Отчёт по лабораторной работе №5

Дисциплина: архитектура компьютера

Черкашина Ангелина Максимовна

Содержание

# 1 Цель работы

Целью данной 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 работ

# 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 (учетве- рённое слово); - DT (define ten bytes) — определяет переменную размером в 10 байт. Директивы используются для объявления простых переменных и для объявления массивов. Для определения строк принято использовать директиву DB в связи с особенностями хранения данных в оперативной памяти. Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике.

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.

int n

Здесь n — номер прерывания, принадлежащий диапазону 0–255. При программировании в Linux с использованием вызовов ядра sys\_calls n=80h (принято задавать в шестнадцатеричной системе счисления).

# 4 Выполнение лабораторной работы

## 4.1 Основы работы с mc

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

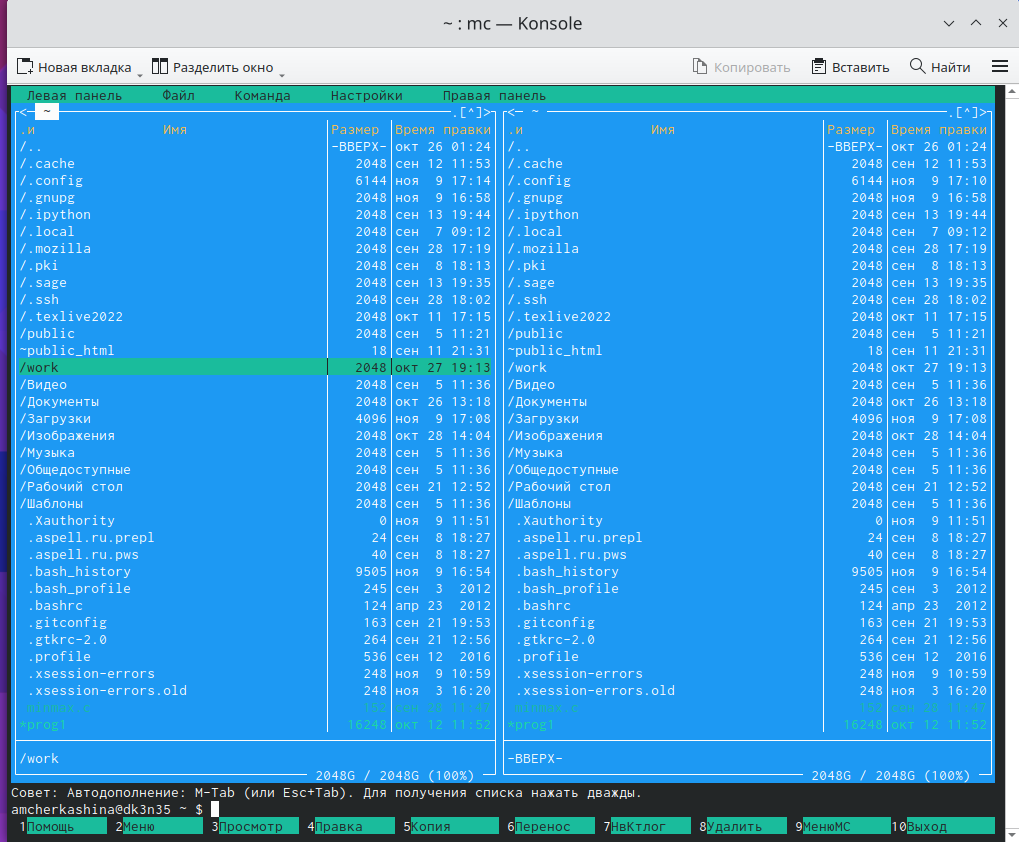


Рис. 1: Открытый Midnight Commander

Используя файловый менеджер mc, перехожу в каталог ~/work/arch-pc, созданный при выполнении лабораторной работы №4 (рис. 2).

