Отчёт по лабораторной работе

Дисциплина: архитектура компьютера

Шакиров Индус Равилевич

Содержание

# 1 Цель работы

Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander, освоение инструкций языка ассемблера mov n 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).

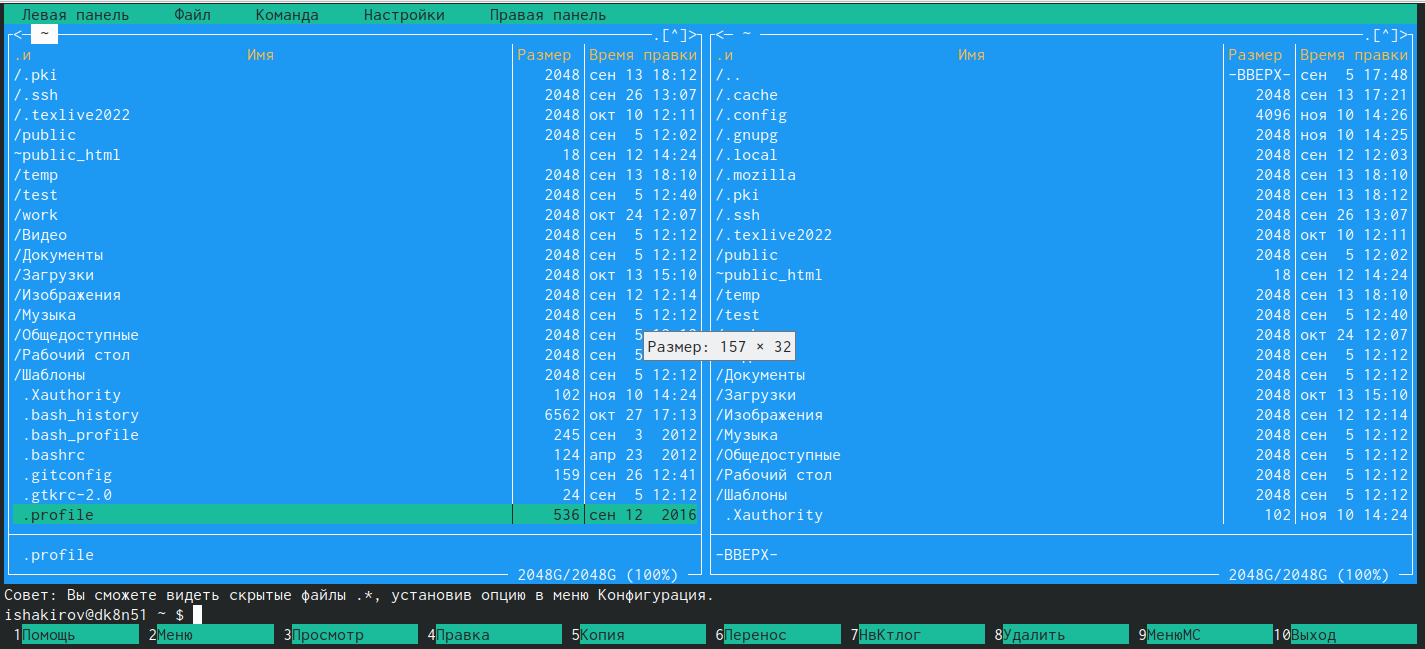
Для объявления инициированных данных в секции .data используются директивы DB, DW, DD, DQ и DT, которые резервируют память и указывают, какие значения должны храниться в этой памяти: • DB (define byte) — определяет переменную размером в 1 байт; • DW (define word) — определяет переменную размеров в 2 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.

Директивы используются для объявления простых переменных и для объявления масси- вов. Для определения строк принято использовать директиву DB в связи с особенностями хранения данных в оперативной памяти.

