**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**по лабораторной работе № *5***

*дисциплина: Архитектура компьютера*

Студент: Парчиев Султан Багаудинович

Группа: НММ бд-03-24

**МОСКВА**

**2024 г.**

**Содержание**

**1 Цель работы 4**

**2 Задание 5**

**3 Теоретическое введение 6**

**4 Выполнение лабораторной работы 7**

**5 Выводы 16**

**Список литературы 17**

**Список иллюстраций**

**4.1 Открытый mc 7**

**4.2 Перемещение между директориями 8**

**4.3 Создание каталога 8**

**4.4 Перемещение между директориями 9**

**4.5 Созданный файл 9**

**4.6 Открытие файла для редактирования 9**

**4.7 Редактирование файла 10**

**4.8 Открытие файла для просмотра 10**

**4.9 Копирование файла 11**

**4.10 Редактирование файла 12**

**4.11 Исполнение файла 12**

**4.12 Исполнение файла 13**

**4.13 Копирование файла 13**

**4.14 Отредактированный файл 14**

**4.15 Исполнение файла 14**

**4.16 Отредактированный файл 15**

**4.17 Исполнение файла 15**

**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 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 6 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. 4.1).

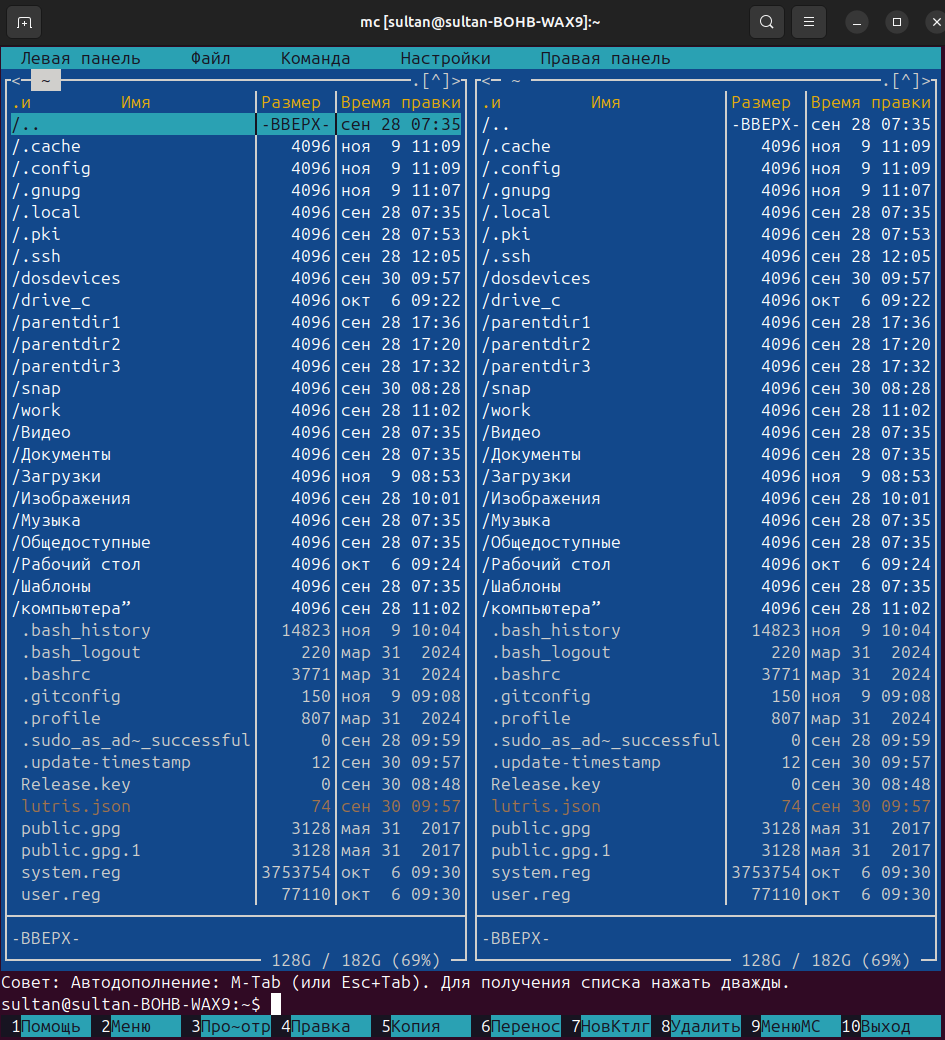