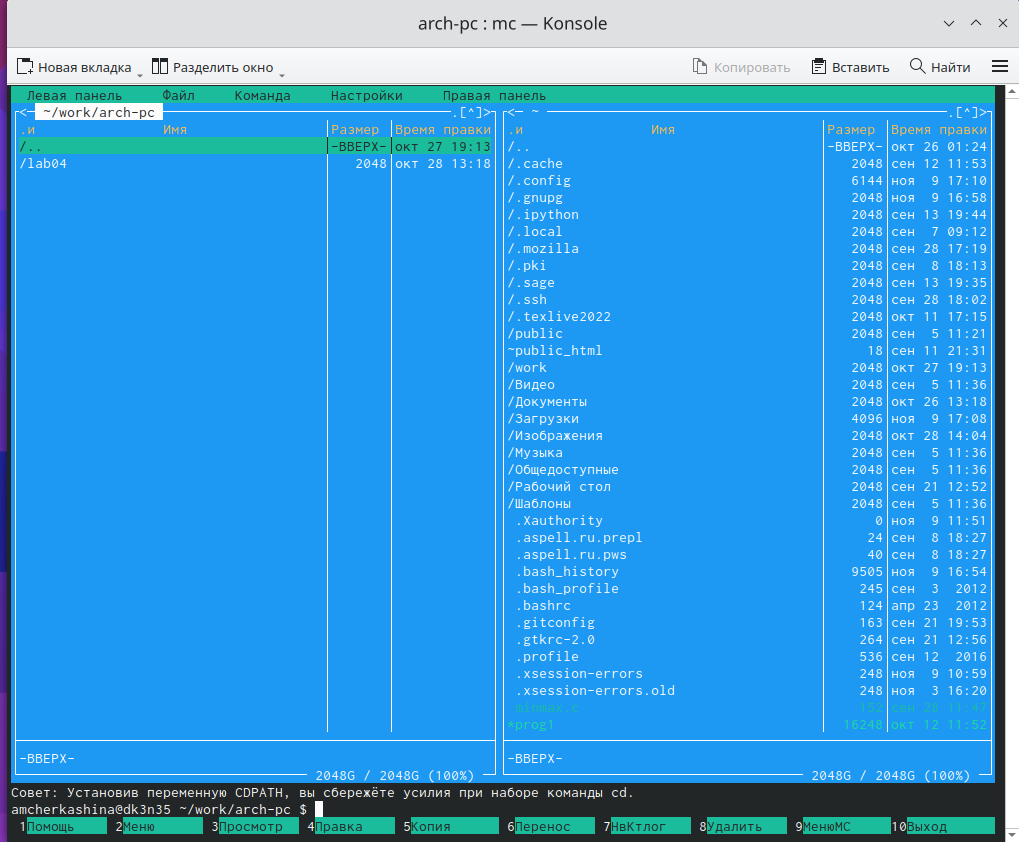


Рис. 2: Перемещение между директориями

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

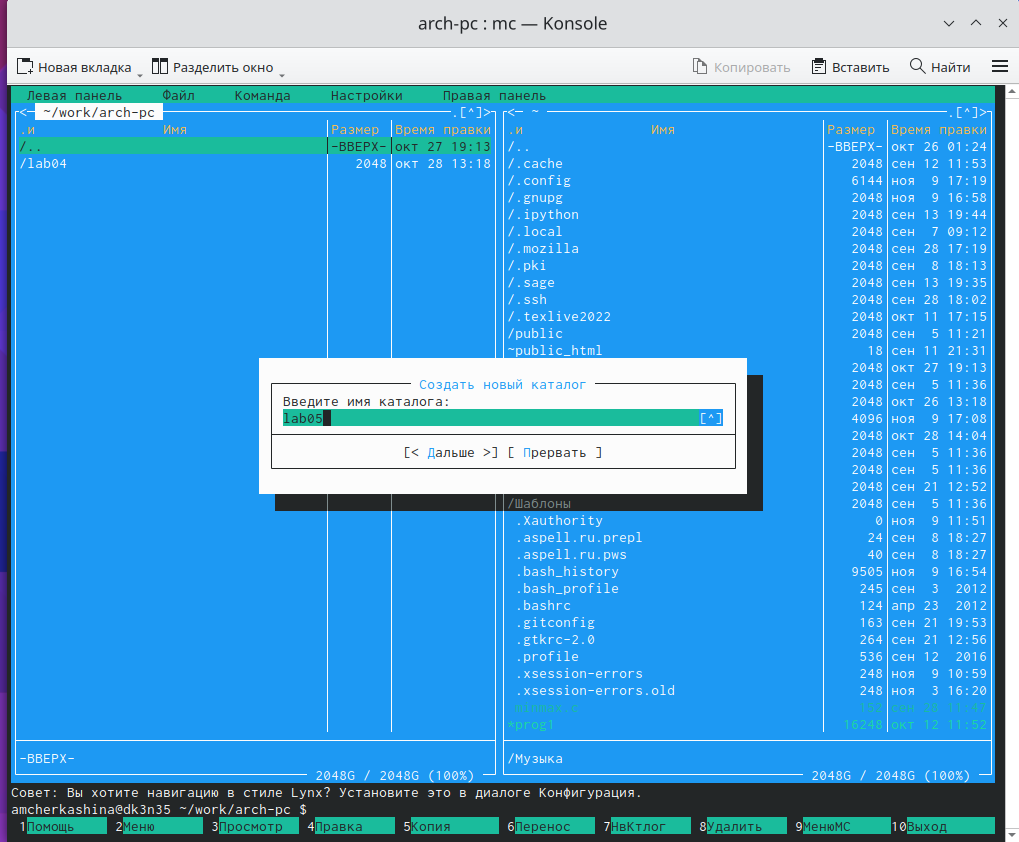


Рис. 3: Создание каталога

Перехожу в созданный каталог (рис. 4).