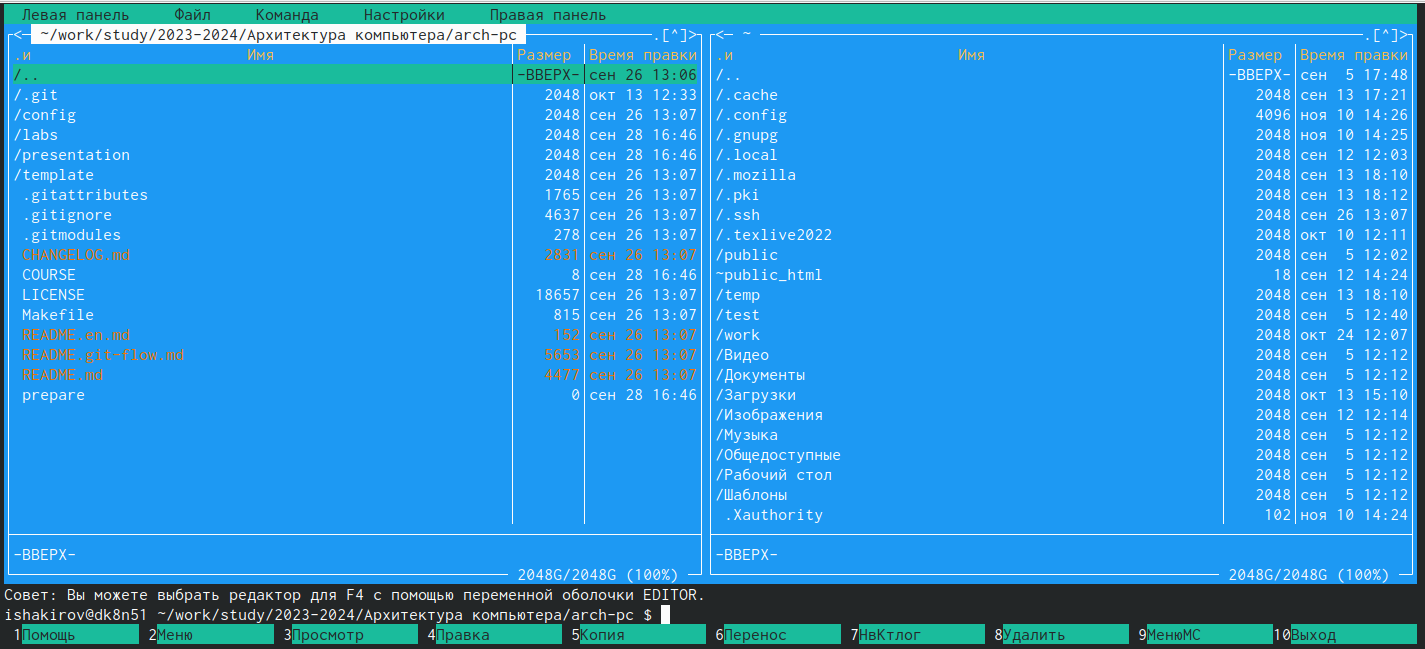
Для объявления неинициированных данных в секции .bss используются директивы resb, resw, resd и другие, которые сообщают ассемблеру, что необходимо зарезервировать за- данное количество ячеек памяти.

