Отчет по лабораторной работе №5

Дисциплина: архитектура компьютера

Гомазкова Алина

Содержание

1 Цель работы **5**

2 Задание **6**

3 Теоретическое введение **7**

4 Выполнение лабораторной работы **9**
4.1 Основы работы с mc 9
4.2 Структура программы на языке ассемблера NASM 11
4.3 Подключение внешнего файла 12
4.4 Выполнение заданий для самостоятельной работы 16

5 Выводы **20**

Список литературы **21**

Список иллюстраций

4.1 Рис. 1 Открытый mc 9
4.2 Рис. 2 Перемещение между директориями 10
4.3 Рис. 3 Создание каталога 10
4.4 Рис. 4 Перемещение между директориями 10
4.5 Рис. 5 Создание файла 11
4.6 Рис. 6 Редактирование файла 11
4.7 Рис. 7 Компиляция файла и передача на обработку компоновщику 12
4.8 Рис. 8 Исполнение файла 12
4.9 Рис. 9 Исполнение файла 13
4.10 Рис. 10 Копирование файла 13
4.11 Рис. 11 Копирование файла 14
4.12 Рис. 12 Редактирование файла 14
4.13 Рис. 13 Исполнение файла 15
4.14 Рис. 14 15
4.15 Рис. 15 Отредактированный файл 15
4.16 Рис. 16 Исполнение файла 15
4.17 Рис. 17 Копирование файла 16
4.18 Рис. 18 Редактирование файла 17
4.19 Рис. 19 Исполнение файла 17
4.20 Рис. 20 Копирование файла 18
4.21 Рис. 21 Редактирование файла 18
4.22 Рис. 22 Исполнение файла 19

Список таблиц

1 Цель работы

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 (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, введя в терминал mc (рис. 1)

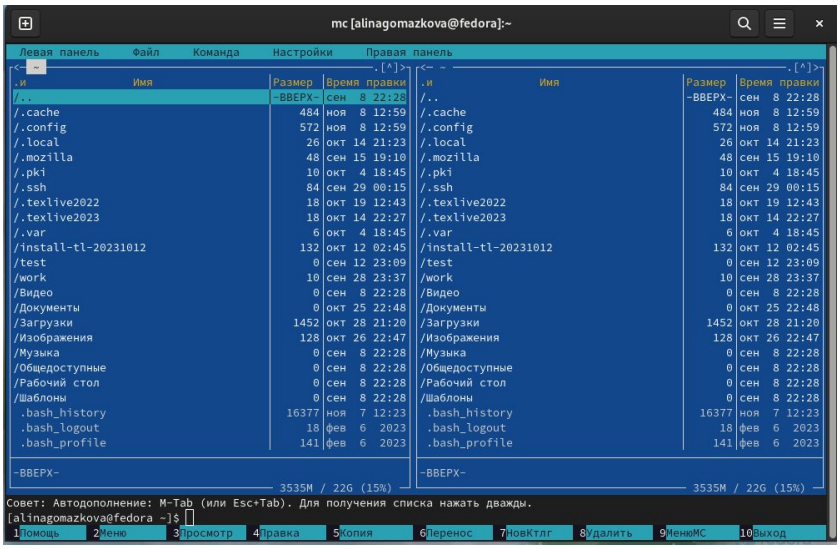

Рис. 4.1: Рис. 1 Открытый mc

Перехожу в каталог ~/work/study/2023-2024/Архитектура Компьютера/arch-pc, используя файловый менеджер mc (рис. 2)

