Отчёт по лабораторной работе

Дисциплина: архитектура компьютера

Назармамадов Умед Джамшедович

Содержание

1 Цель работы

2 Задание

3 Теоретическое введение

4 Выполнение лабораторной работы

5 Выводы

Список литературы

Список иллюстраций

4.1 Midnight commander 9
4.2 Перемещение между директориями 10
4.3 Перемещение между директориями 10
4.4 Создание файла 10
4.5 Открытие файла 11
4.6 Редактирование файла 11
4.7 Открытие файла 12
4.8 Компиляция файла 12
4.9 Запуск файла 12
4.10 Скачивание файла 13
4.11 Копиравание файла 13
4.12 Копиравание файла 13
4.13 Редактирование файла 14
4.14 Исполнение файла 14
4.15 Исполнение файла 14
4.16 Отредактированние файла 15
4.17 Запуск файла 15

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 просматриват

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 ко

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. Синтаксис директив определения данных следующий: DB [,] [,]

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.

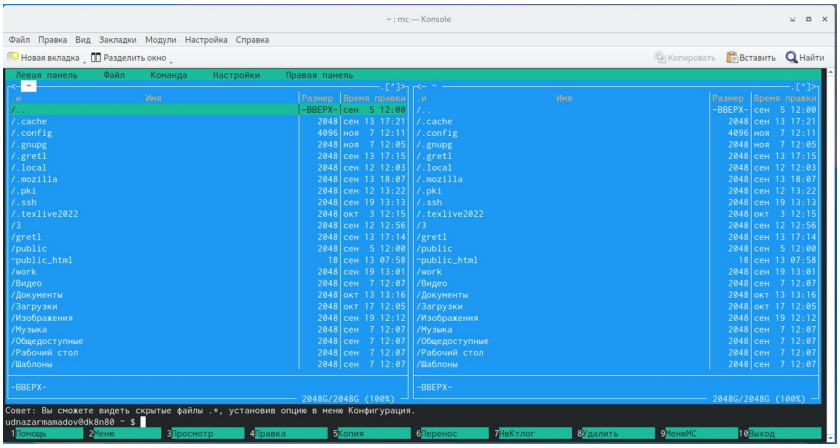

Рис. 4.1: Midnight commander

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

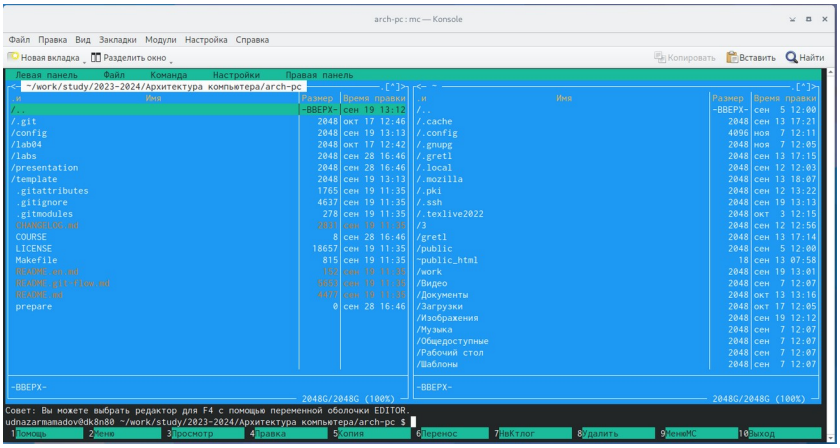

Рис. 4.2: Перемещение между директориями

С помощью функциональной клавиши F7 создаю каталог lab05 и перехожу в созданный каталог.