Рис. 4.1: Открытый mc

Перехожу в каталог ~/work/study/2024-2025/Архитектура Компьютера/arch-pc, используя файловый менеджер mc (рис. 4.2)

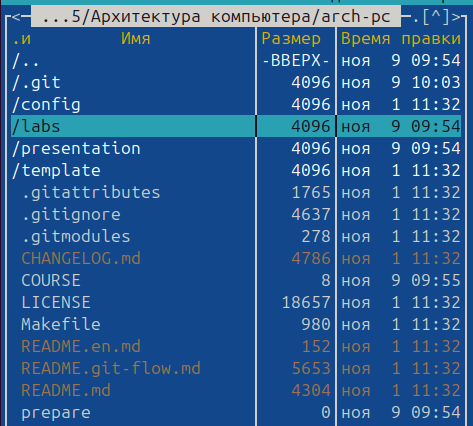


Рис. 4.2: Перемещение между директориями

С помощью функциональной клавиши F7 создаю каталог lab05 (рис. 4.3).

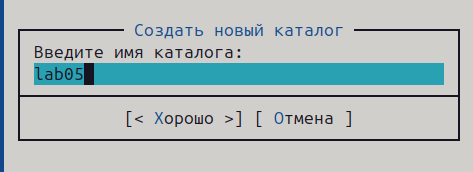


Рис. 4.3: Создание каталога

Перехожу в созданный каталог (рис. 4.4).

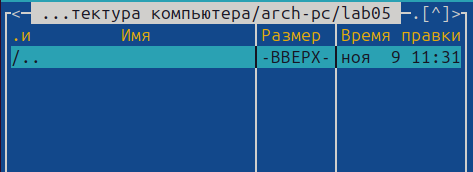


Рис. 4.4: Перемещение между директориями

В строке ввода прописываю команду touch lab05-1.asm, чтобы создать файл, в котором буду работать (рис. 4.5).

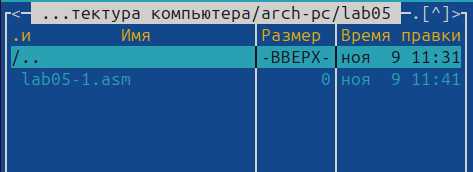
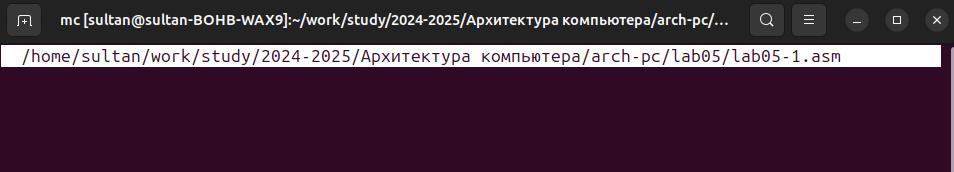


Рис. 4.5: Созданный файл

**4.2 Структура программы на языке ассемблера NASM**

С помощью функциональной клавиши F4 открываю созданный файл для редактирования в редакторе nano (рис. 4.6)

Рис. 4.6: Открытие файла для редактирования



Ввожу в файл код программы для запроса строки у пользователя (рис. 4.7). Далее выхожу из файла, сохраняя изменения.

