Шаблон отчёта по лабораторной работе

Простейший вариант

Дмитрий Сергеевич Кулябов

Содержание

# 1 Цель работы

Приобреcnb 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 предоставляет возможность просматривать структуру каталогов и выполнять основные операции управления файловой системой, то есть mc является файловым менеджером. Midnight Commander упрощает работу с файлами, делая её более удобной и наглядной. Программа на языке ассемблера NASM обычно состоит из трёх секций: секция кода программы (SECTION .text), секция инициализированных данных (SECTION .data), известная на этапе компиляции, и секция неинициализированных данных (SECTION .bss), для которых память отводится во время компиляции, но значения присваиваются в ходе выполнения программы. Для объявления инициализированных данных в секции .data используются директивы DB, DW, DD, DQ и DT, которые резервируют память и указывают, какие значения будут храниться в этой памяти:

* DB (define byte) — определяет переменную размером в 1 байт;
* DW (define word) — определяет переменную размером 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.

Эти директивы применяются для объявления простых переменных и массивов. Для определения строк обычно используется директива DB из-за особенностей хранения данных в оперативной памяти. Инструкция языка ассемблера mov предназначена для копирования данных из источника в приёмник.

# 4 Выполнение лабораторной работы

## 4.1 Основы работы с mc

Открываю Midnight Commander, введя в терминал mc (рис. 1).