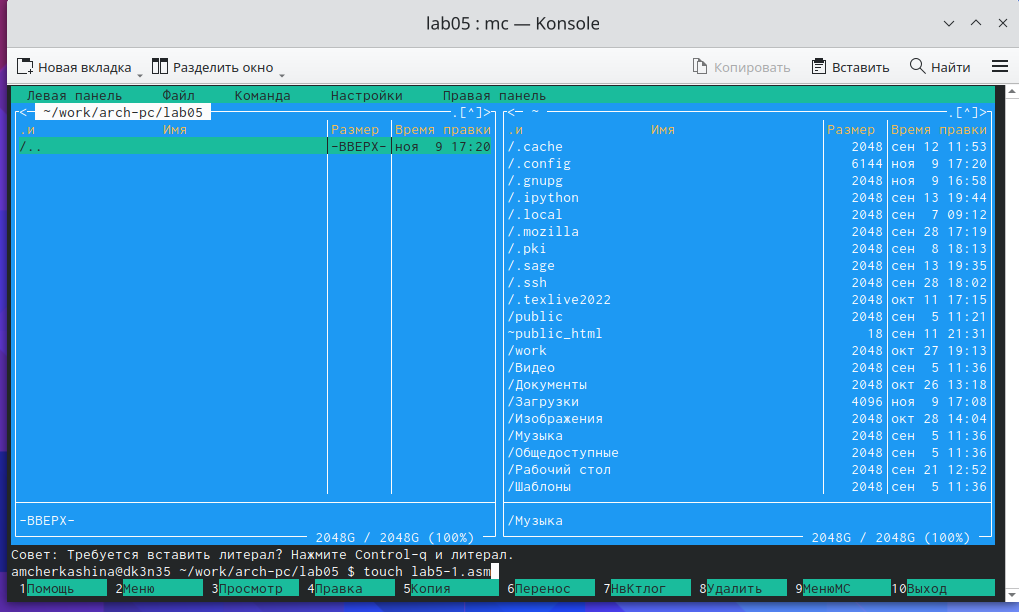


Рис. 4: Перемещение между директориями

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

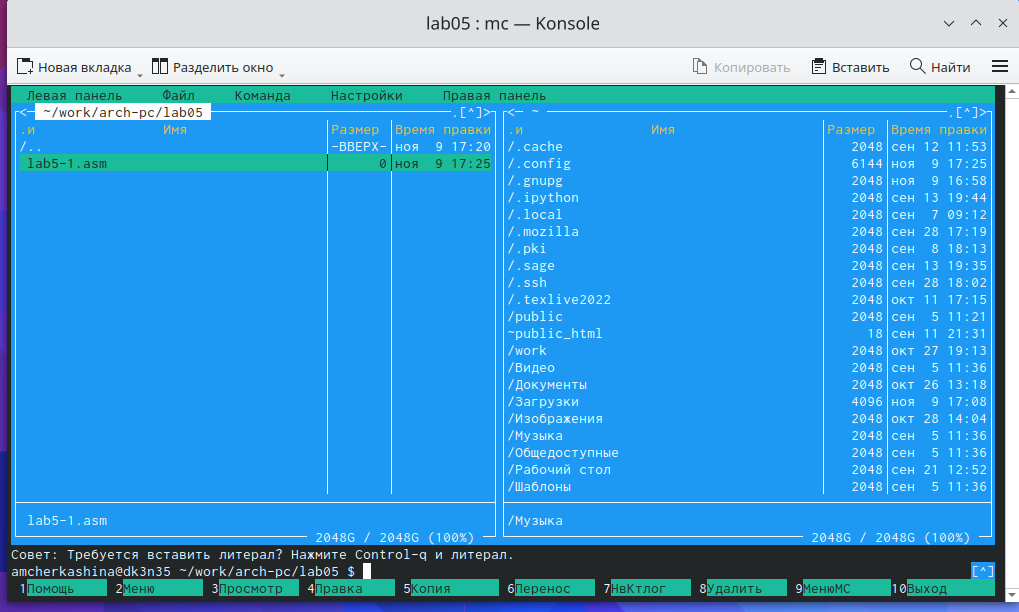


Рис. 5: Создание файла

## 4.2 Структура программы на языке ассемблера NASM

С помощью функциональной клавиши F4 открываю созданный файл для редактирования во встроенном редакторе mcedit (рис. 6).



Рис. 6: Открытие файла lab5-1.asm для редактирования

Ввожу в файл код программы вывода сообщения на экран и ввода строки с клавиатуры (рис. 7). Сохраняю изменения и закрываю файл.

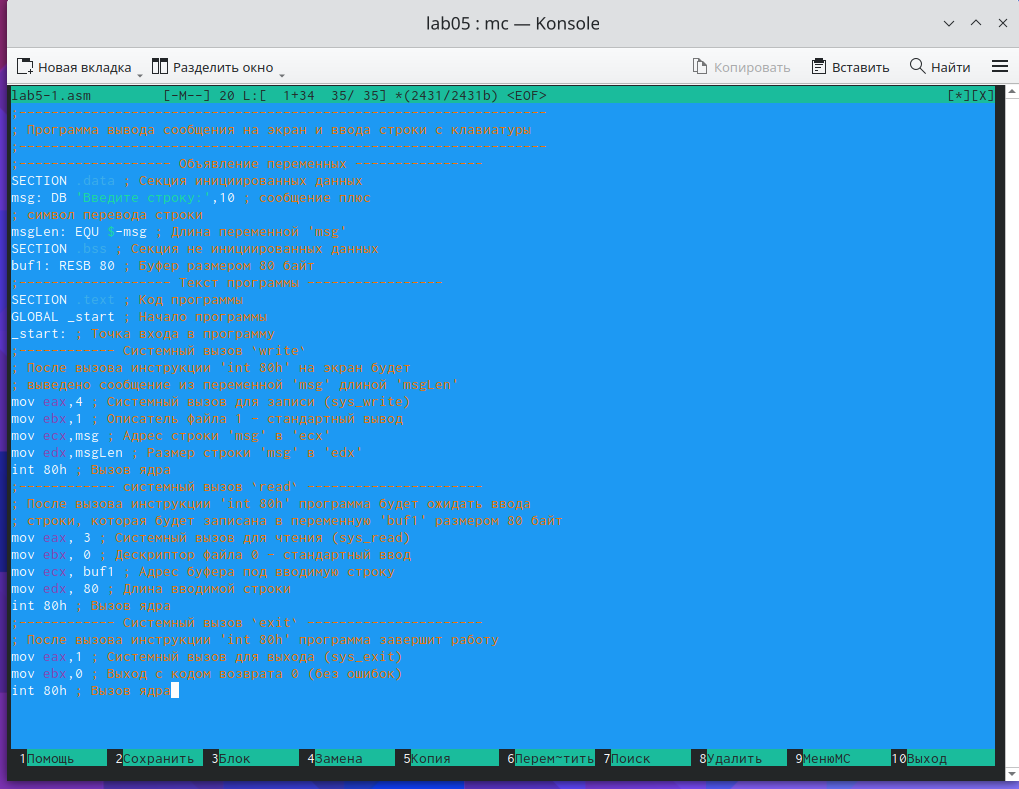


Рис. 7: Редактирование файла lab5-1.asm

С помощью функциональной клавиши F3 открываю файл lab5-1.asm для просмотра, чтобы убедииться, что файл содержит текст программы (рис. 8).

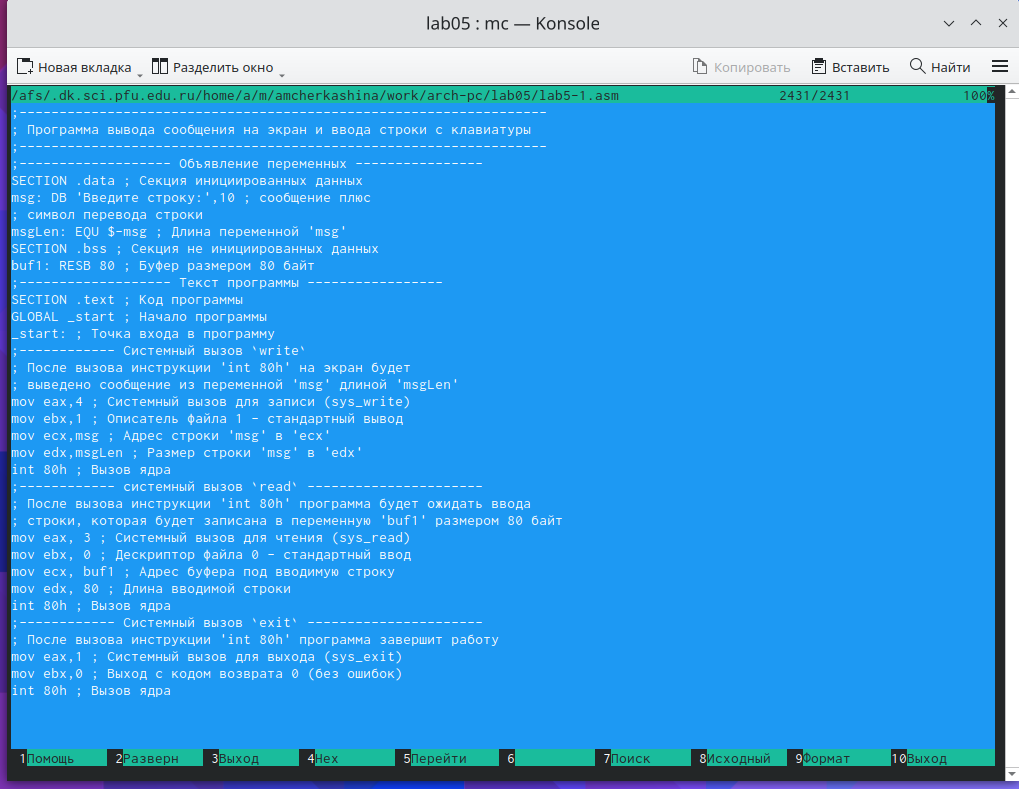


Рис. 8: Открытие файла lab5-1.asm для просмотра

Транслирую текст программы файла lab5-1.asm 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. Создался исполняемый файл lab5-1 (рис. 9).