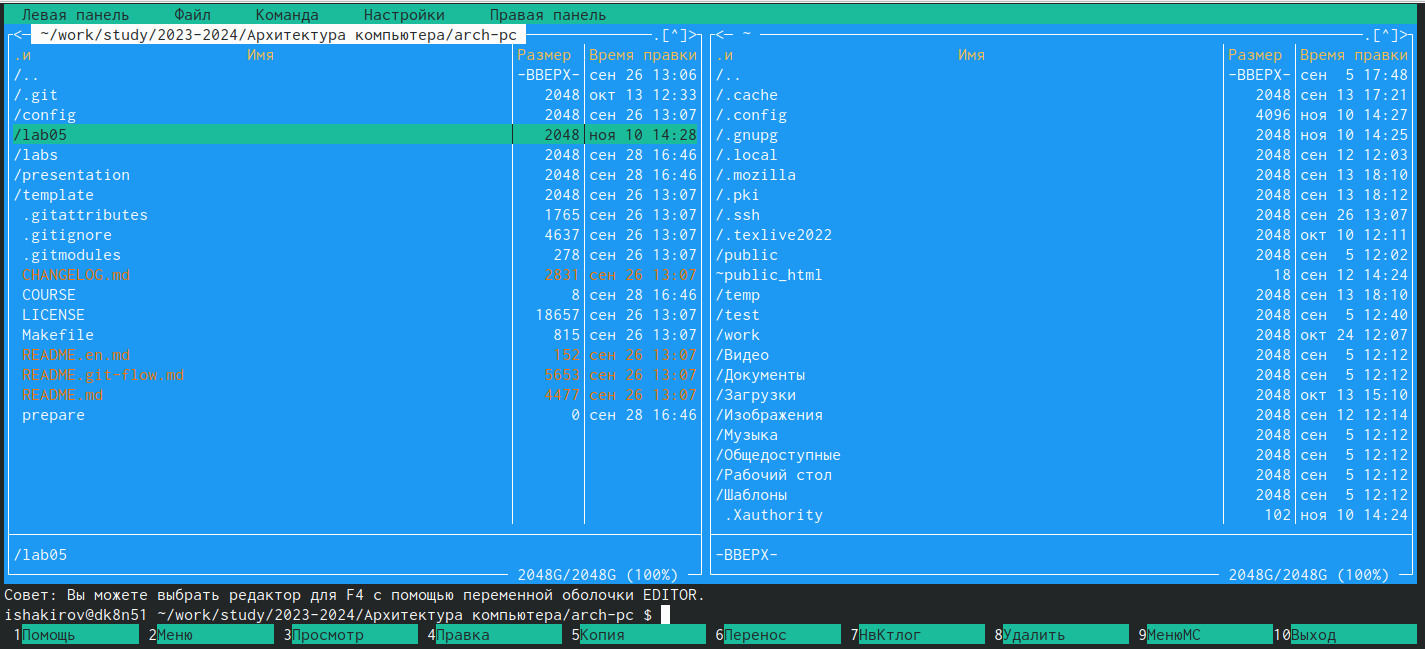
# 4 Выполнение лабораторной работы

Открываю Midnight Commander, введя в терминал mc

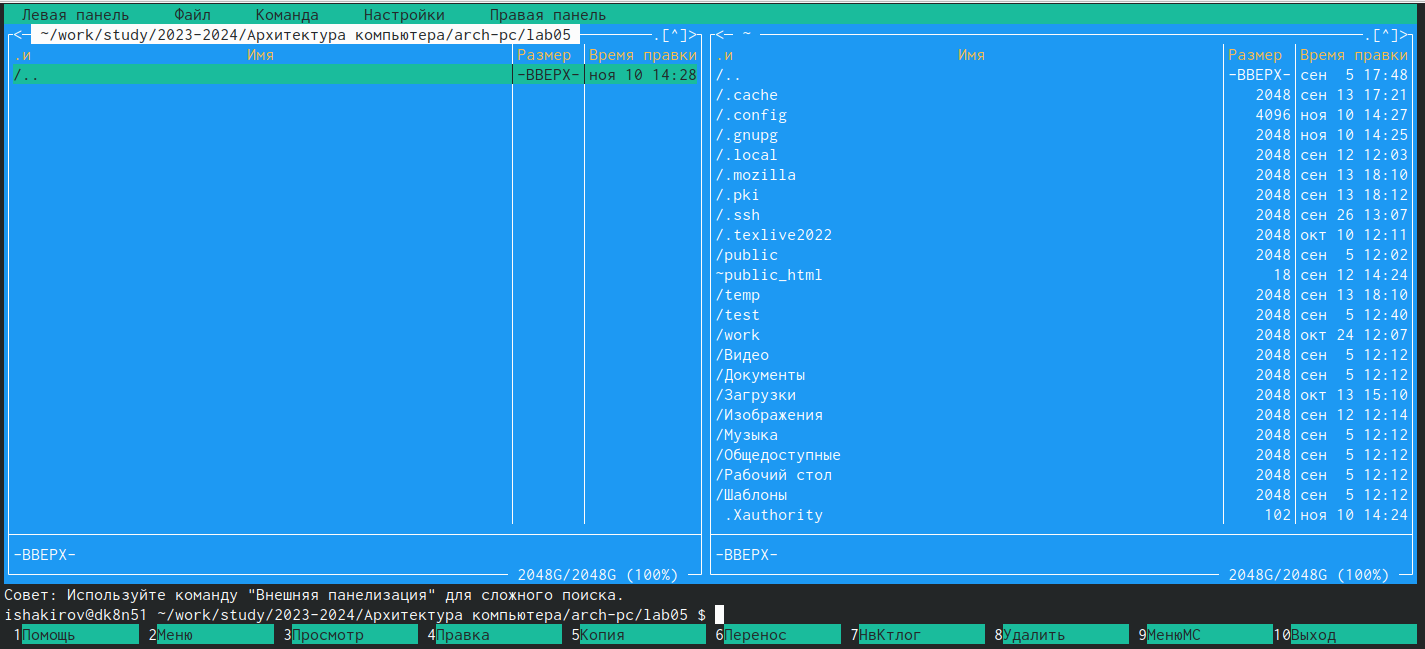