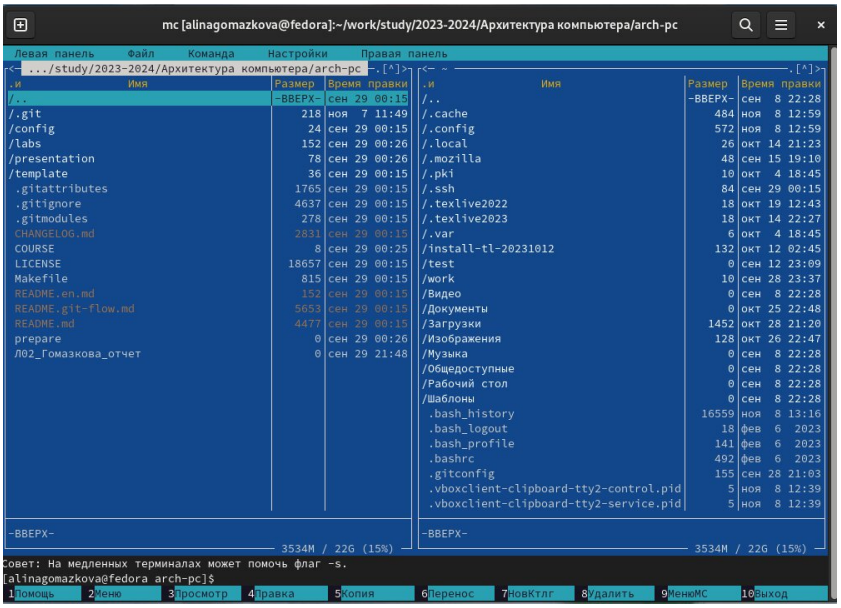

Рис. 4.2: Рис. 2 Перемещение между директориями

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

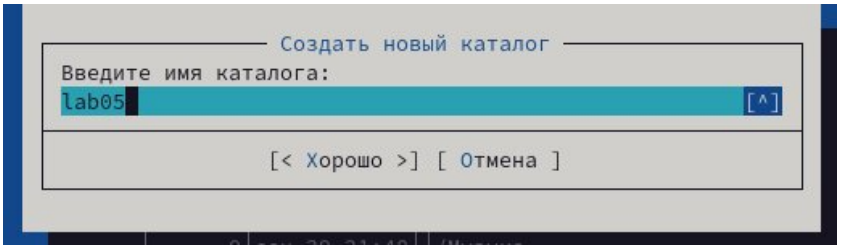

Рис. 4.3: Рис. 3 Создание каталога

Переходу в созданный каталог (рис. 4)

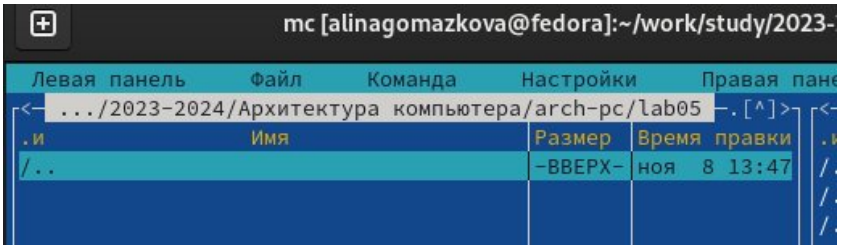

Рис. 4.4: Рис. 4 Перемещение между директориями

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

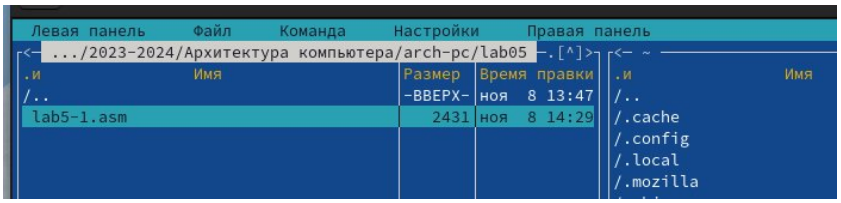

Рис. 4.5: Рис. 5 Создание файла

4.2 Структура программы на языке ассемблера NASM

С помощью функциональной клавиши F4 открываю созданный файл для редактирования. 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 (рис. 6)