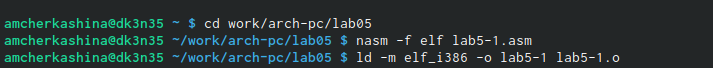


Рис. 9: Компиляция файла и передача на обработку компановщику

Запускаю получившийся исполняемый файл. Программа выводит строку “Введите строку:” и ожидает ввода с клавиатуры. На запрос я ввожу свои ФИО, на этом программа заканчивает свою работу (рис. 10).

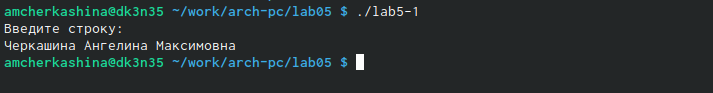


Рис. 10: Исполнение файла

## 4.3 Подключение внешнего файла

Скачиваю файл in\_out.asm со страницы курса в ТУИС. Он сохранился в каталог Загрузки (рис. 11).

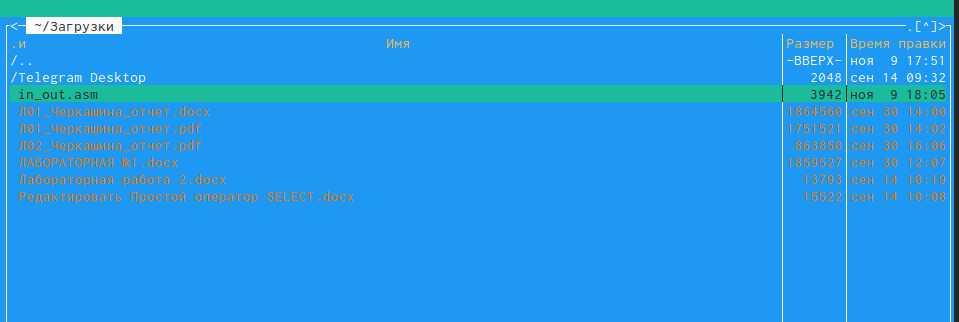


Рис. 11: Скачанный файл in\_out.asm

С помощью функциональной клавиши F5 копирую файл in\_out.asm из каталога Загрузки в созданный каталог lab05 (рис. 12).

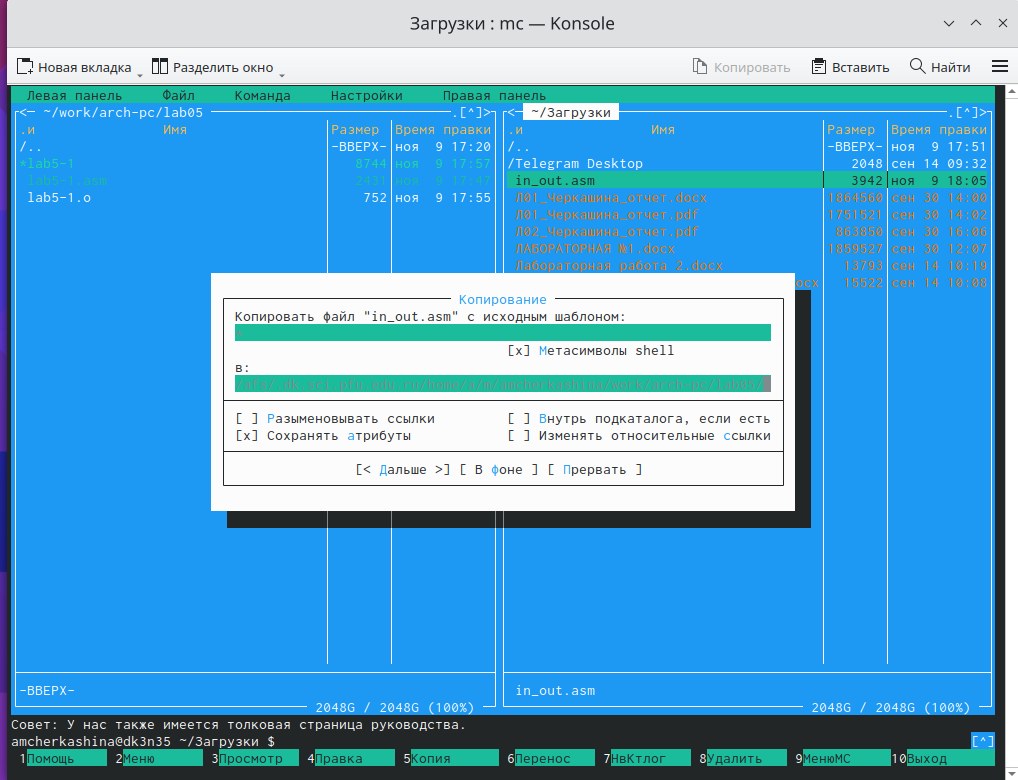


Рис. 12: Копирование файла в нужный каталог

С помощью функциональной клавиши F5 создаю копию файла lab5-1 в том же каталоге, но с другим именем (lab5-2.asm). Для этого в появившемся окне mc прописываю путь к копии файла с новым именем (рис. 13).