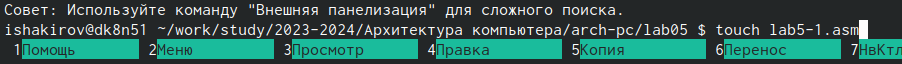
Открытый mc

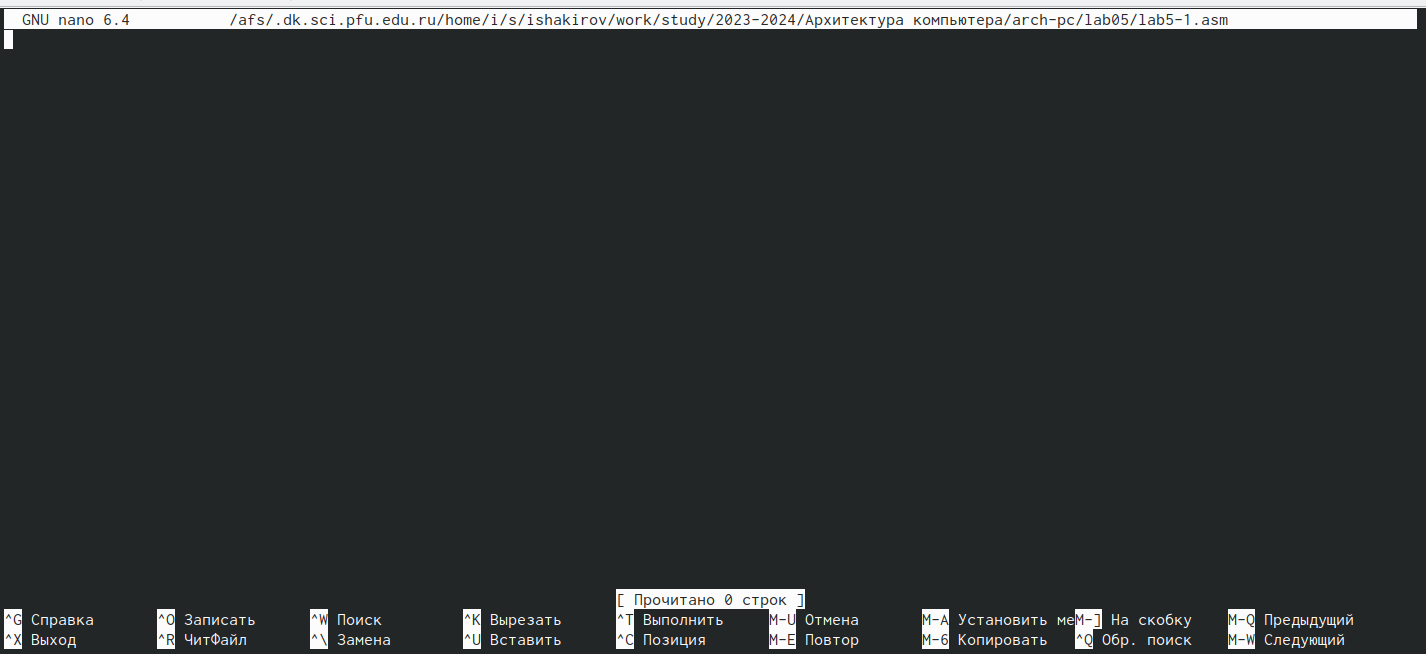
Перехожу в каталог -/work/study/2022-2023/Архитектура Компьютера/arch-pc, используя файловый менеджер mc 

