Отчет по лабораторной работе №5

Дисциплина: архитектура компьютера

Учаева Алёна Сергеевна

Содержание

Список иллюстраций

# 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. 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. 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 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. 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). Инструкция языка ассемблера intпредназначена для вызова прерывания с указанным номером. В общем виде она записывается в виде int n Здесь n — номер прерывания, принадлежащий диапазону 0–255

# 4 Выполнение лабораторной работы

## 4.1 Основы работы с mc

Открываю Midnight Commander, с помощью команды mc (рис. 1).

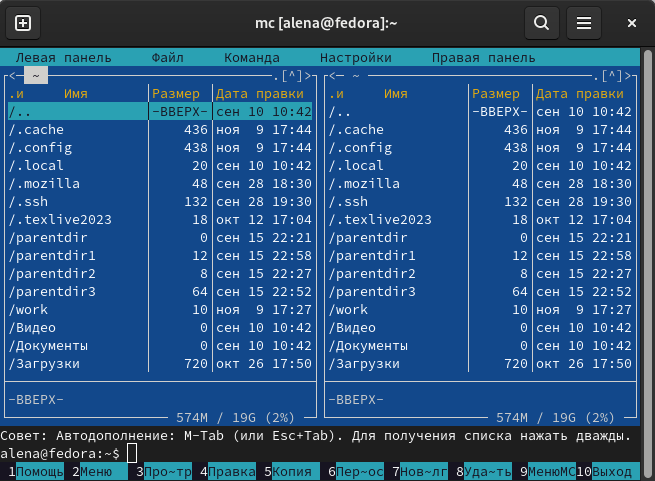


Рис. 1: Открытый mc

Перехожу в каталог ~/work/study/2024-2025/Архитектура компьютера/arch-pc (рис. 2).

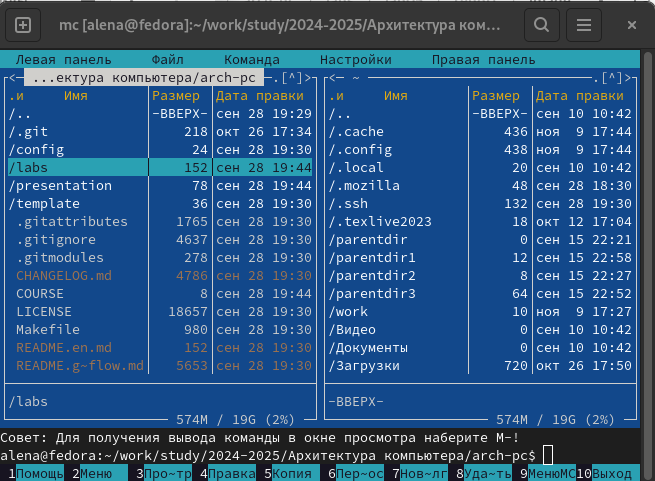


Рис. 2: Перемещение между директориями

Создаю каталог lab05 с помощью функциональной клавиши F7 (рис. 3).

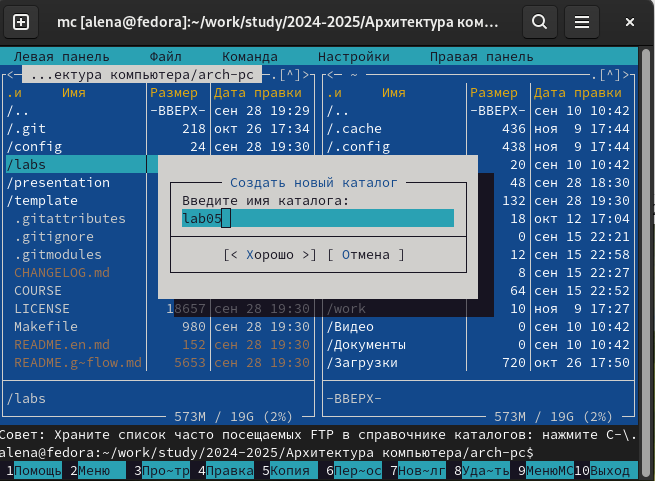


Рис. 3: Создание каталога

Перехожу в созданный каталог и в строке ввода прописываю команду touch lab5-1.asm, чтобы создать файл (рис. 4).