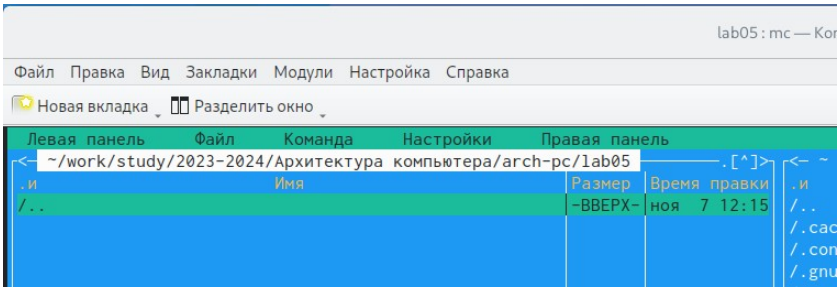

Рис. 4.3: Перемещение между директориями

Прописываю команду touch lab5.asm, чтобы создать файл, в котором буду работать.

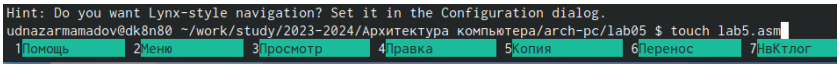

Рис. 4.4: Создание файла

С помощью функциональной клавиши F4 открываю созданный файл для редактирования в редакторе nano.

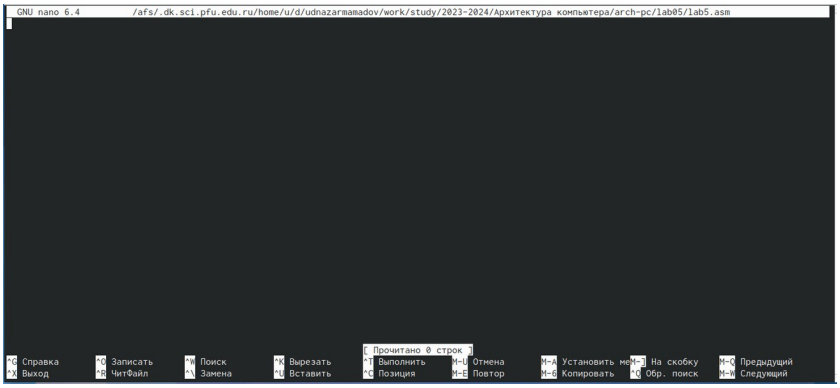

Рис. 4.5: Открытие файла

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

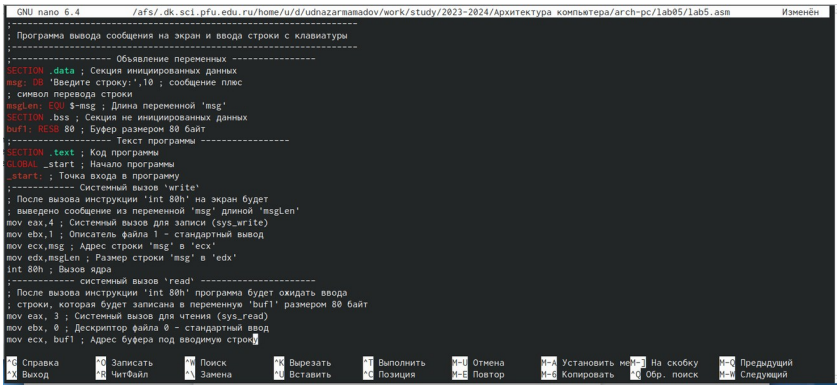

Рис. 4.6: Редактирование файла

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

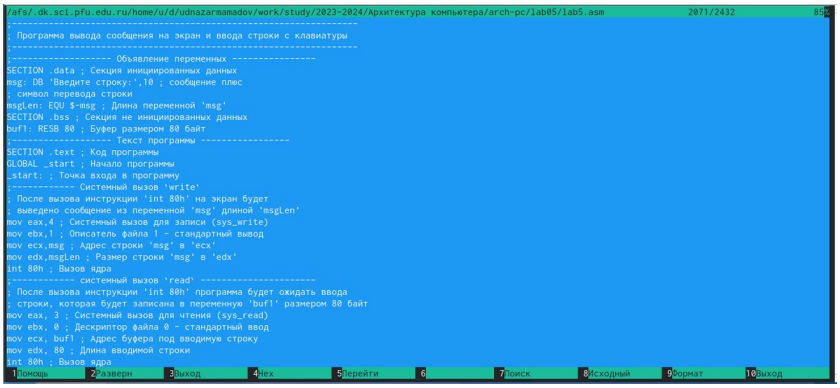

Рис. 4.7: Открытие файла

Транслирую текст программы файла в объектный файл командой nasm -f elf lab5.asm. Создался объектный файл lab5.0. Выполняю компоновку объектного файла с помощью команды ld -m elf i386 -o lab5 lab5.0.

