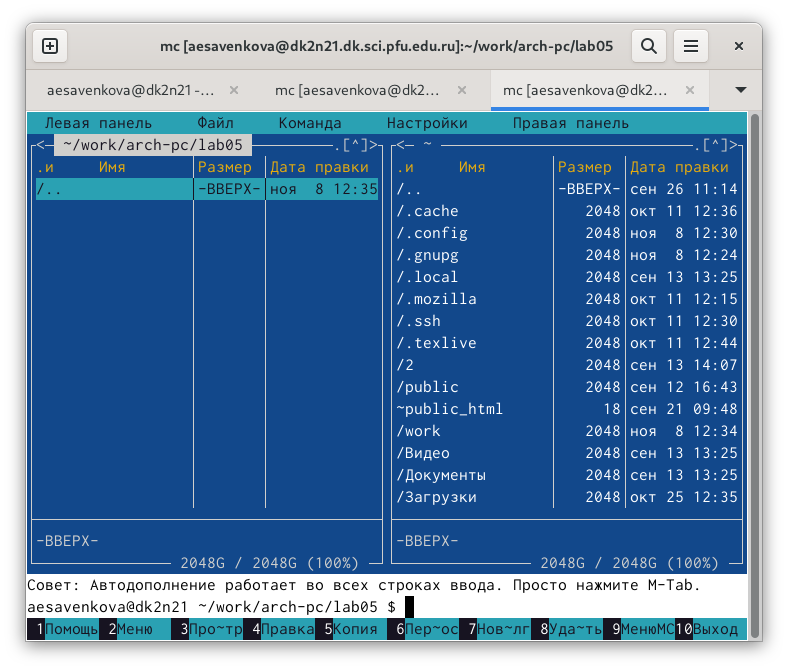


Рис. 4: Новый каталог

Пользуясь строкой ввода и командой touch создаю файл lab5-1.asm (рис. 5).

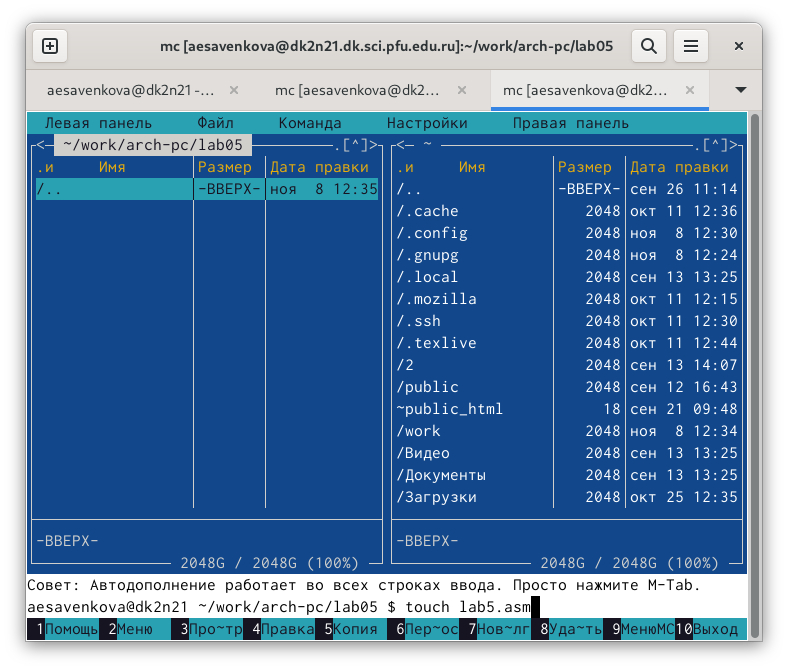


Рис. 5: Новый файл

С помощью функциональной клавиши F4 открываю файл lab5-1.asm для редактирования во встроенном редакторе (рис. 6).