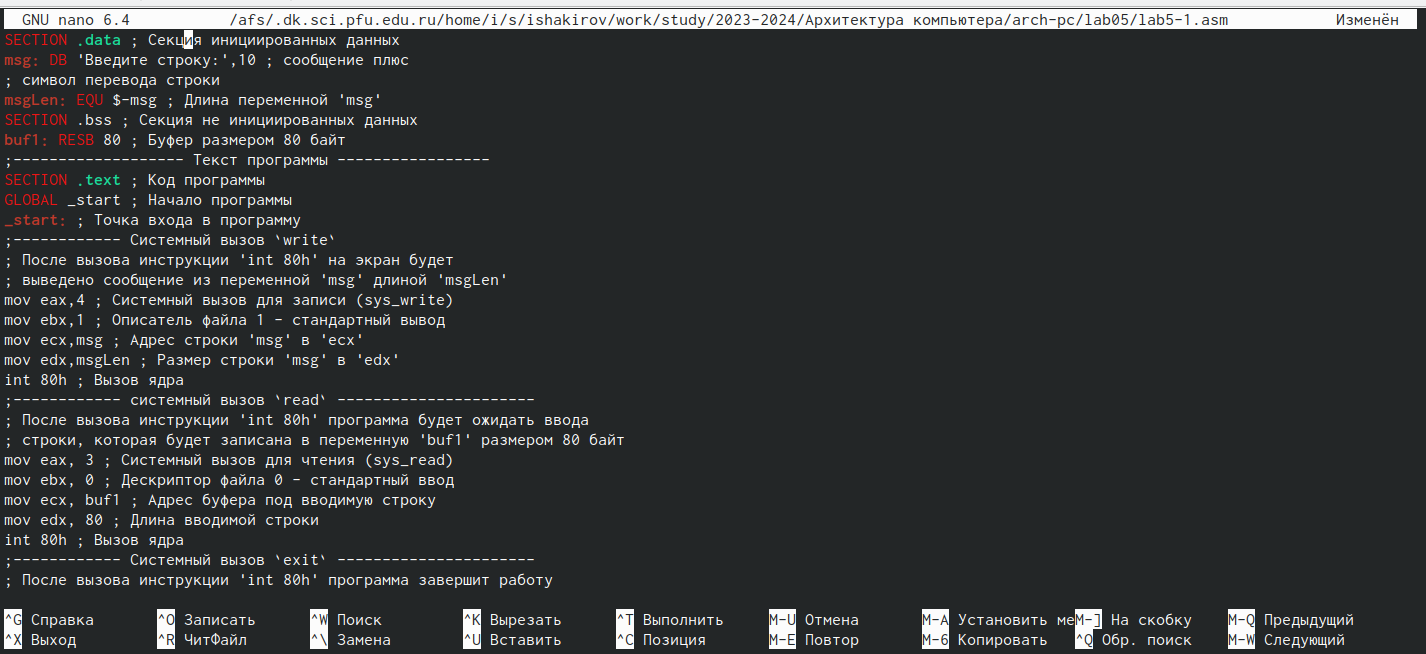


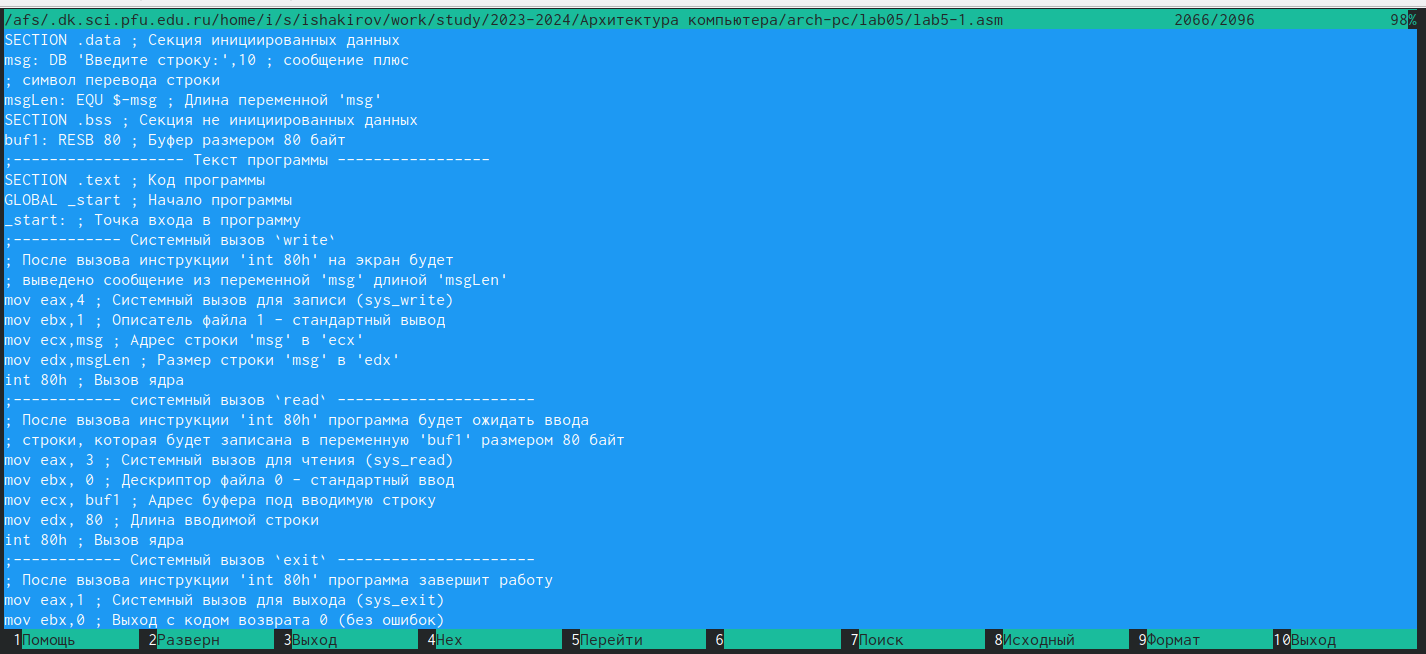
Перемещение между директориями

С помощью функциональной клавиши F7 создаю каталог lab05 Переходу в созданный каталог 