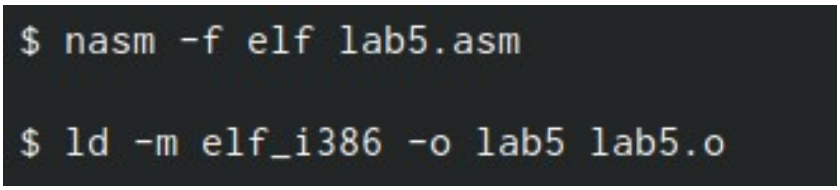

Рис. 4.8: Компиляция файла

Запускаю исполняемый файл.

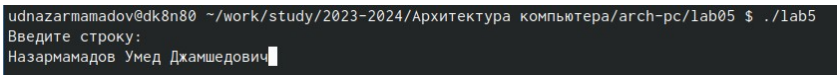

Рис. 4.9: Запуск файла

Скачиваю файл in out.asm со страницы курса в ТУИС. С помощью функциональной клавиши F5 копирую файл in out.asm из каталога загрузки в созданный каталог lab05

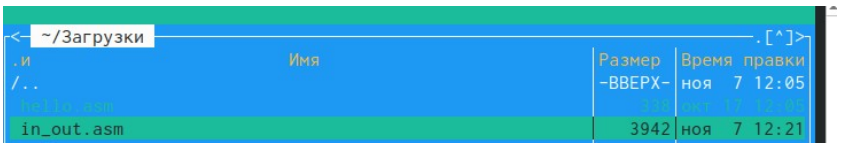

Рис. 4.10: Скачивание файла

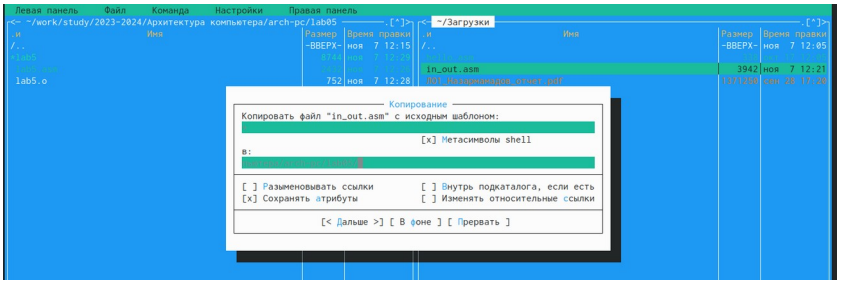

Рис. 4.11: Копиравание файла

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

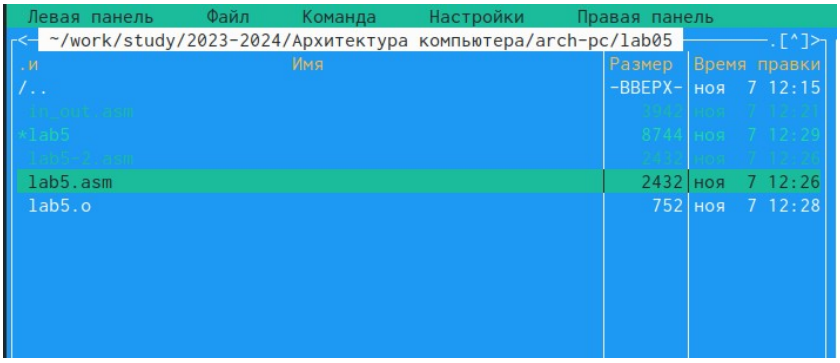

Рис. 4.12: Копиравание файла

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

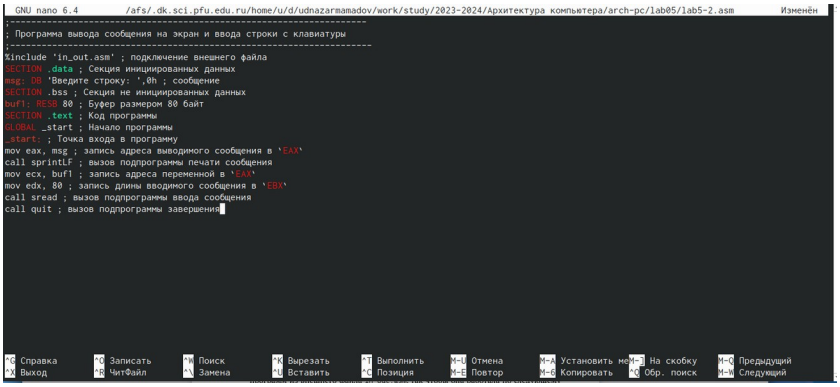

Рис. 4.13: Редактирование файла