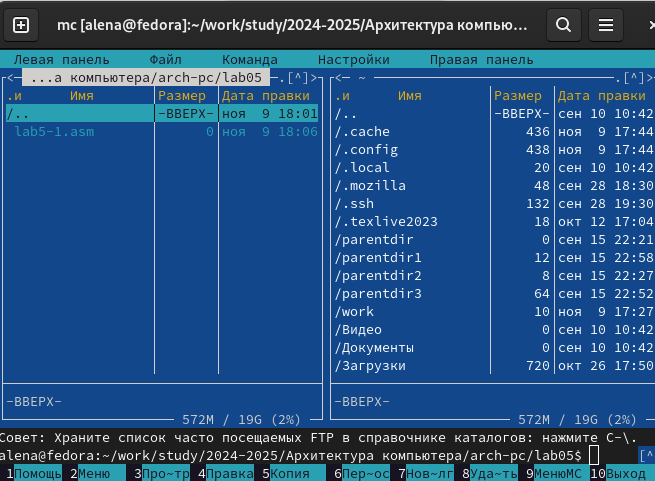


Рис. 4: Создание файла

## 4.2 Структура программы на языке ассемблера NASM

С помощью функциональной клавиши открываю созданный файл для редактирования во встроенном редакторе (рис. 5).

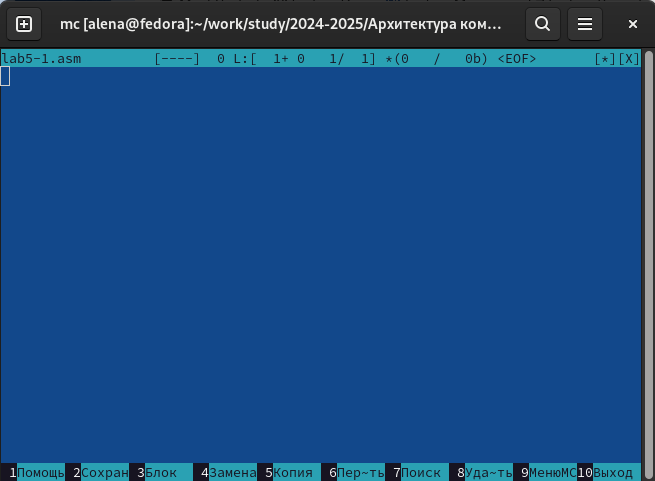


Рис. 5: Открытие файла для редактирования

Далее ввожу код программы, сохраняю изменения и сохраняю файл (рис. 6).

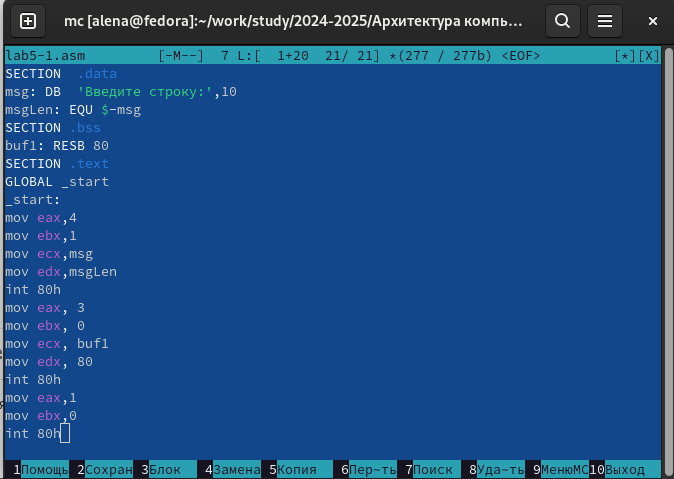


Рис. 6: Редактирование файла

С помощью функциональной клавиши F3 открываю файл и провреяю, что файл содержит текст программы (рис. 7).