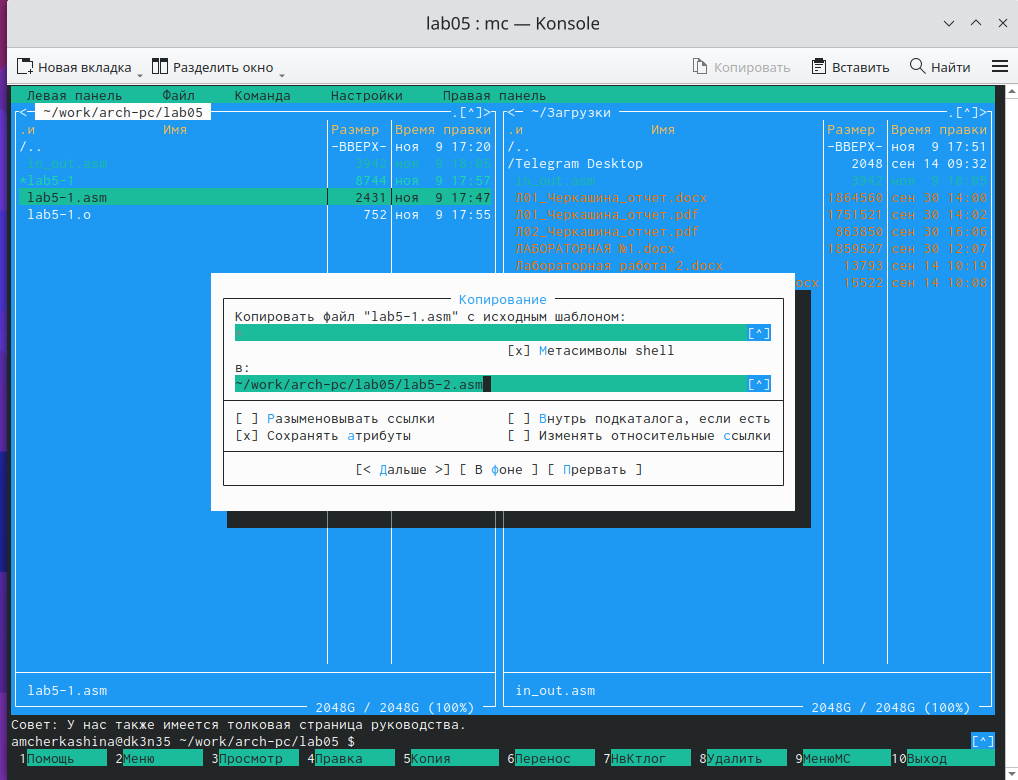


Рис. 13: Копирование файла с другим именем

Изменяю содержимое файла lab5-2.asm во встроенном редакторе mcedit так, чтобы в программе использовались подпрограммы из внешнего файла in\_out.asm (рис. 14).

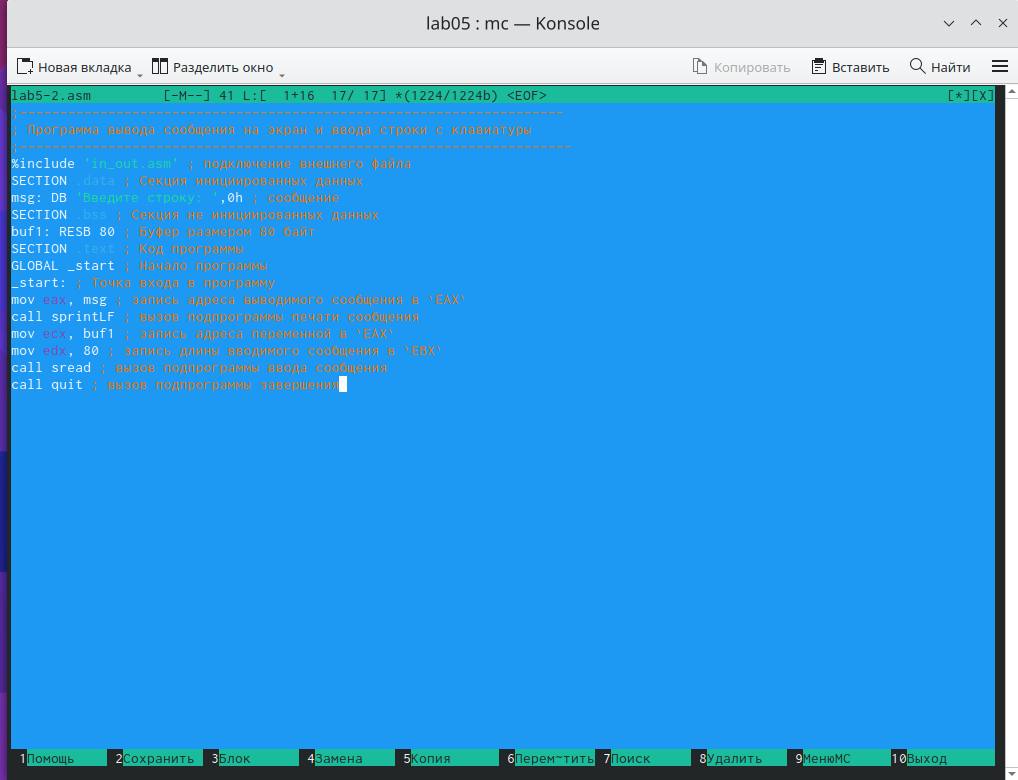


Рис. 14: Редактирование файла lab5-2.asm

Транслирую текст программы файла lab5-2.asm в объектный файл командой nasm -f elf lab5-2.asm. Создался объектный файл lab5-2.o. Выполняю компоновку объектного файла с помощью команды ld -m elf\_i386 -o lab5-2 lab5-2.o Создался исполняемый файл lab5-2. Запускаю получившийся исполняемый файл (рис. 15).

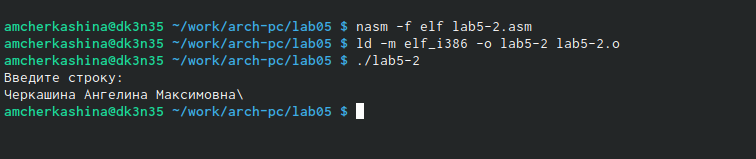


Рис. 15: Компиляция, компоновка и исполнение файла

Открываю файл lab5-2.asm для редактирования в mcedit функциональной клавишей F4. Изменяю в нем подпрограмму sprintLF на sprint (рис. 16).