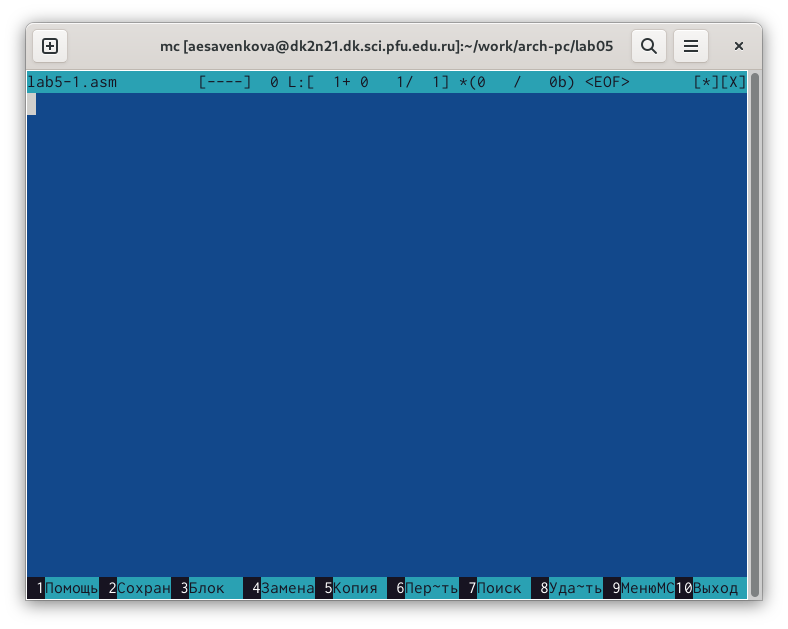


Рис. 6: Новый файл

Ввожу текст программы из листинга 5.1, сохраняю изменения и закрываю файл (рис. 7 - 8).