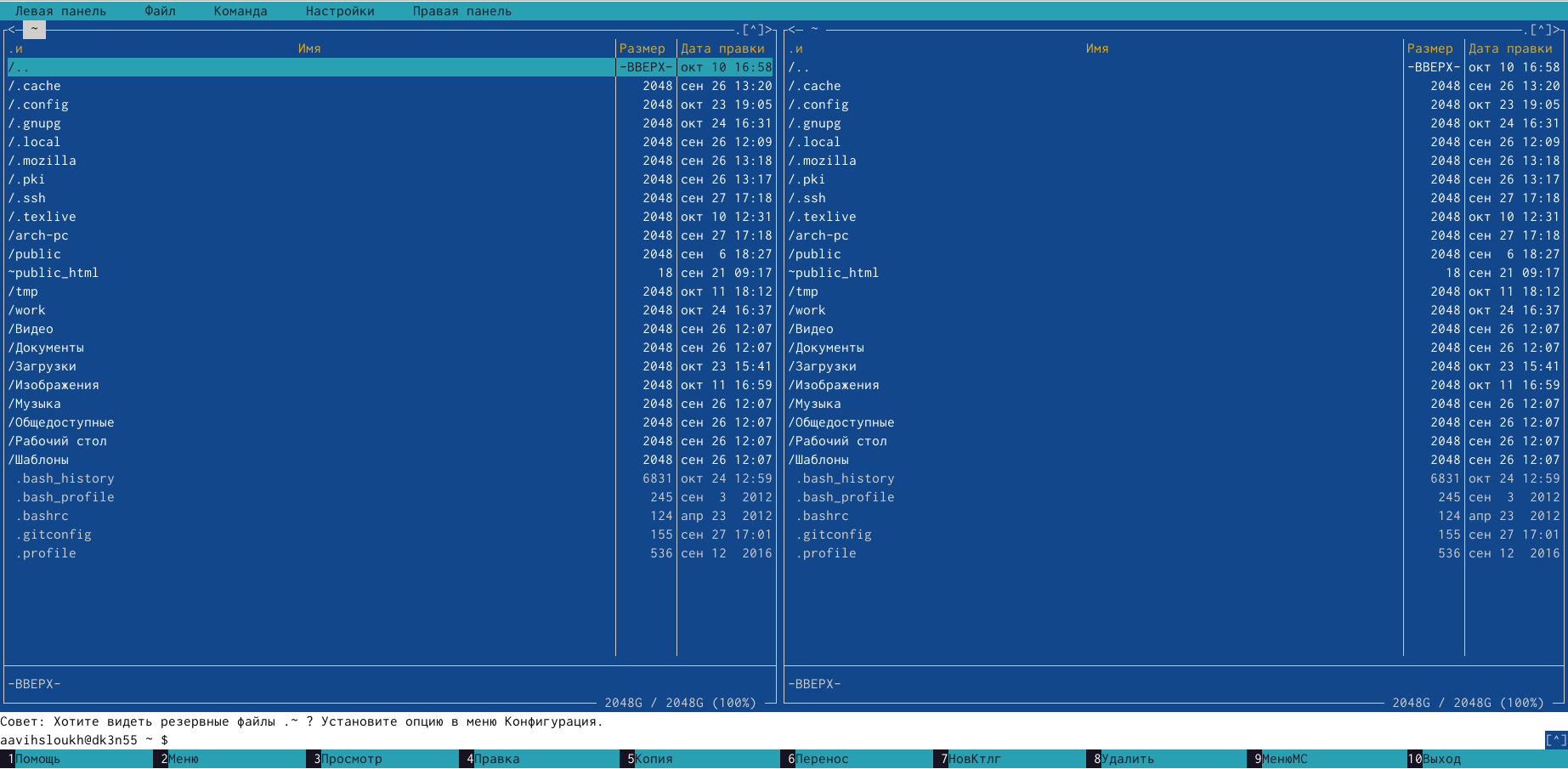


Рис. 1: Открываю Midnight comander

Перехожу в каталог ~/work/arch-pc , используя файловый менеджер mc (рис. 2)

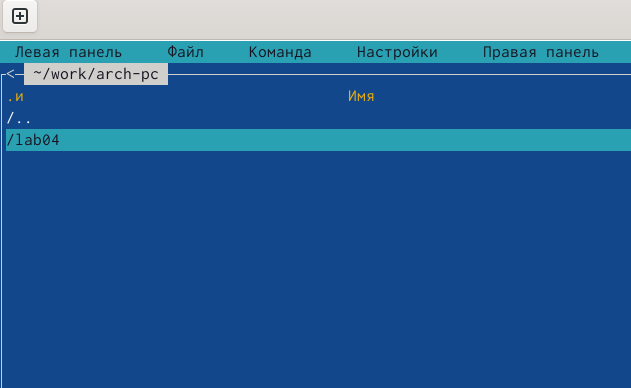


Рис. 2: Перемещение между директориями

С помощью функциональной клавиши F7 создаю каталог lab05 (рис. 3).

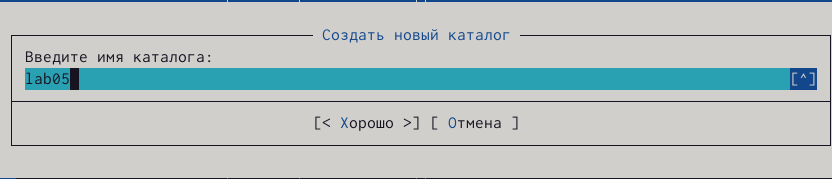


Рис. 3: Создание каталога

Вижу, что каталог создан (рис. 4).

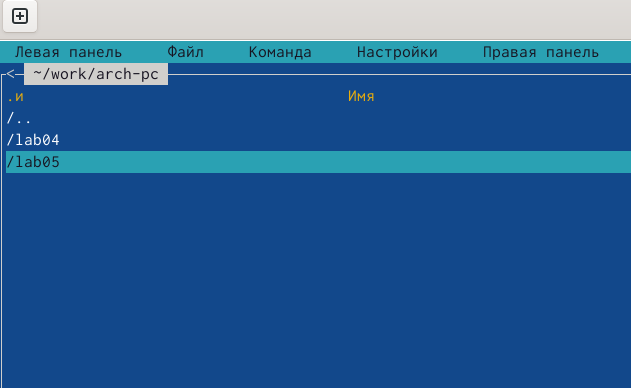


Рис. 4: Создание каталога

Захожу в каталог и в строке ввода прописываю команду touch lab5-1.asm, чтобы создать файл, в котором буду работать (рис. 5).

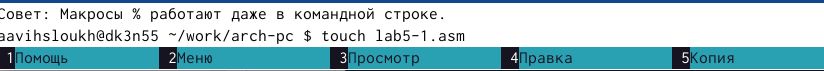


Рис. 5: Команда touch

## 4.2 Структура программы на языке ассемблера NASM

С помощью функциональной клавиши F4 открываю созданный файл в редакторе mcedit и ввожу в файл код программы (рис. 6).