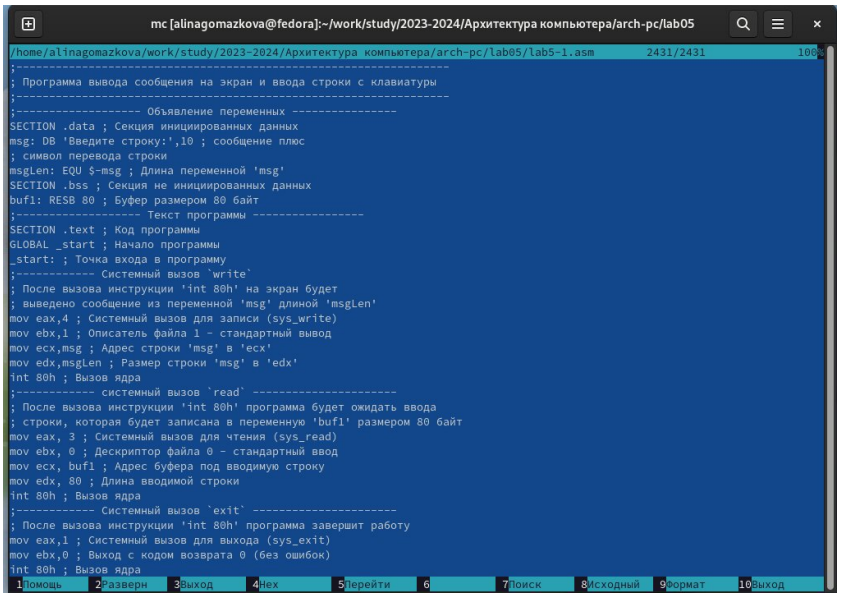

Рис. 4.6: Рис. 6 Редактирование файла

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. Создался исполняемый файл lab5-1 (рис. 7)

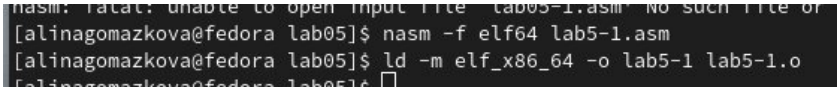

Рис. 4.7: Рис. 7 Компиляция файла и передача на обработку компоновщику

Запускаю исполняемый файл. Программа выводит строку “Введите строку:” и ждет ввода с клавиатуры, я ввожу свои ФИО, на этом программа заканчивает свою работу (рис. 8)

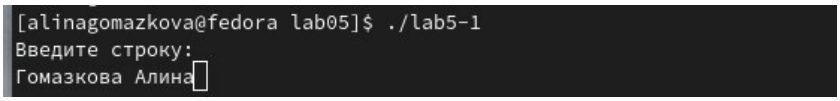

Рис. 4.8: Рис. 8 Исполнение файла

4.3 Подключение внешнего файла

Скачиваю файл in_out.asm со страницы курса в ТУИС. Он сохранился в каталог “Загрузки” (рис. 9)

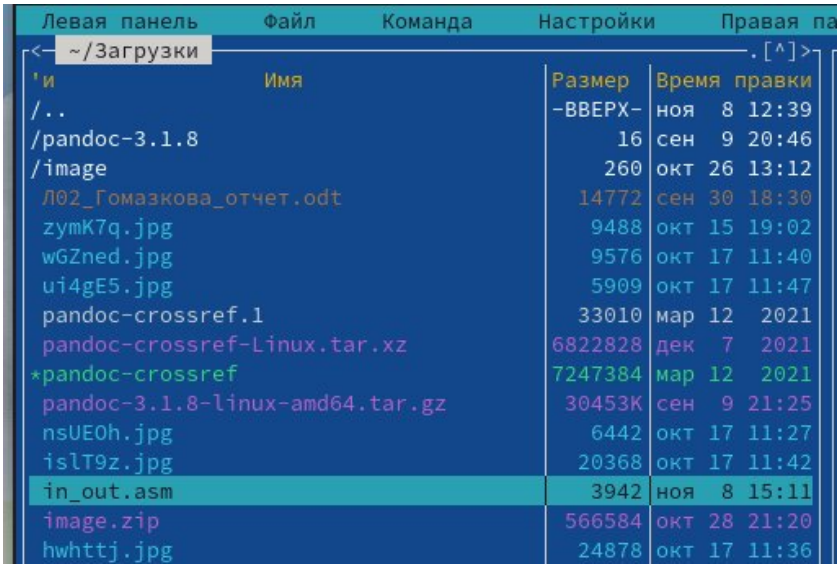

Рис. 4.9: Рис. 9 Исполнение файла

С помощью функциональной клавиши F5 копирую файл in_out.asm из каталога Загрузки в созданный каталог lab05 (рис. 10)