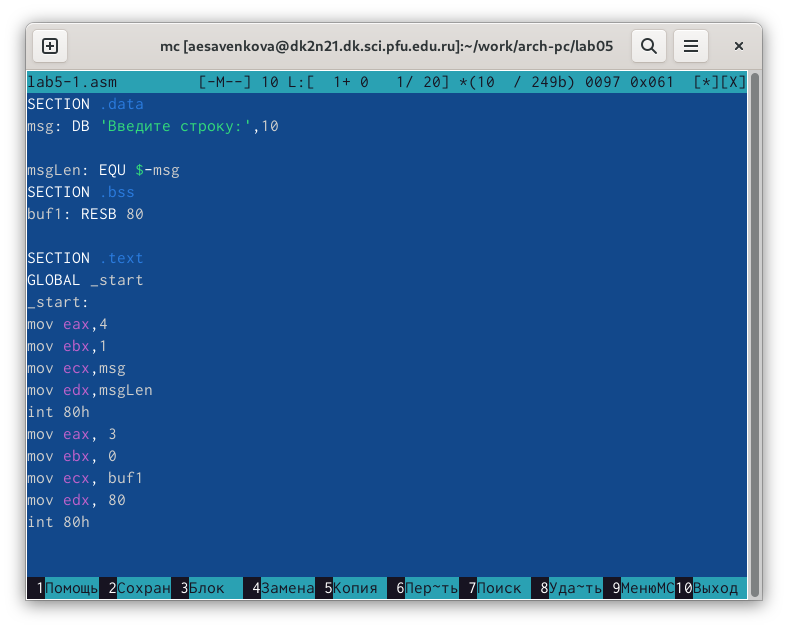


Рис. 7: Измененный файл

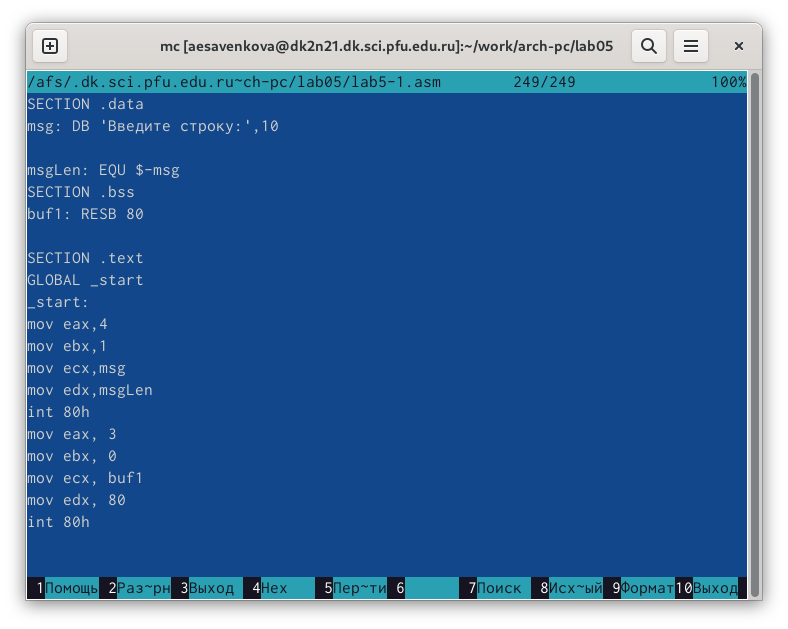


Рис. 8: Измененный файл

Транслирую 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. 9).

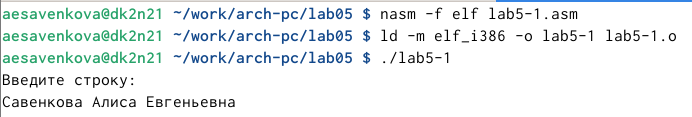


Рис. 9: Запуск файла

В одной из панелей mc открываю каталог с файлом lab5-1.asm. В другой панели каталог со скаченным файлом in\_out.asm. Копирую файл in\_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5 (рис. 10).

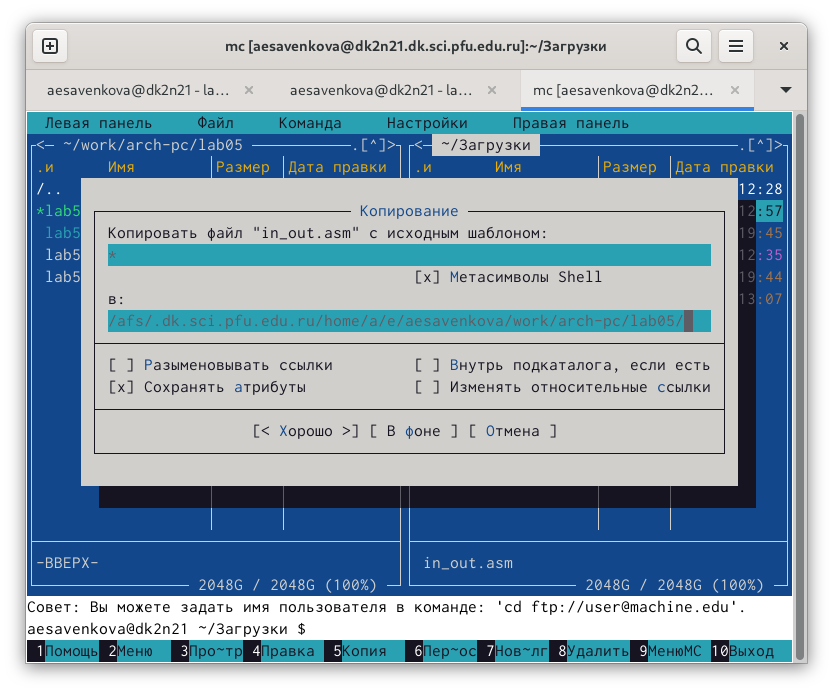


Рис. 10: Копирование файла

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. 11).