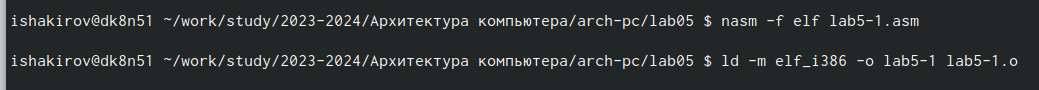
В строке ввода прописываю команду touch lab5-1.asm, чтобы создать файл, в котором буду работать 

С помощью функциональной клавиши F4 открываю созданный файл для ре-актирования в редакторе nano 

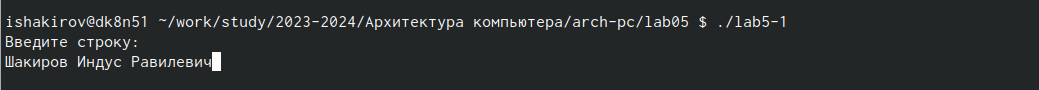
Ввожу в файл код программы для запроса строки у пользователя.Далее выхожу из файла (Ctrl+X), сохраняя изменения (Y, Enter). 

С помощью функциональной клавиши F3 открываю файл для просмотра, чтобы проверить, содержит ли файл текст программы 

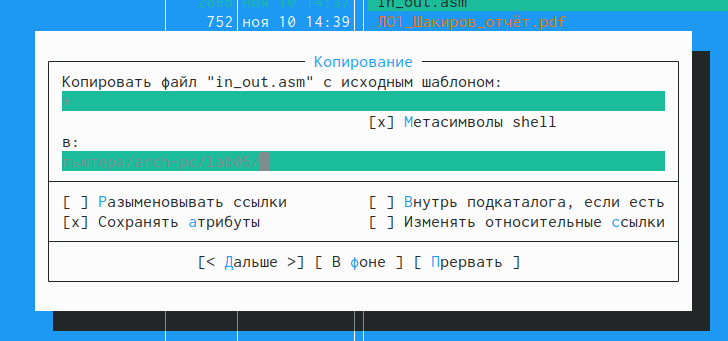
Транслирую текст программы файла в объектный файл командой nasm -f elf lab5-1.asm. Создался объектный файл lab5-1.0. Выполняю компоновку объектного файла с помощью команды Id -m elf i386 -o lab5-1 lab5-1.0. Создался исполняемый файл lab5-1.