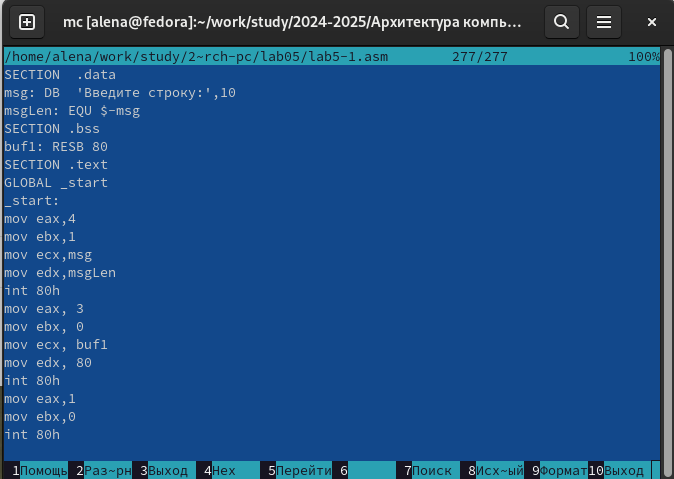


Рис. 7: Открытие файла для проверки

Транслирую текст программы в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл. На запрос ввожу свое ФИО (рис. 8).

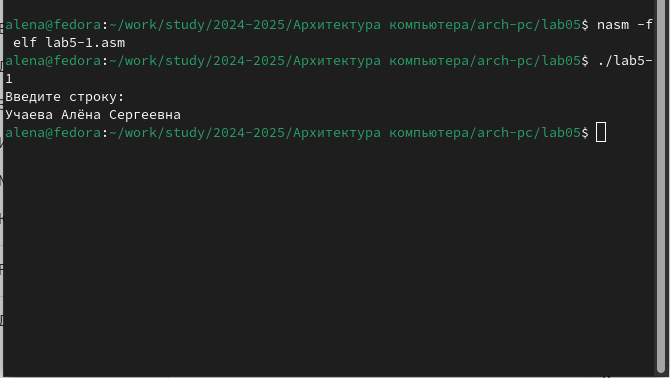


Рис. 8: Исполнение файла

## 4.3 Подключение внешнего файла

Скачиваю файл in\_out.asm со страницы курса в ТУИС и перемещаю в созданный каталог lab05 (рис. 9).

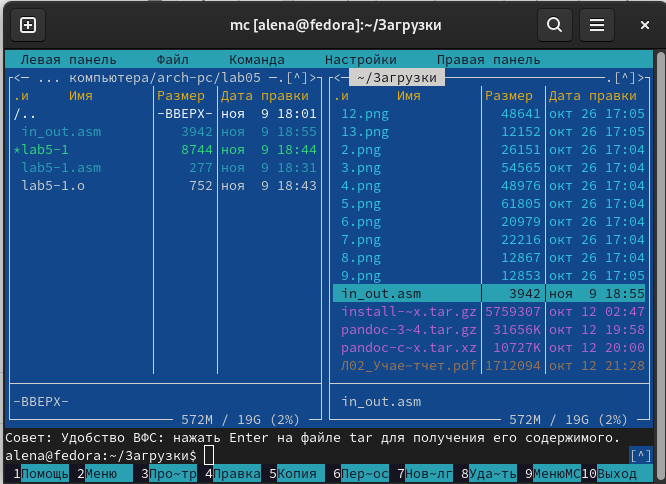


Рис. 9: Копирование файла

С помощью функциональной клавиши F5 создаю копию файла с другим именем (рис. 10).