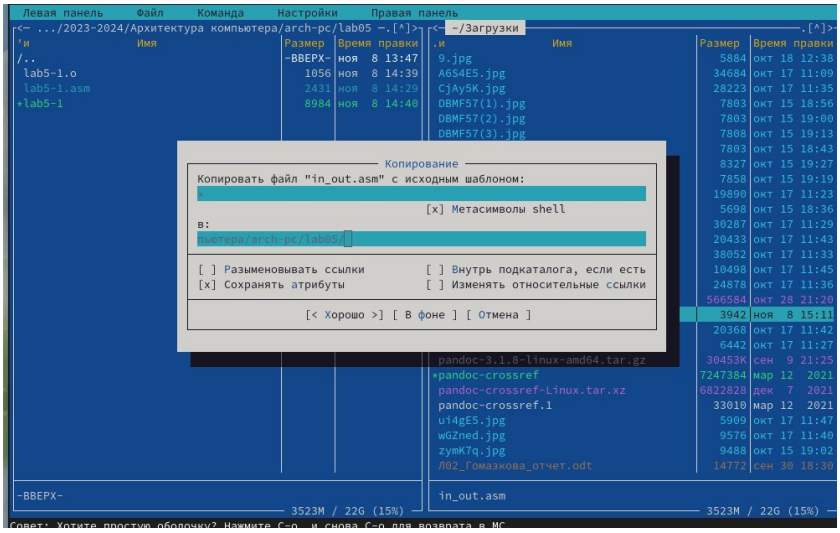

Рис. 4.10: Рис. 10 Копирование файла

С помощью функциональной клавиши F5 копирую файл lab5-1 в тот же каталог, но с другим именем, для этого в появившемся окне mc прописываю имя для копии файла (рис. 11)

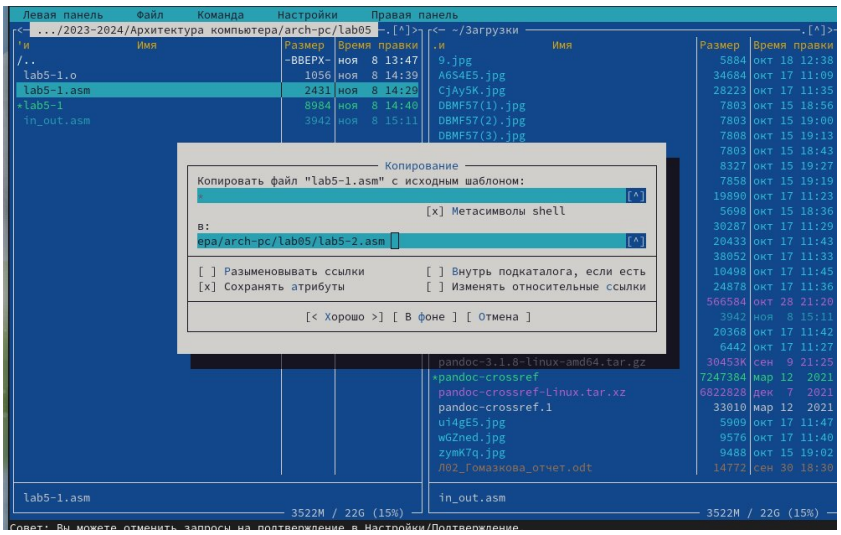

Рис. 4.11: Рис. 11 Копирование файла

Изменяю содержимое файла lab5-2.asm, чтобы в программе использовались подпрограммы из внешнего файла in_out.asm (рис. 12)