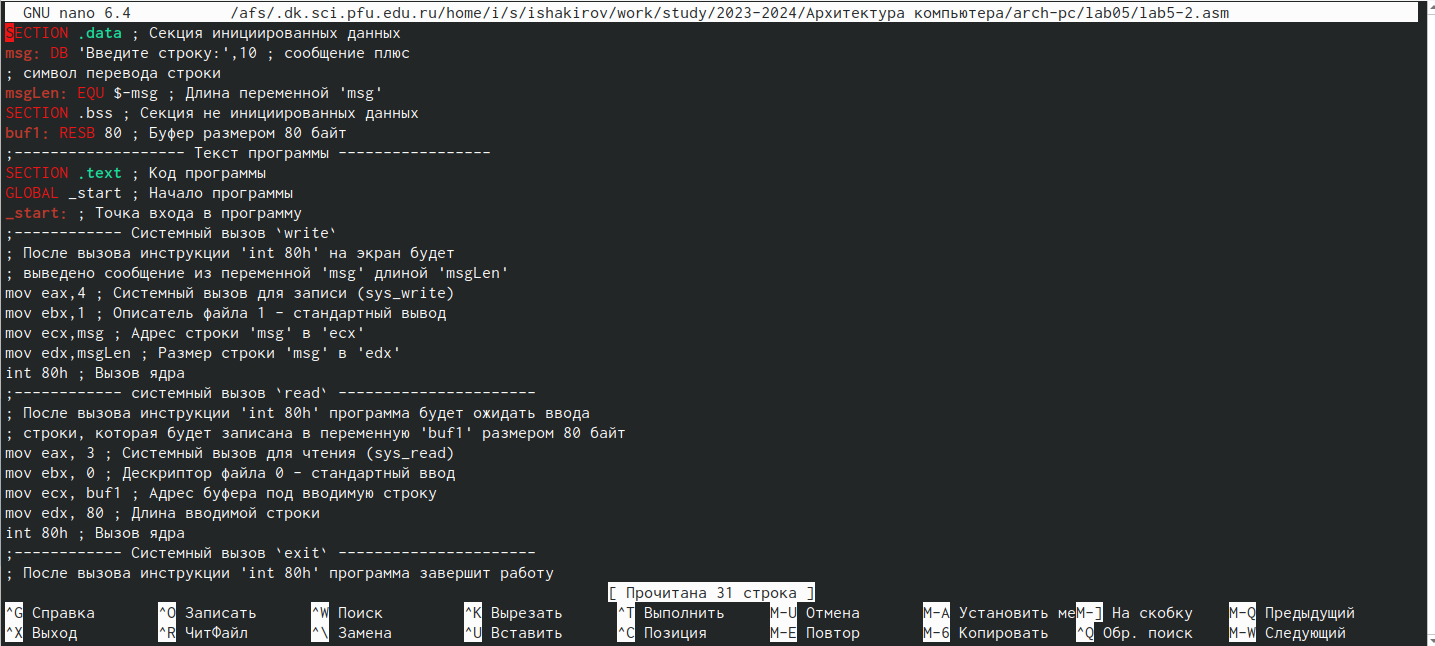


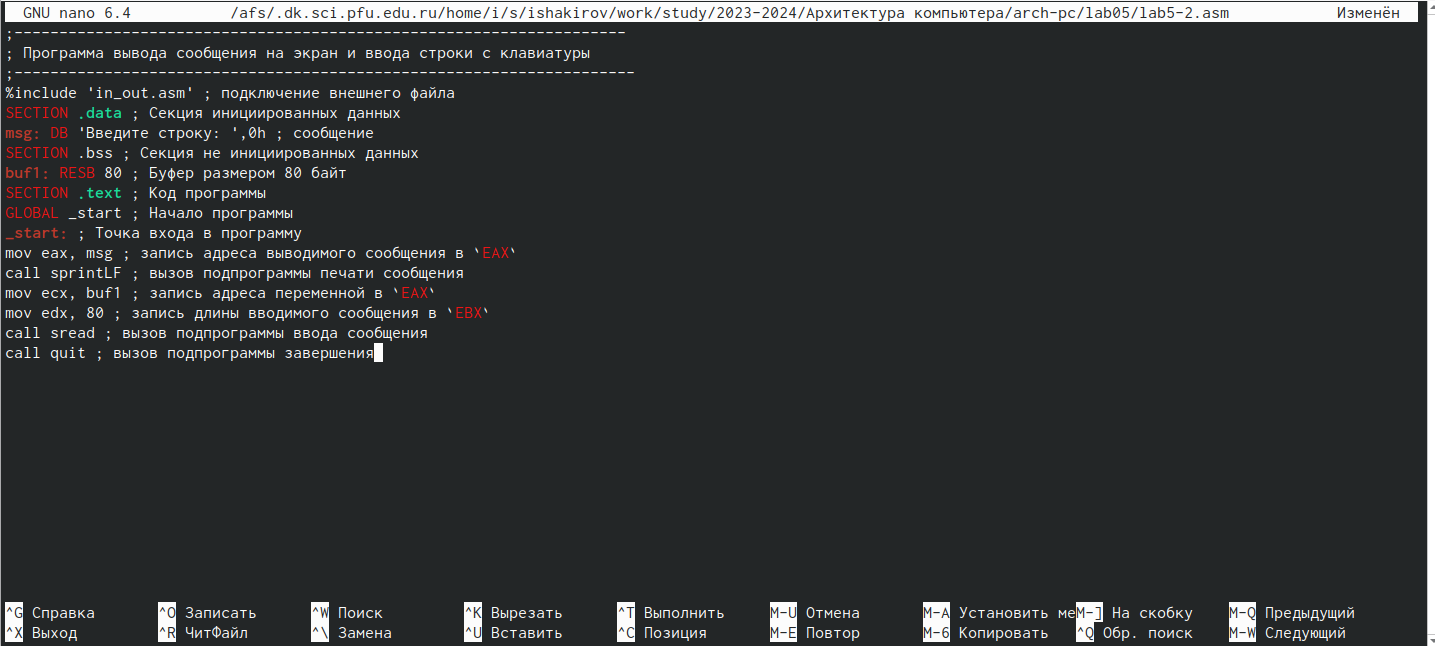
Компиляция файла

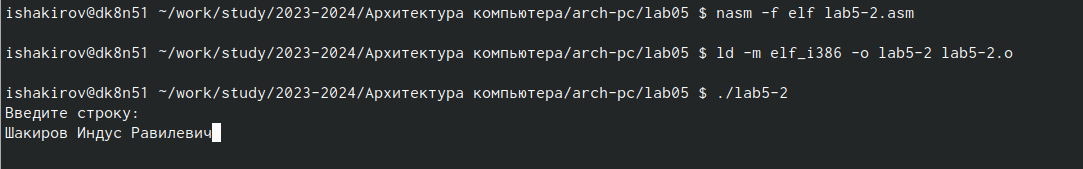
Запускаю исполняемый файл. Программа выводит строку “Введите строку:” и ждет ввода с клавиатуры, я ввожу свои ФИО, на этом программа заканчивает свою работу 

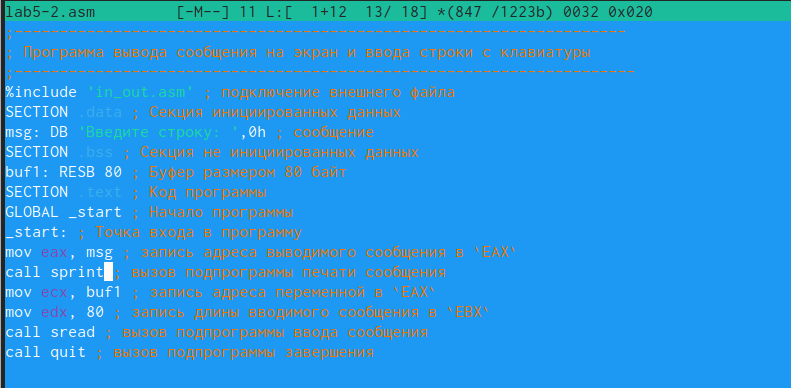
Скачиваю файл in out.asm со страницы курса в ТУИС. Он сохранился в каталог “Загрузки” 

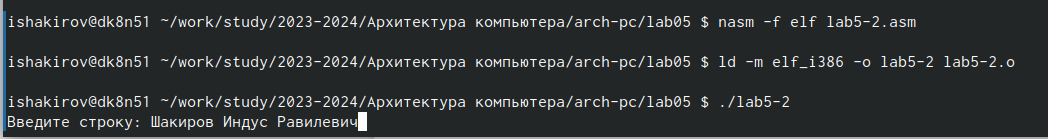
С помощью функциональной клавиши F5 копирую файл in out.asm из каталога Загрузки в созданный каталог lab05 

С помощью функциональной клавиши F5 копирую файл lab5-1 в тот же каталог, но с другим именем, для этого в появившемся окне тс прописываю имя для копии файла 

Изменяю содержимое файла lab5-2.asm во встроенном редакторе nano, чтобы в программе использовались подпрограммы из внешнего файла in\_out.asm. 

Транслирую текст программы файла в объектный файл командой nasm -f elf lab5-2.asm. Создался объектный файл lab5-2.0. Выполняю компоновку объектного файла с помощью команды Id - m elf i386 -o lab5-2 lab5-2.0 Создался исполняемый файл lab5-2. Запускаю исполняемый файл 

Открываю файл lab5-2.asm для редактирования в папо функциональной клавишей F4. Изменяю в нем подпрограмму sprintl.F на sprint. Сохраняю изменения и открываю файл для просмотра, чтобы проверить сохранение действий 

Снова транслирую файл, выполняю компоновку созданного объектного файла, запускаю новый исполняемый файл 

Разница между первым исполняемым файлом lab5-2 и вторым lab5-2-2 в том, что запуск первого запрашивает ввод с новой строки, а программа, которая исполняется при запуске второго, запрашивает ввод без переноса на новую строку, потому что в этом заключается различие между подпрограммами sprintLF и sprint.

# 5 Выводы

При выполнении данной лабораторной работы я приобрел практические навыки работы в midnight commander, а также освоил инструкции языка ассэмблера mov и int.

# Список литературы

1. Лабораторная работа №5