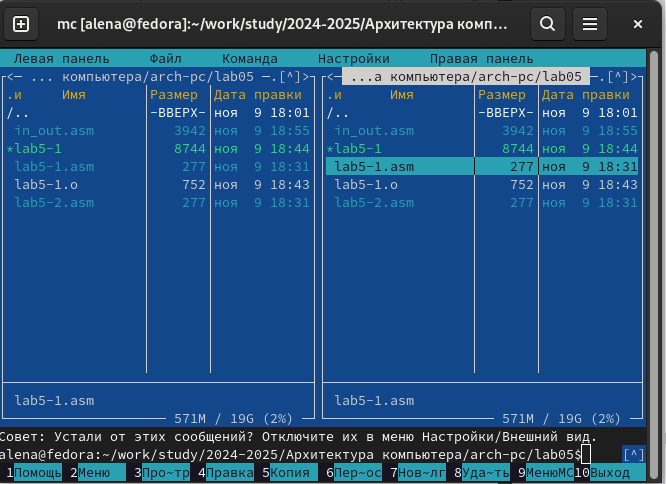


Рис. 10: Копирование файла

Исправляю текст программы в файле lab5-2.asm (рис. 11).

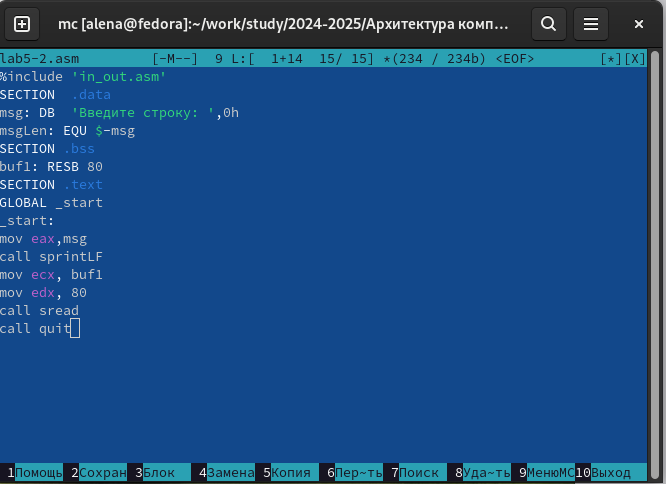


Рис. 11: Редактирование файла

Транслирую текст программы в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл. На запрос ввожу свое ФИО (рис. 12).

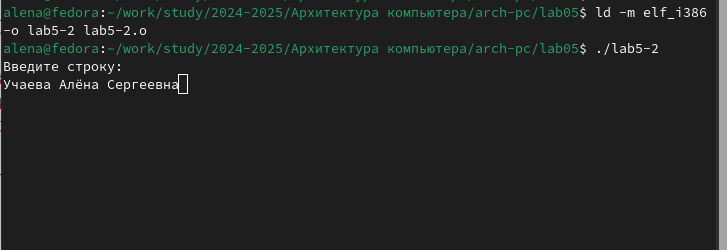


Рис. 12: Исполнение файла

В файле lab5-2.asm заменяю подпрограмму sprintLF на sprint, создаю исполняемый файл и проверяю его работу (рис. 13).

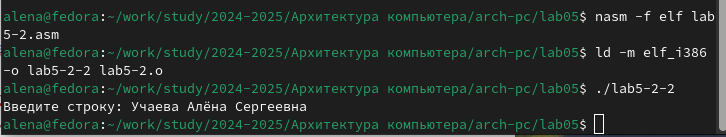


Рис. 13: Исполнение файла

Разница между первым исполняемым файлом и вторым в том, что запуск первого файла запрашивает ввод с новой строки, а второго запрашивает ввод без перехода на новую строку

## 4.4 Выполнение заданий для самостоятельной работы

Создаю копию файла lab5-1.asm и вношу изменения в программу (рис. 14).