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 ld - m elf i386 -o lab5-2 lab5-2.0 Создался исполняемый файл lab5-2. Запускаю исполняемый файл

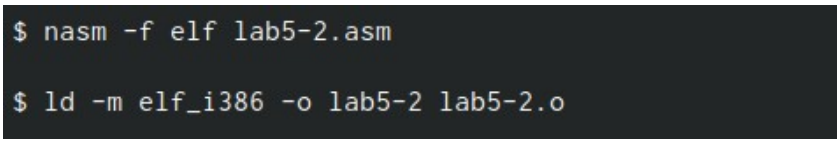

Рис. 4.14: Исполнение файла

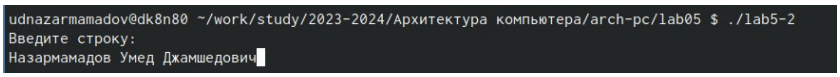

Рис. 4.15: Исполнение файла

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

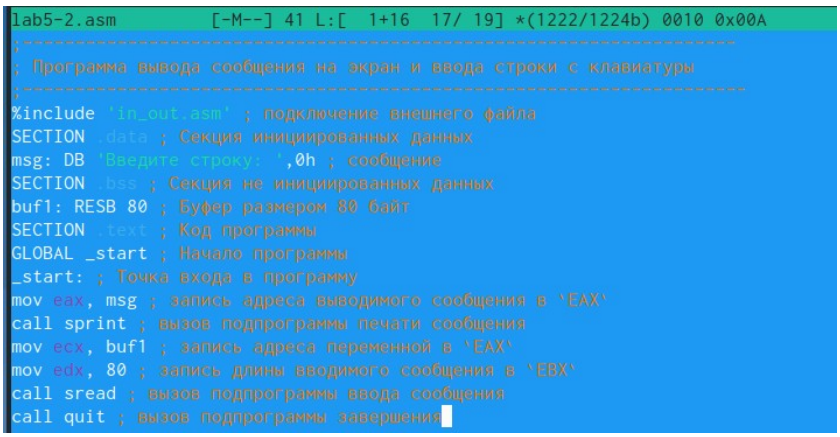

Рис. 4.16: Отредактированние файла

Запускаю новый исполняемый файл.

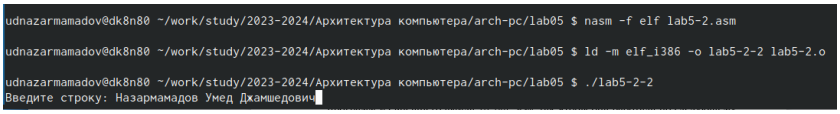

Рис. 4.17: Запуск файла

5 Выводы

При выполнение лабораторной работы я приобрел практические навыки работы в Midnight Commander и инструкции языка ассемблера mov и int.

Список литературы

1. Лабораторная работа №5