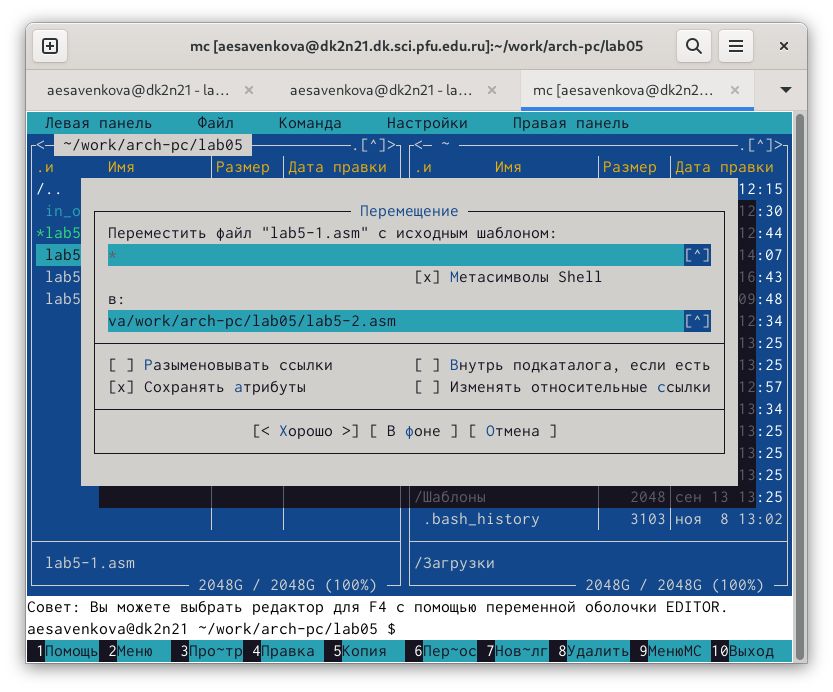


Рис. 11: Копирование файла

Исправляю текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm в соответствии с листингом 5.2. Создаю исполняемый файл и проверяю его работу (рис. 12 - 13).

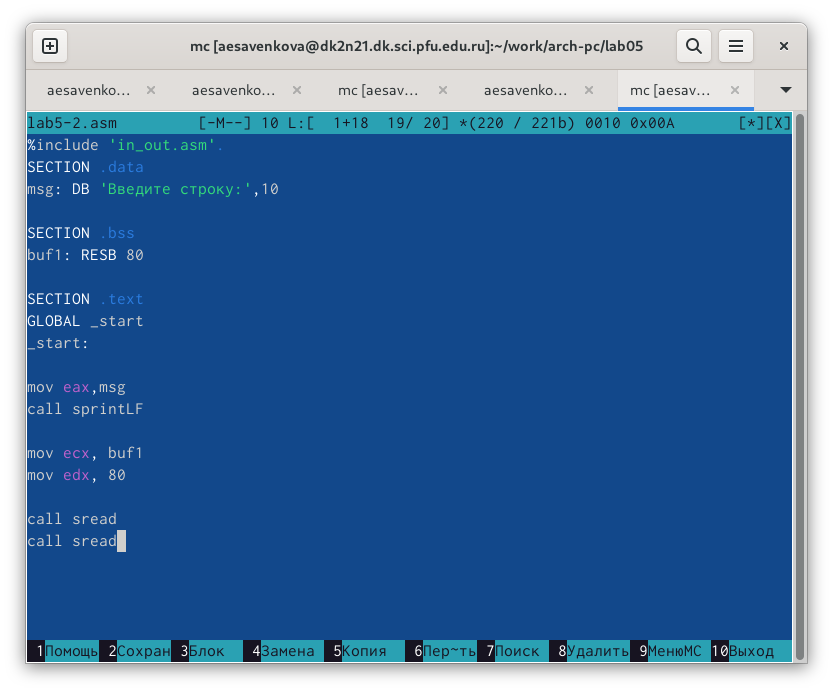


Рис. 12: Изменение файла

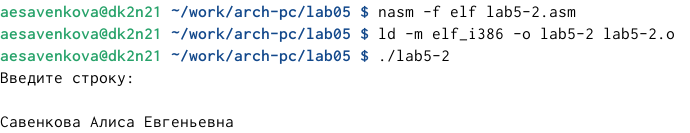


Рис. 13: Запуск файла

В файле lab5-2.asm заменяю подпрограмму sprintLF на sprint. Создаю исполняемый файл и проверяю его работу (рис. 14 - 15).