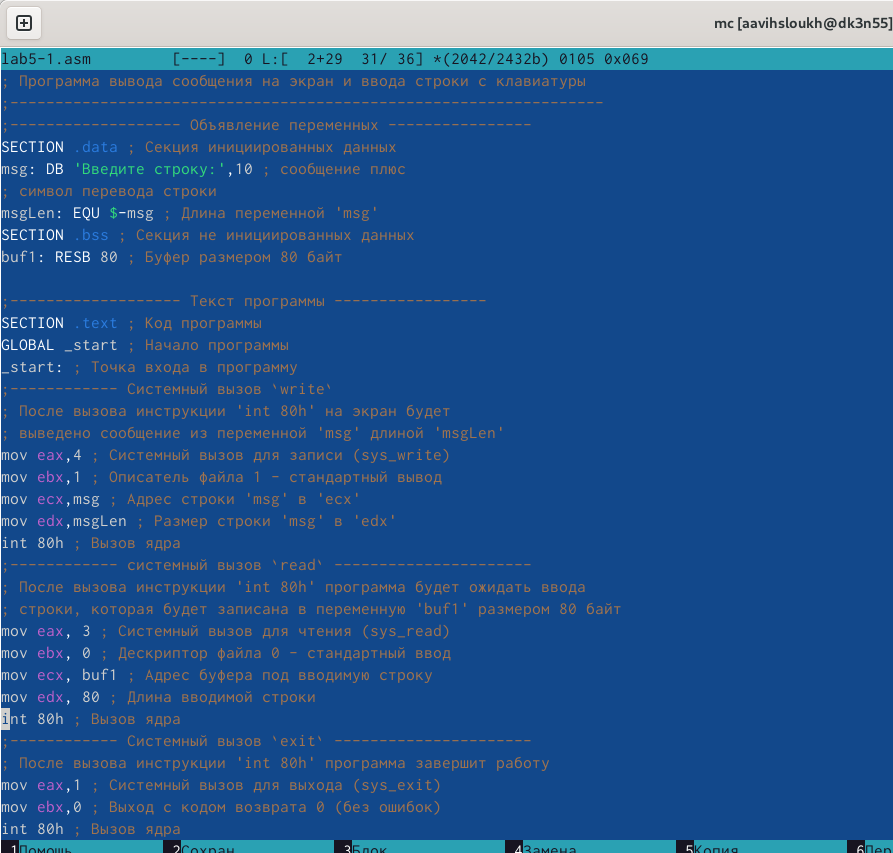


Рис. 6: Клавиша F4 и редактирование файла

С помощью функциональной клавиши F3 открываю файл для просмотра, чтобы проверить, содержит ли файл текст программы (рис. 7).

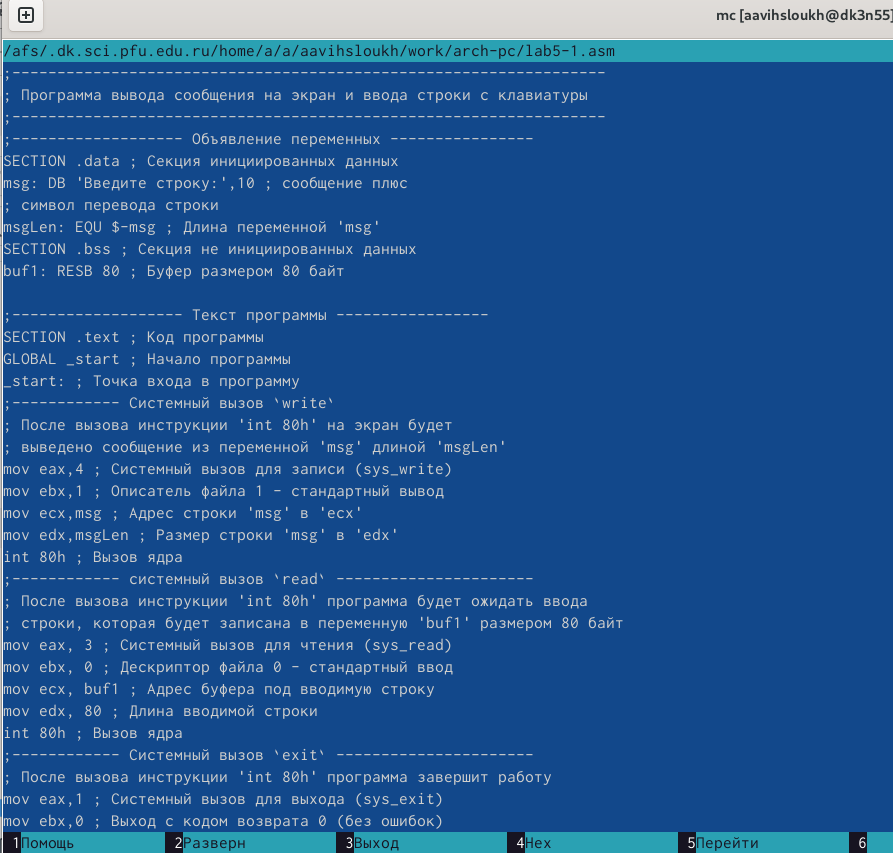


Рис. 7: Проверка

Транслирую текст программы файла в объектный файл командой nasm -f elf lab5-1.asm. Создался объектный файл lab5-1.o. Выполняю компоновку объектного файла с помощью команды ld -m elf\_i386 -o lab5-1 lab5-1.o (рис. 8). Создался исполняемый файл lab5-1.