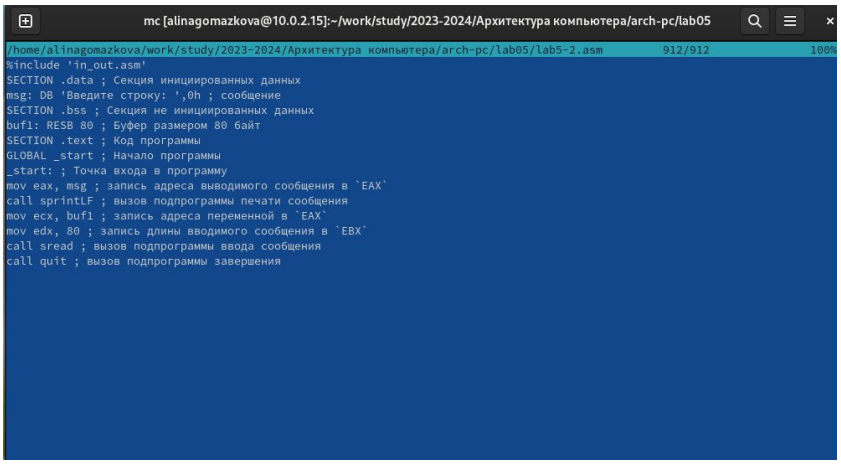

Рис. 4.12: Рис. 12 Редактирование файла

Транслирую текст программы файла 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 исполняемый файл (рис. 13)(рис. 14)

Рис. 4.13: Рис. 13 Исполнение файла

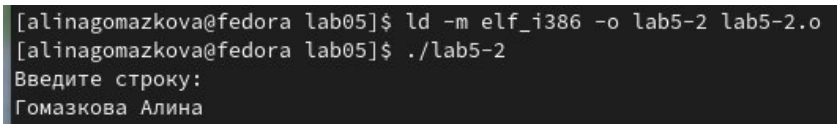

Рис. 4.14: Рис. 14

Открываю файл lab5-2.asm для редактирования функциональной клавишей F4. Изменяю в нем подпрограмму sprintLF на sprint. Сохраняю изменения и открываю файл для просмотра, чтобы проверить сохранение действий (рис. 15)

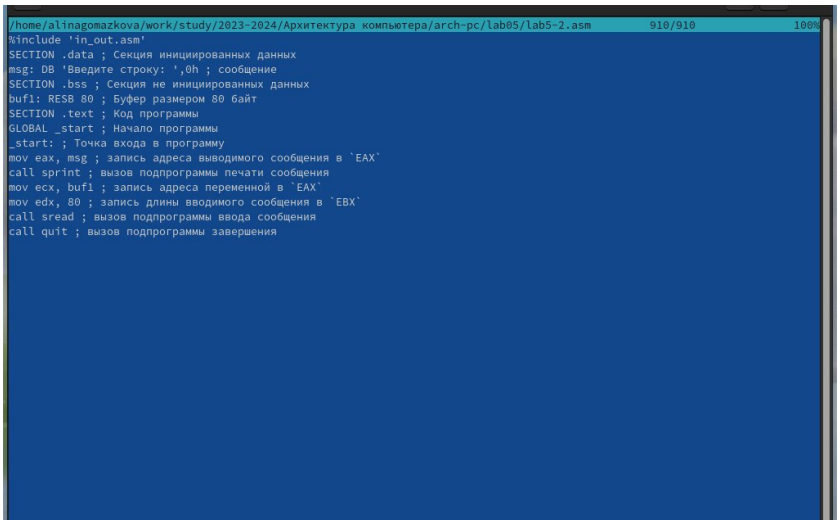

Рис. 4.15: Рис. 15 Отредактированный файл

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

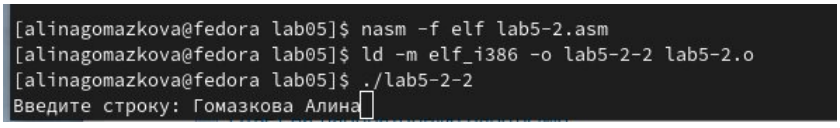

Рис. 4.16: Рис. 16 Исполнение файла

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.