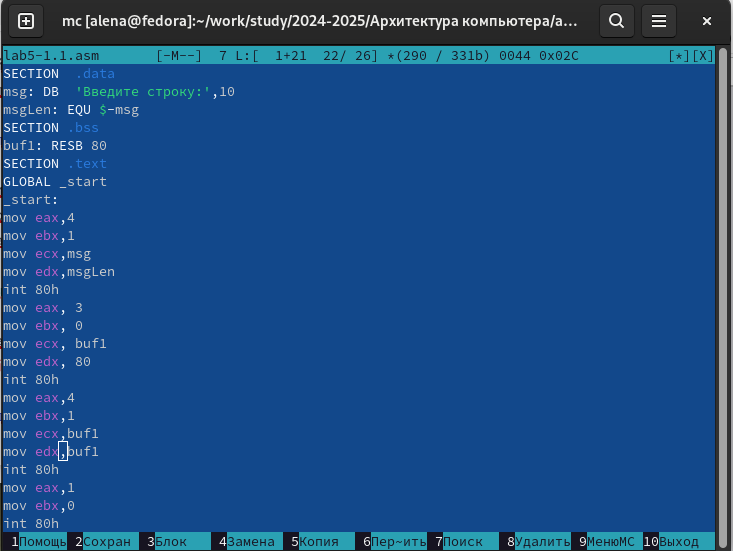


Рис. 14: Редактирование файла

Транслирую текст программы в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл. На запрос ввожу свое ФИО (рис. 15).

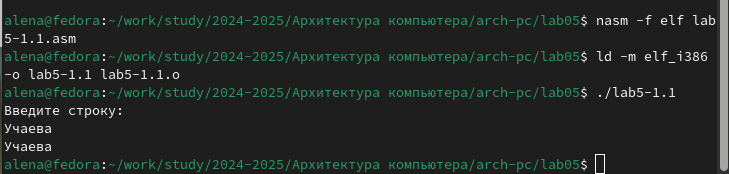


Рис. 15: Исполнение файла

Создаю копию файла lab5-2.asm, исправляю текст программы с использованием подпрограмм из внешнего файла n\_out.asm (рис. 16).

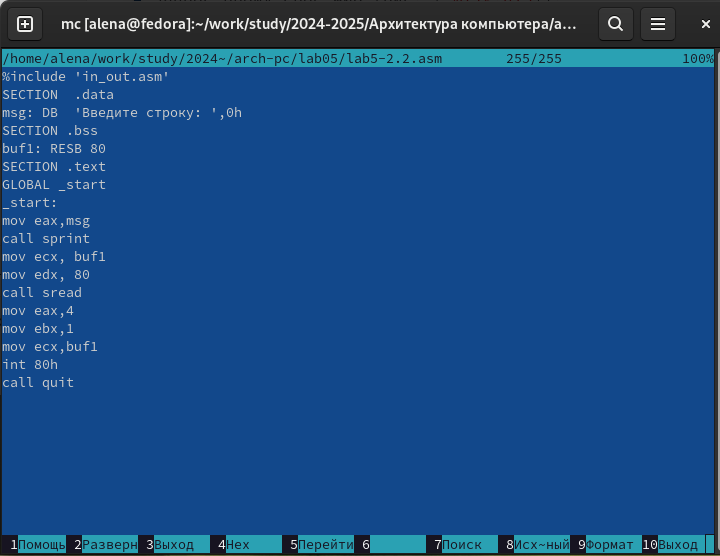


Рис. 16: Редактирование файла

Транслирую текст программы в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл. На запрос ввожу свое ФИО (рис. 17).

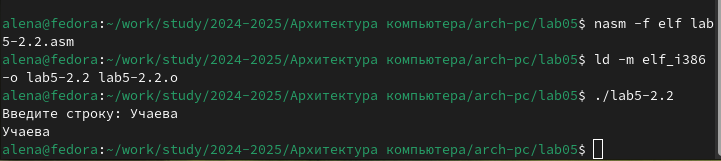


Рис. 17: Исполнение файла

# 5 Выводы

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

# Список литературы

1. [Архитектура ЭВМ](https://esystem.rudn.ru/pluginfile.php/2089085/mod_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%965.%20%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D1%8B%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D1%8B%20%D1%81%20Midnight%20Commander%20%28%29.%20%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D1%8B%20%D0%BD%D0%B0%20%D1%8F%D0%B7%D1%8B%D0%BA%D0%B5%20%D0%B0%D1%81%D1%81%D0%B5%D0%BC%D0%B1%D0%BB%D0%B5%D1%80%D0%B0%20NASM.%20%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D1%8B%D0%B5%20%D0%B2%D1%8B%D0%B7%D0%BE%D0%B2%D1%8B%20%D0%B2%20%D0%9E%D0%A1%20GNU%20Linux.pdf)