Рис. 8: Ввод команд

Рис. 8: Ввод команд

Запускаю исполняемый файл. И ввожу свое фио (рис. 9).

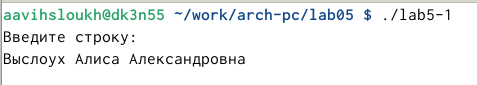


Рис. 9: Запуск

## 4.3 Подключение внешнего файла

Скачиваю файл in\_out.asm со страницы курса в ТУИС. Он сохранился в каталог “Загрузки” (рис. 10).

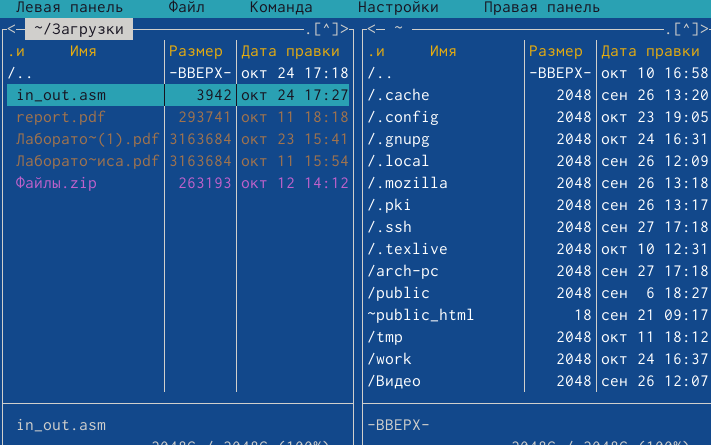


Рис. 10: Файл in\_out.asm

С помощью функциональной клавиши F5 копирую файл in\_out.asm из каталога Загрузки в созданный каталог lab05 (рис. 11).

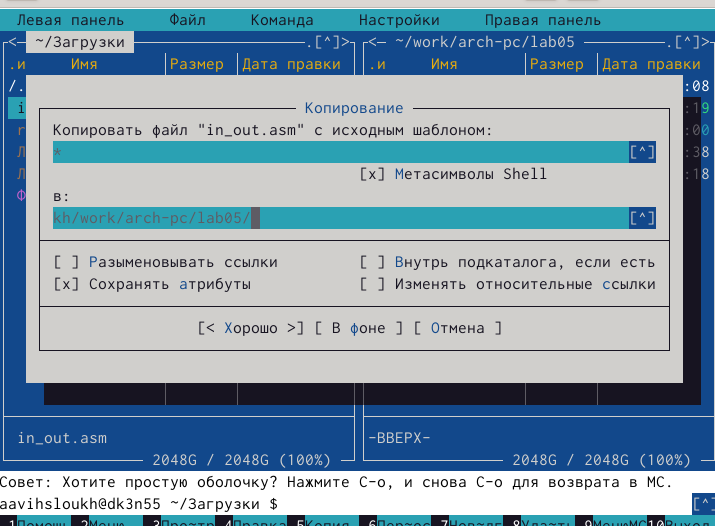


Рис. 11: Перемещение файла

С помощью функциональной клавиши F5 копирую файл lab5-1 в тот же каталог, но с именем lab5-2 (рис. 12).

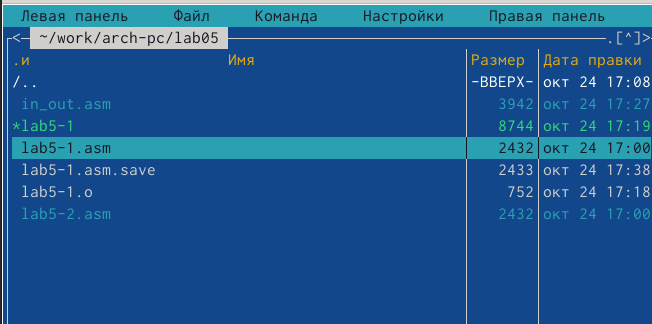


Рис. 12: Копирую файл и меняю названия

Изменяю содержимое файла lab5-2.asm во встроенном редакторе mcedit (рис. 13), чтобы в программе использовались подпрограммы из внешнего файла in\_out.asm.