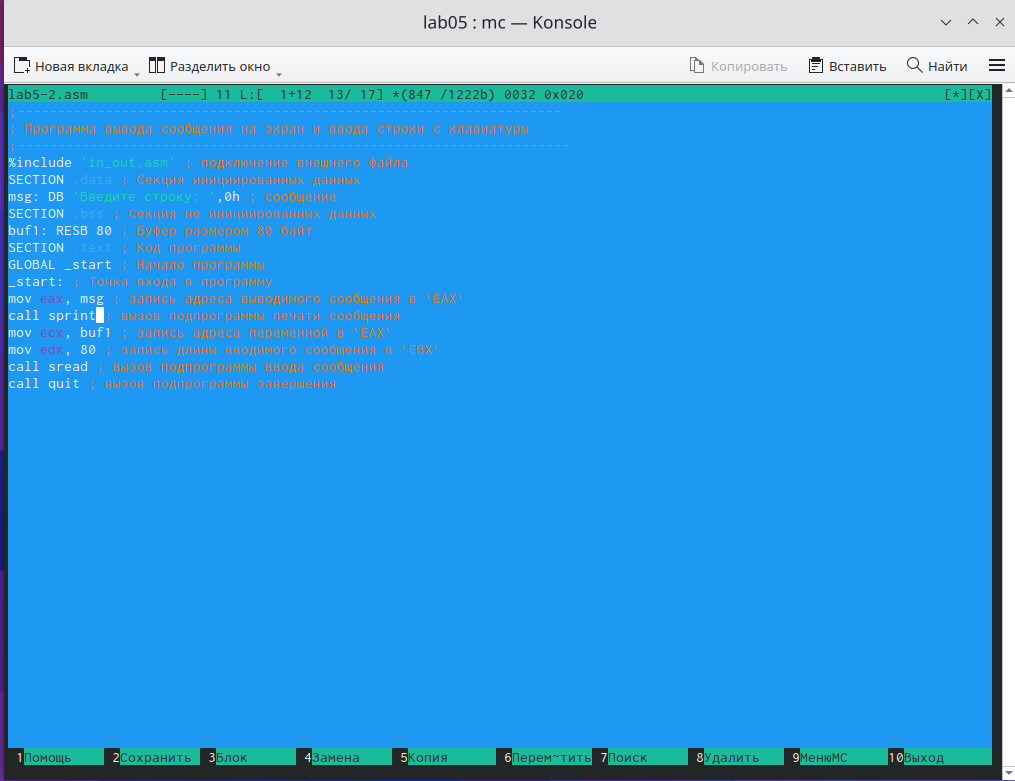


Рис. 16: Изменение подпрограммы программы файла lab5-2.asm

Сохраняю изменения и открываю файл для просмотра, чтобы проверить корректность сохранения моих действий (рис. 17).

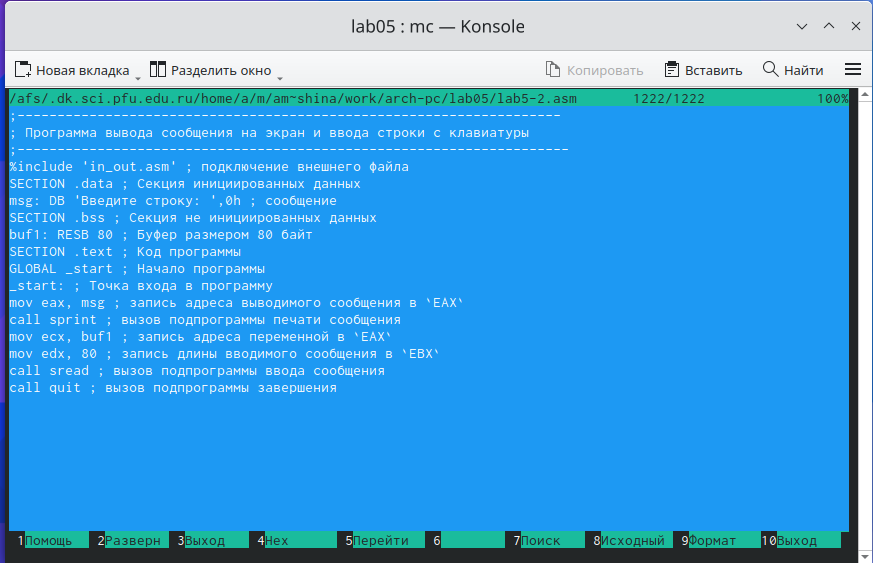


Рис. 17: Просмотр отредактированного файла lab5-2.asm

Снова транслирую текст программы файла в объектный файл, выполняю компоновку созданного объектного файла, получаю исполняемый файл и запускаю его (рис. 18).

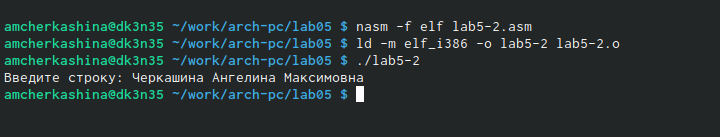


Рис. 18: Исполнение отредактированного файла

Разница между первым и вторым исполняемыми файлами в том, что запуск первого запрашивает ввод с новой строки, а программа, которая исполняется при запуске второго, запрашивает ввод без переноса на новую строку. В этом заключается различие между подпрограммами sprintLF и sprint.

## 4.4 Выполнение заданий для самостоятельной работ

1. Создаю копию файла lab5-1.asm с именем lab5-1-1.asm с помощью функциональной клавиши F5 (рис. 19).

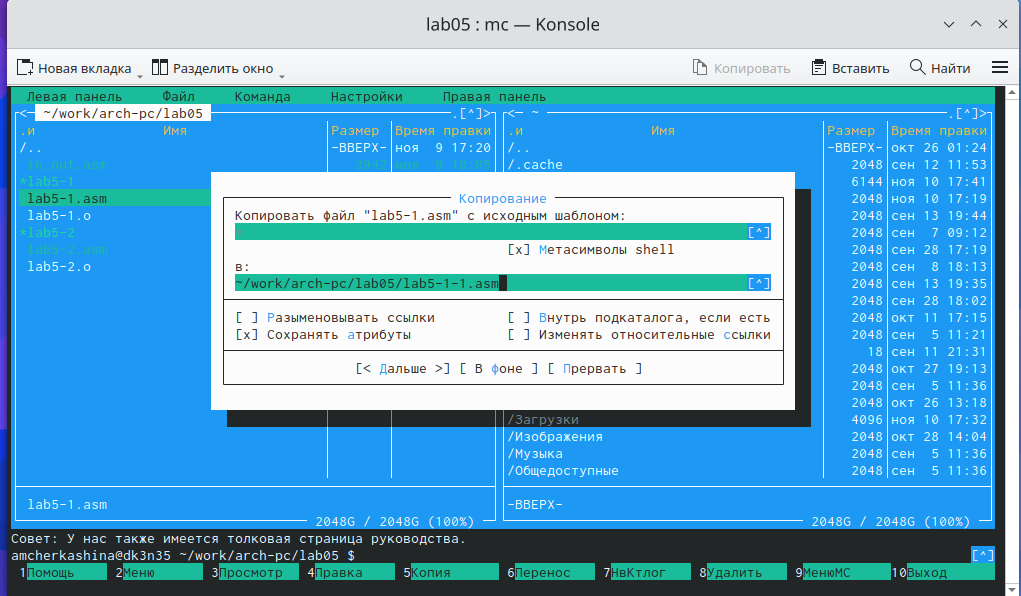


Рис. 19: Копирование файла с новым именем

С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу (без использования внешнего файла in\_out.asm) так, чтобы кроме вывода приглашения типа “Введите строку:” и запроса ввода строки с клавиатуры она выводила на экран вводимую пользователем строку (рис. 20).