4.4 Выполнение заданий для самостоятельной работы

1. Создаю копию файла lab5-1.asm с именем lab5-1-1.asm с помощью функциональной клавиши F5 (рис. 17)

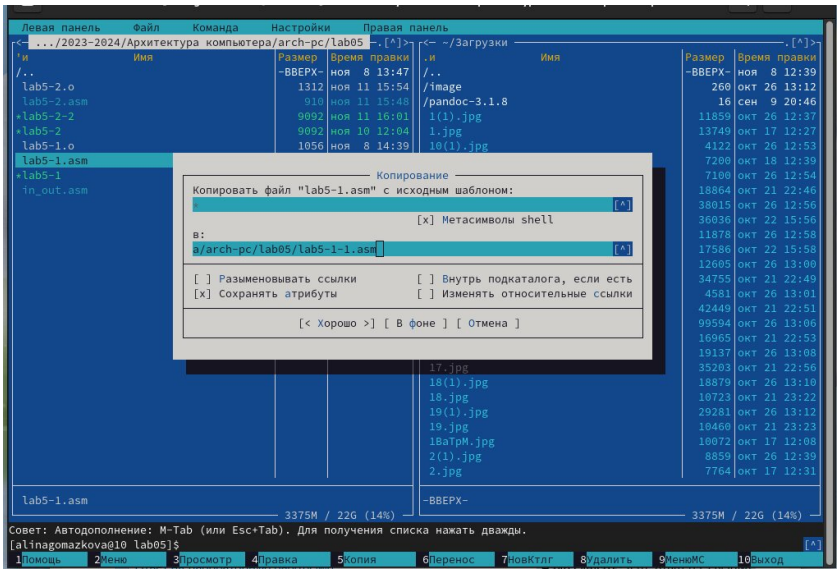

Рис. 4.17: Рис. 17 Копирование файла

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)

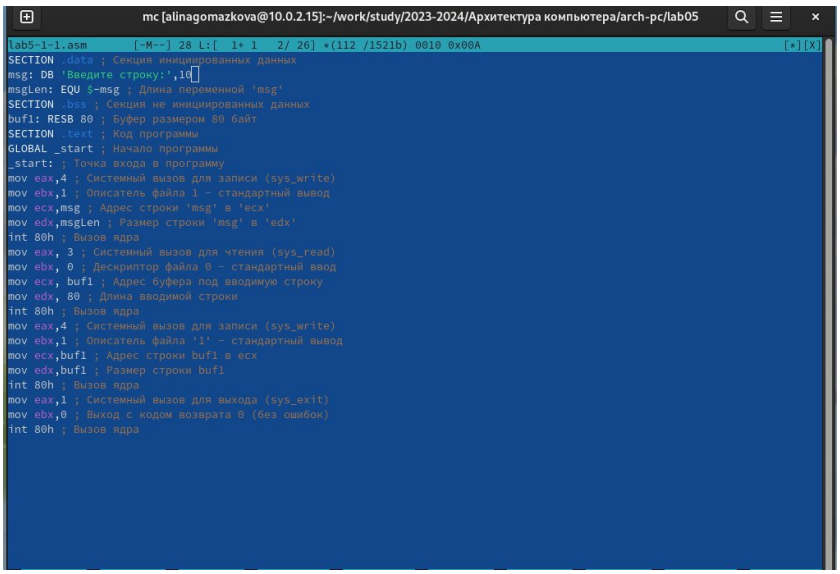

Рис. 4.18: Рис. 18 Редактирование файла

2. 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, ввожу свои ФИО, далее программа выводит введенные мною данные (рис. 19)

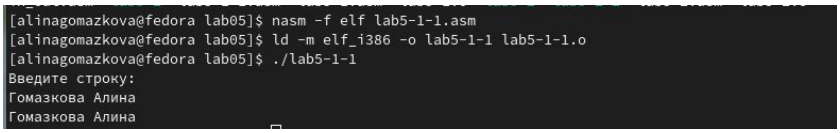

Рис. 4.19: Рис. 19 Исполнение файла

3. Создаю копию файла lab5-2.asm с именем lab5-2-1.asm с помощью функциональной клавиши F5 (рис. 20)