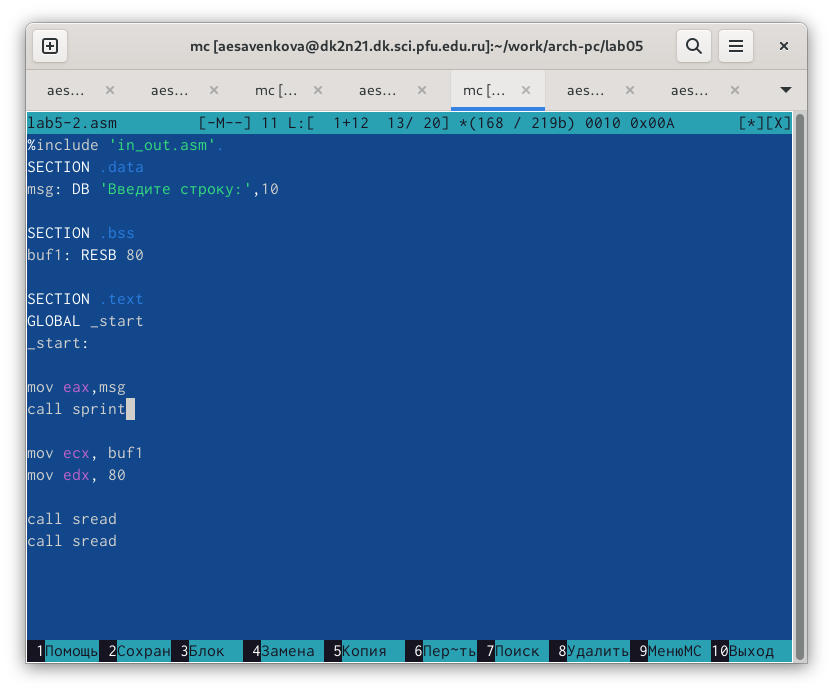


Рис. 14: Изменение файла

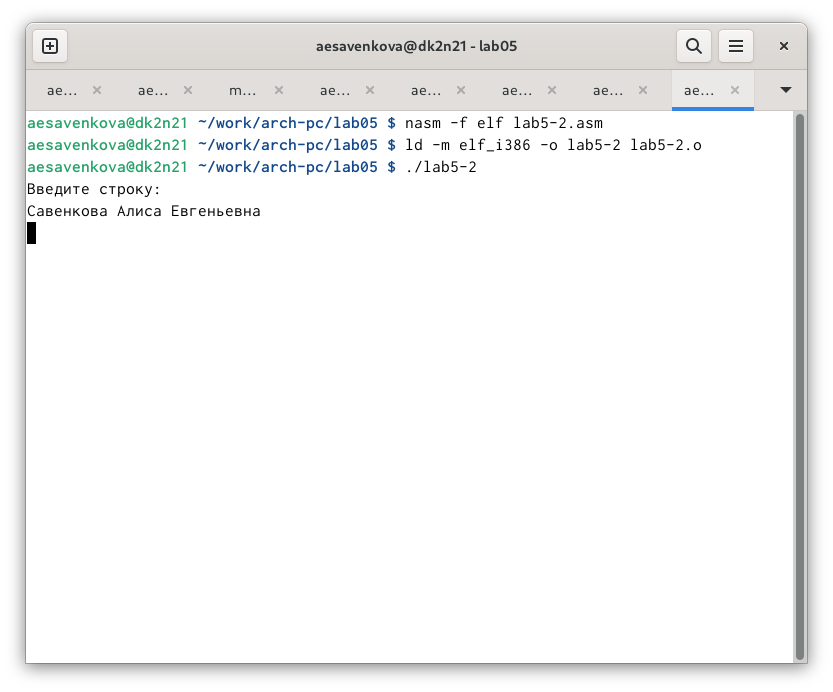


Рис. 15: Запуск файла

Разница в том, что во втором файле ввод своего ФИО происходит не через строку, а сразу после “Введите строку:”.

# 5 Задания для самостоятельной работы

Создаю копию файла lab5-1.asm. Вношу изменения в программу (рис. 16).