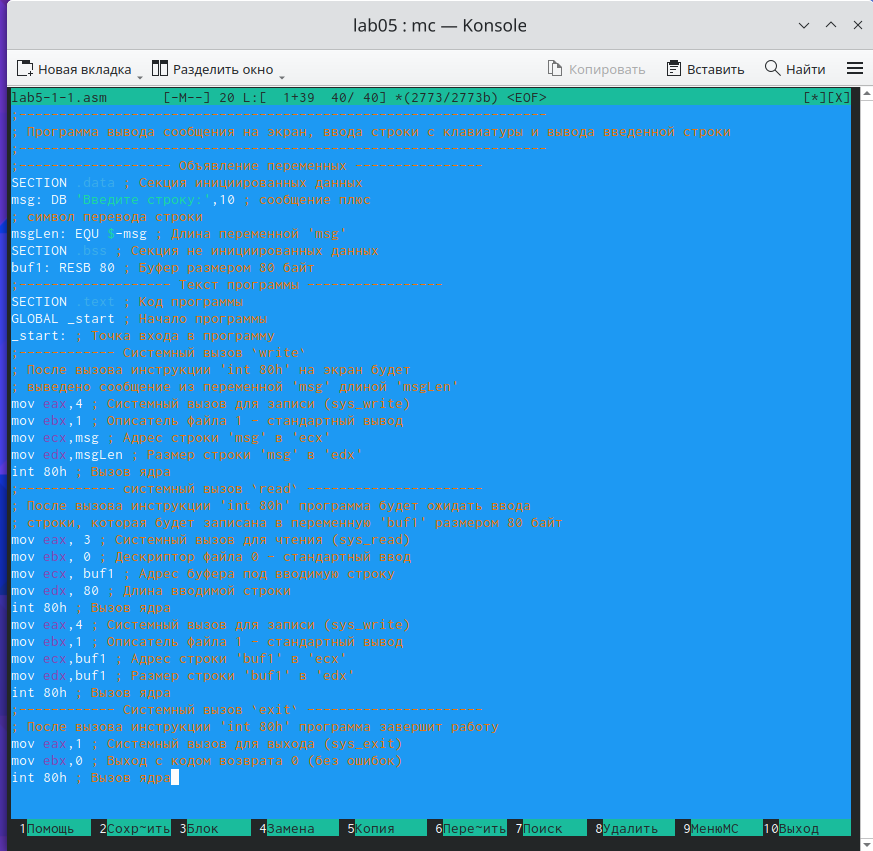


Рис. 20: Редактирование файла lab5-1-1.asm

Открываю файл lab5-1-1.asm для просмотра, чтобы убедиться в корректности внесенных мной изменений (рис. 21).

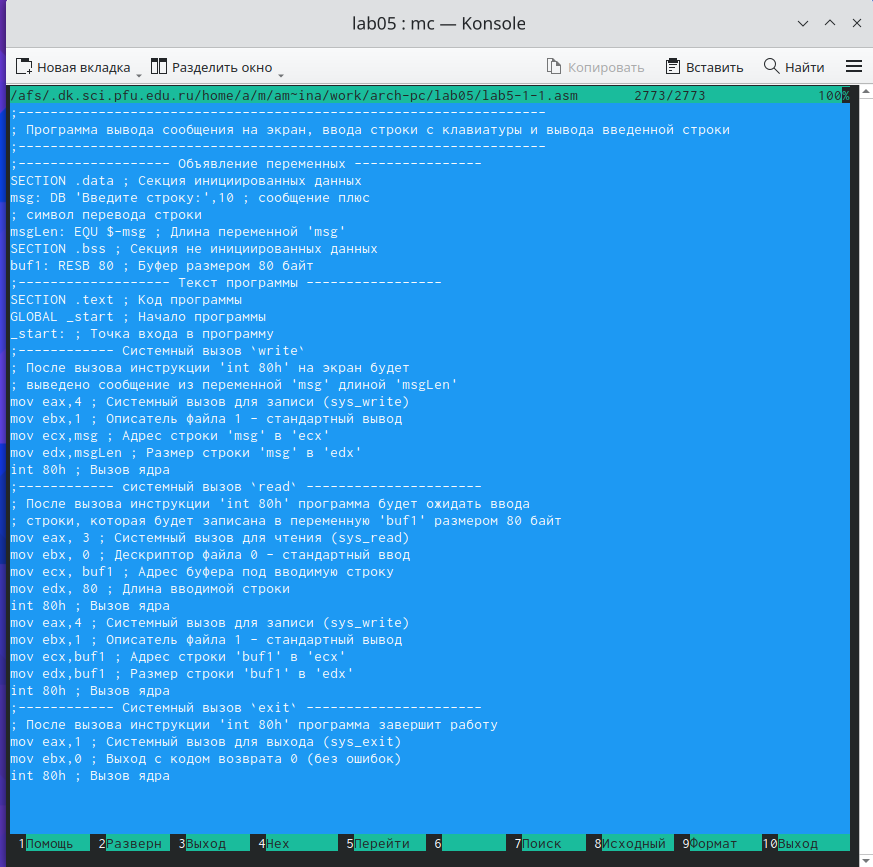


Рис. 21: Просмотр отредактированного файла lab5-1-1.asm

1. Создаю объектный файл lab5-1-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-1-1, запускаю полученный исполняемый файл. Программа запрашивает ввод с новой строки, ввожу свои ФИО. Далее программа выводит введенные мной данные на экран (рис. 22).

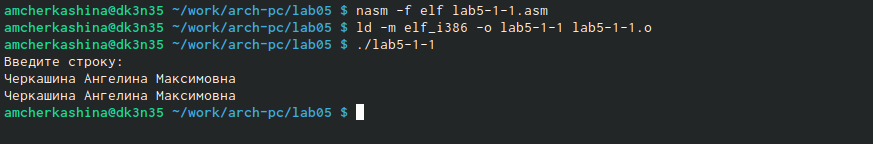


Рис. 22: Исполнение файла lab5-1-1

1. Создаю копию файла lab5-2.asm с именем lab5-2-1.asm с помощью функциональной клавиши F5 (рис. 23).