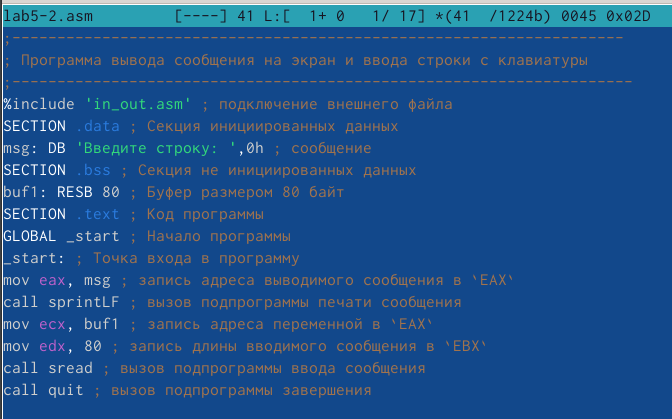


Рис. 13: Изменение содержимого файла.

Я транслирую текст программы из файла в объектный файл, используя команду nasm -f elf lab5-2.asm. В результате был создан объектный файл lab5-2.o. Затем выполняю компоновку объектного файла с помощью команды ld -m elf\_i386 -o lab5-2 lab5-2.o, в результате чего создаётся исполняемый файл lab5-2. Запускаю исполняемый файл (рис. 14).

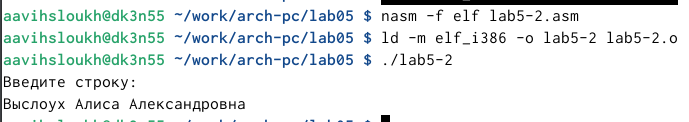


Рис. 14: Транслирование файла в объектный файл и компановка.

Открываю файл lab6-2.asm для редактирования с помощью текстового редактора nano, нажав функциональную клавишу F4. Вношу изменения в подпрограмму sprintLF, переименовывая её в sprint. После этого сохраняю изменения и открываю файл для просмотра, чтобы убедиться, что все изменения были сохранены корректно (рис. 15).

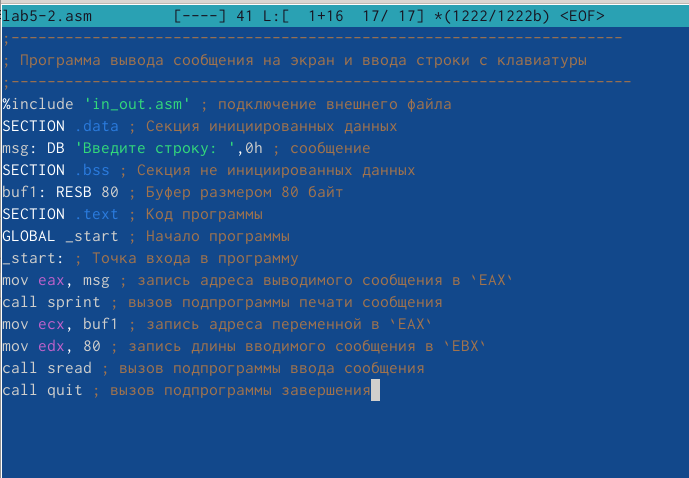


Рис. 15: Изменение файла

Снова транслирую файл, выполняю компоновку созданного объектного файла, запускаю новый исполняемый файл (рис. 16).

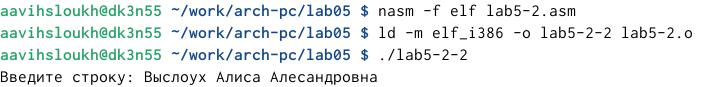


Рис. 16: Транслирование файла в объектный файл и компановка.

Разница между первым исполняемым файлом lab5-2 и вторым lab5-2-2 заключается в том, что при запуске первого файла программа запрашивает ввод с новой строки, в то время как программа во втором файле запрашивает ввод без переноса на новую строку. Это различие обусловлено изменениями в подпрограммах sprintLF и sprint: первая добавляет перенос строки после запроса ввода, а вторая — нет.