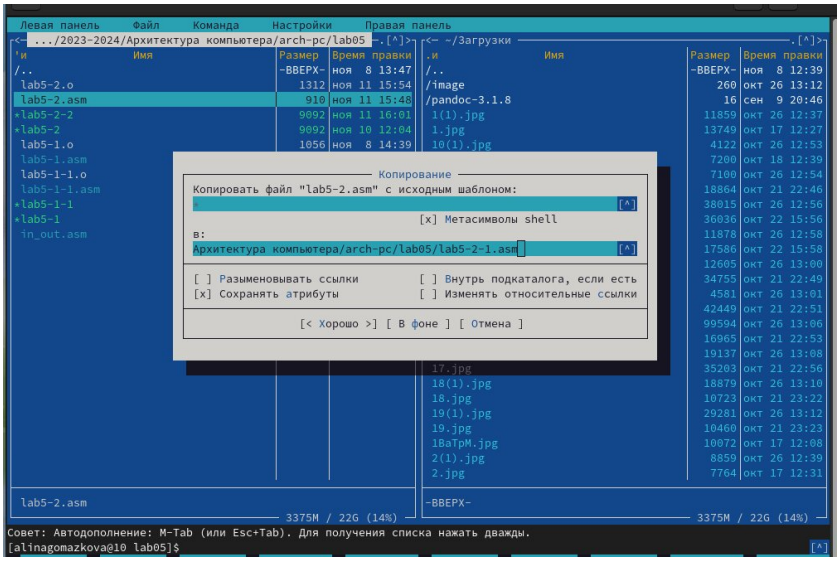

Рис. 4.20: Рис. 20 Копирование файла

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. 21)

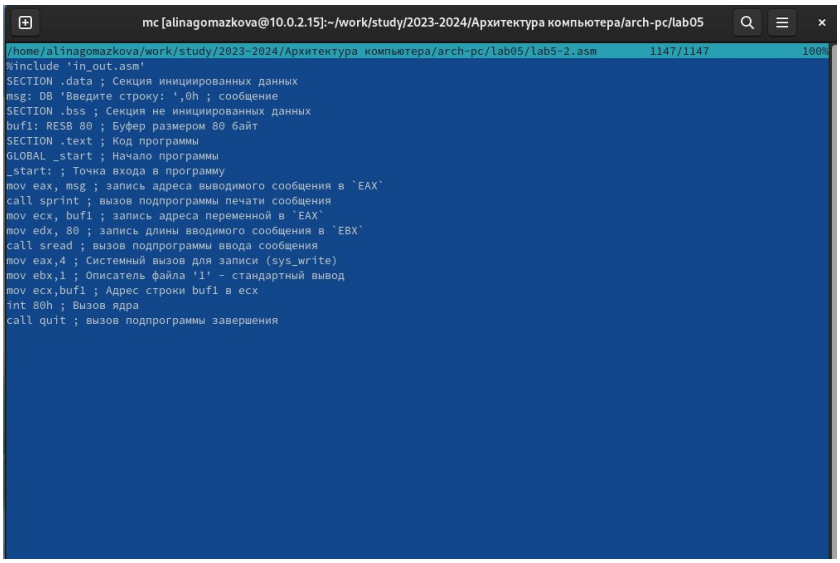

Рис. 4.21: Рис. 21 Редактирование файла

4. 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 мною данные (рис. 22)

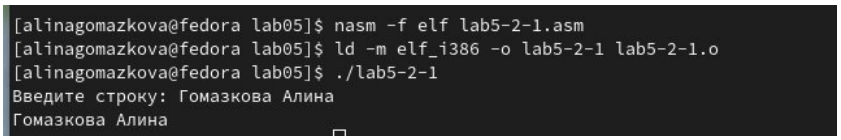

Рис. 4.22: Рис. 22 Исполнение файла

5 Выводы

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

Список литературы

1.Архитектура ЭВМ