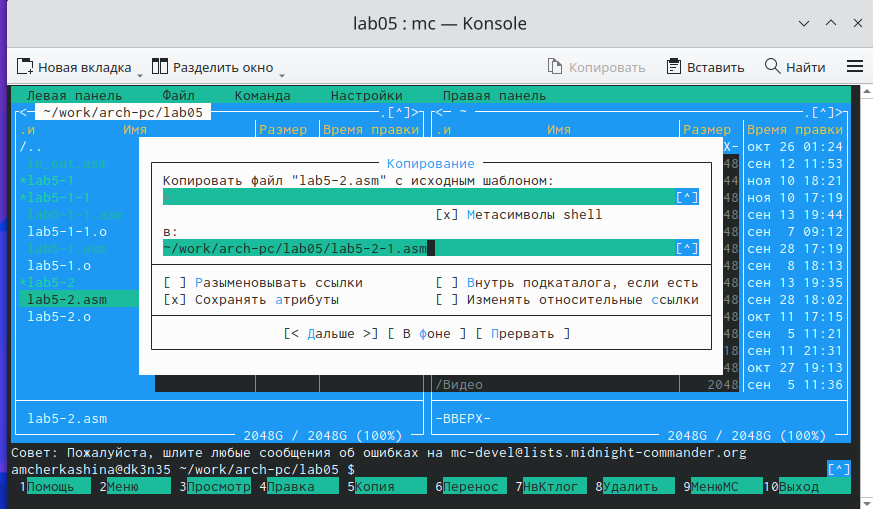


Рис. 23: Копирование файла с новым именем

С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу, используя подпрограммы из внешнего файла in\_out.asm так, чтобы кроме вывода приглашения типа “Введите строку:” и запроса ввода строки с клавиатуры она выводила на экран вводимую пользователем строку (рис. 24).

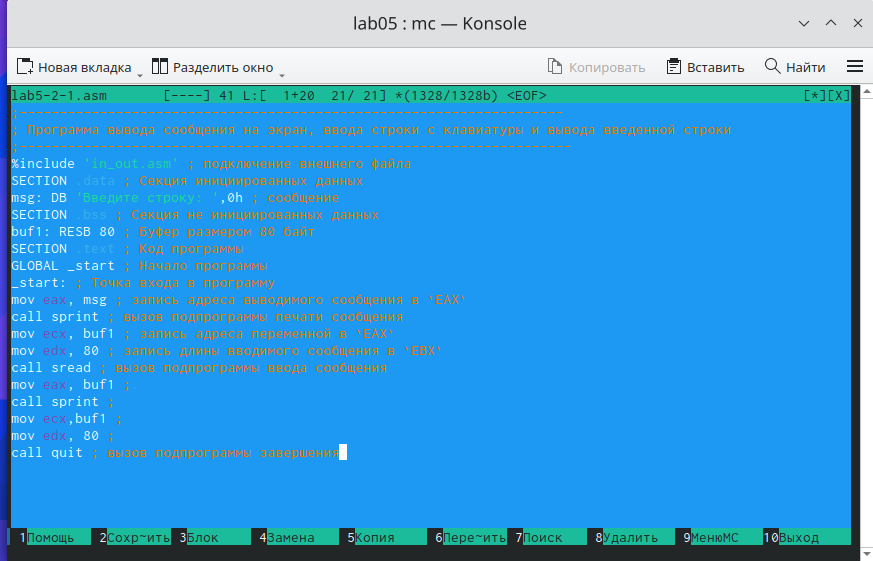


Рис. 24: Редактирование файла lab5-2-1.asm

Открываю файл lab5-2-1.asm для просмотра, чтобы убедиться в корректности внесенных мной изменений (рис. 25).

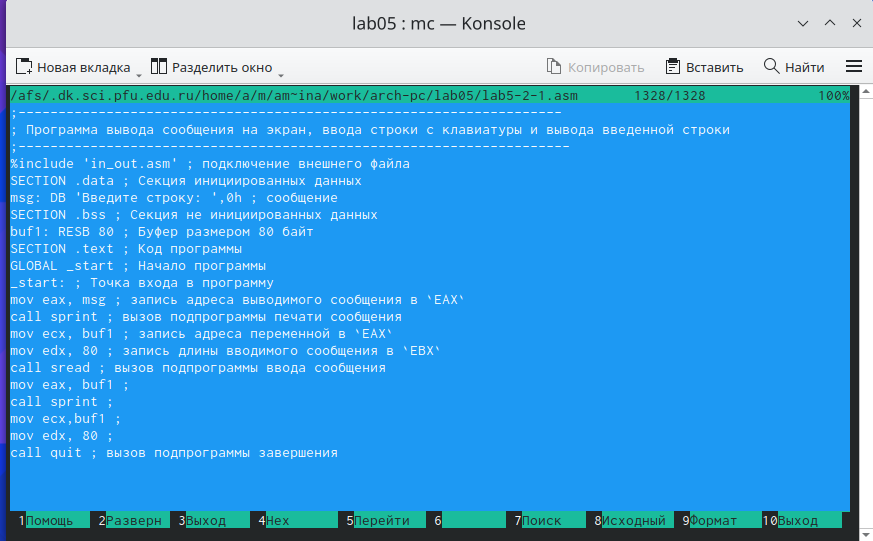


Рис. 25: Просмотр отредактированного файла lab5-2-1.asm

1. Создаю объектный файл lab5-2-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-2-1, запускаю полученный исполняемый файл. Программа запрашивает ввод без переноса на новую строку, ввожу свои ФИО. Далее программа выводит введенные мной данные на экран (рис. 26).

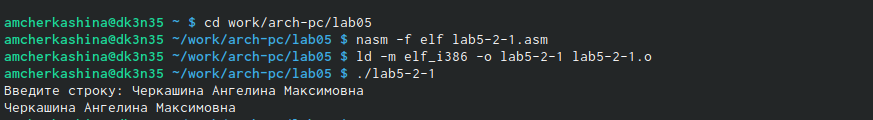


Рис. 26: Исполнение файла lab5-2-1

# 5 Выводы

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

# 6 Список литературы

1. Архитектура ЭВМ