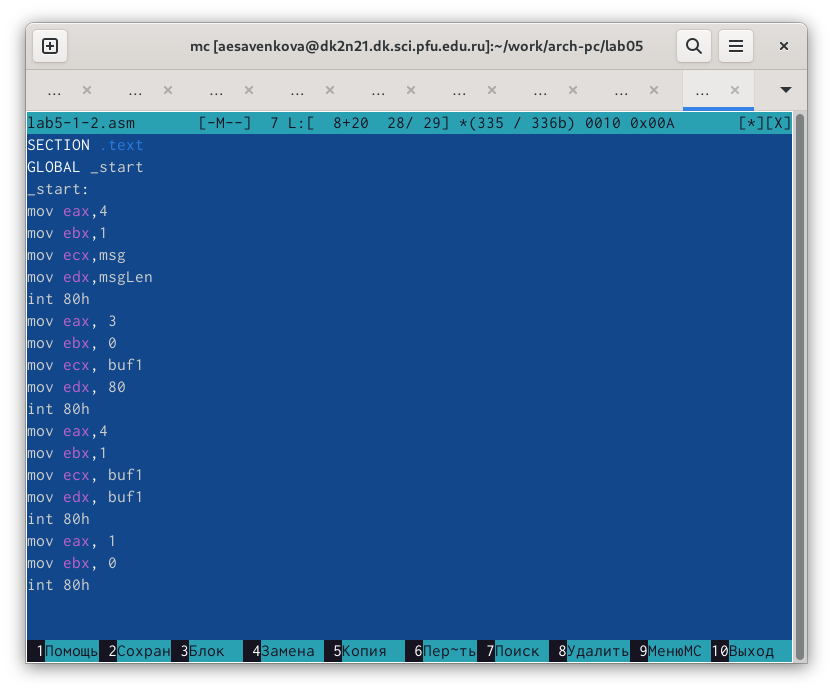


Рис. 16: Изменение файла

Получаю исполняемый файл и проверяю его работу. На приглашение ввести строку ввожу свою фамилию (рис. 17).



Рис. 17: Запуск файла

Создаю копию файла lab5-2.asm. Исправляю текст программы с использование подпрограмм из внешнего файла in\_out.asm (рис. 18).

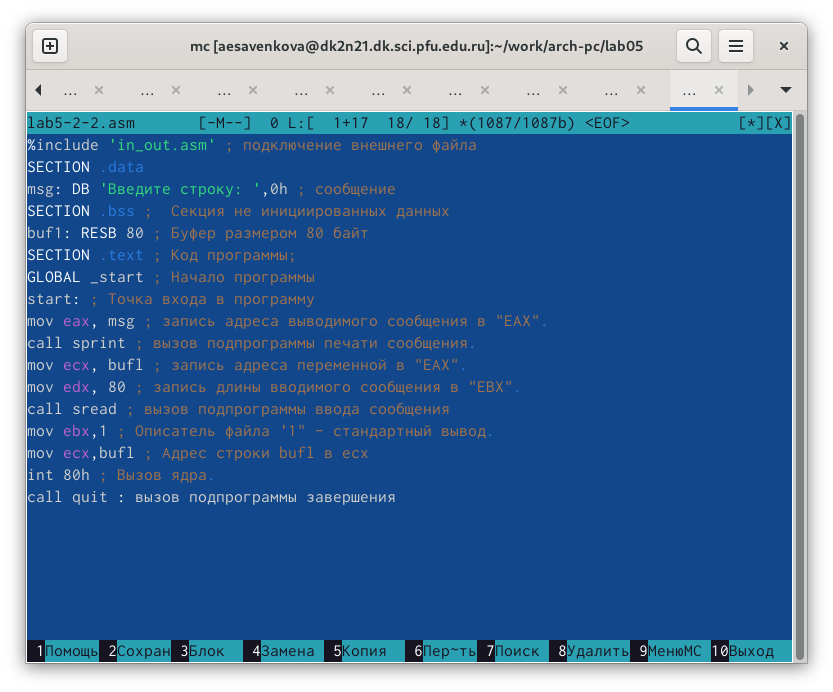


Рис. 18: Создание копии файла

Проверяю работу файла (рис. 19).



Рис. 19: Запуск файла

# 6 Выводы

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

# Список литературы