# 5 Выполнение заданий для самостоятельной работы

Создаю копию файла lab6-1.asm с именем lab6-1-1.asm с помощью функциональной клавиши F5 (рис. 17)

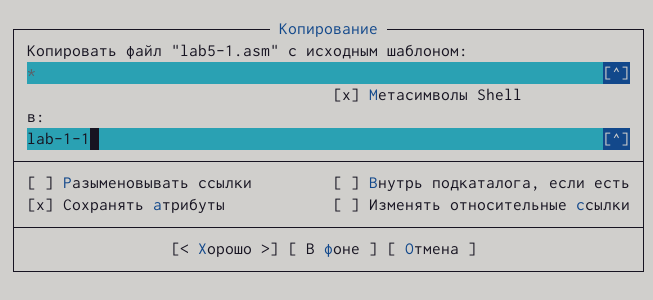


Рис. 17: Копирование файла.

С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. 18).

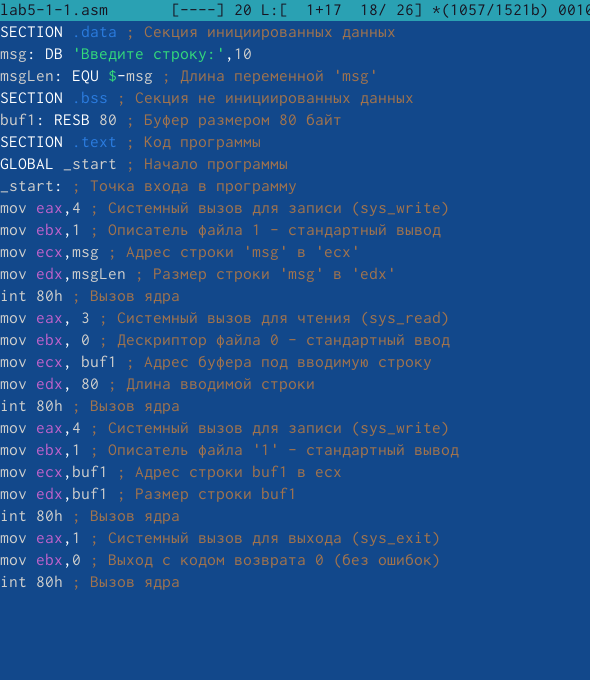


Рис. 18: Изменение файла.

Создаю объектный файл lab5-1-1.o и передаю его на обработку компоновщику. После компоновки получаю исполняемый файл lab5-1-1. Запускаю полученный исполняемый файл. Программа запрашивает ввод, куда я ввожу свои ФИО, после чего программа выводит введенные мною данные (рис. 19).

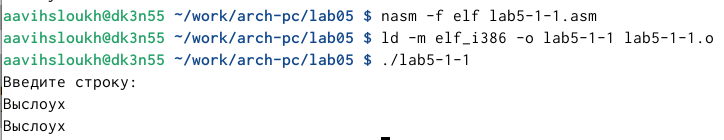


Рис. 19: Проверка.

Создаю копию файла lab5-2.asm с именем lab6-2-1.asm с помощью функциональной клавиши F5, открываю файл с помощью клавиши F4 и изменяю ее так, чтобы кроме вывода приглашения и запроса ввода, программа выводила пользовательскую строку (рис. 20).

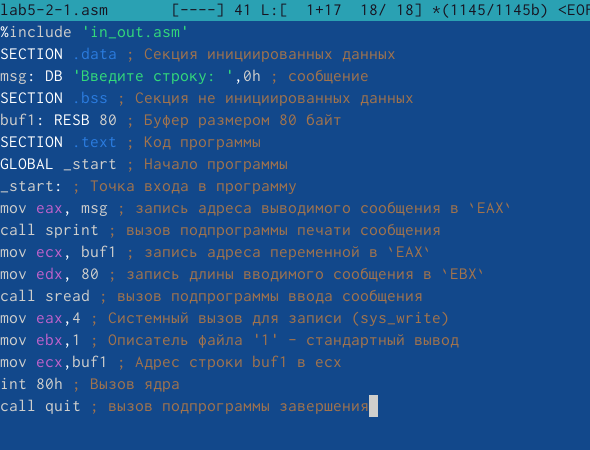


Рис. 20: Изменение файла.

Проверяю аналогично предыдущему (рис. 21).

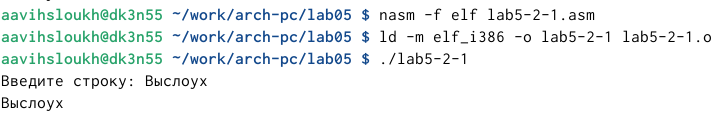


Рис. 21: Проверка.

# 6 Выводы

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