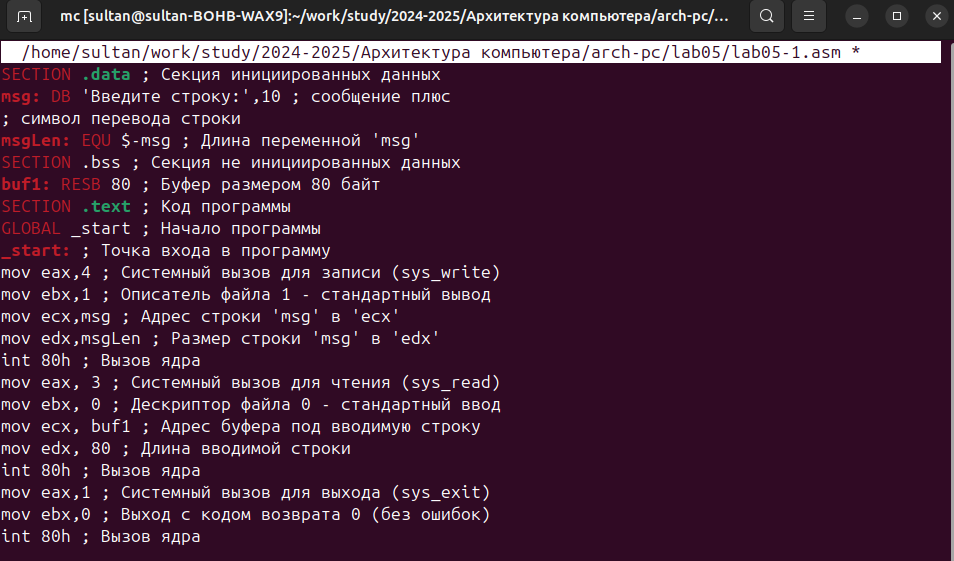


Рис. 4.7: Редактирование файла

С помощью функциональной клавиши F3 открываю файл для просмотра, чтобы проверить, содержит ли файл текст программы (рис. 4.8).

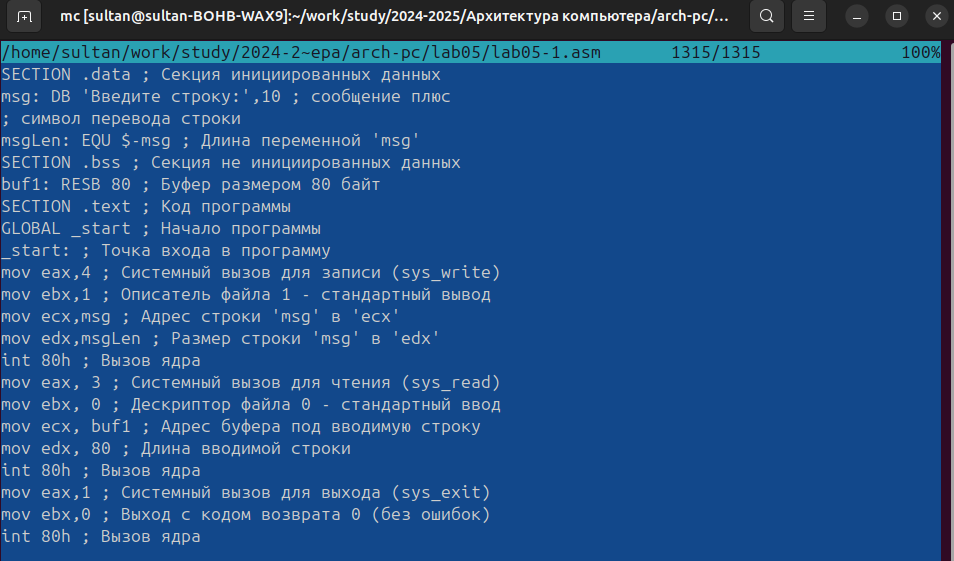


Рис. 4.8: Открытие файла для просмотра

Транслирую текст программы файла в объектный файл командой nasm -f elf lab05-1.asm. Создался объектный файл lab05-1.o. Выполняю компоновку объектного файла с помощью команды ld -m elf\_i386 -o lab05-1 lab05-1.o. Создался исполняемый файл lab05-1.

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

**4.3 Подключение внешнего 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. 4.9).

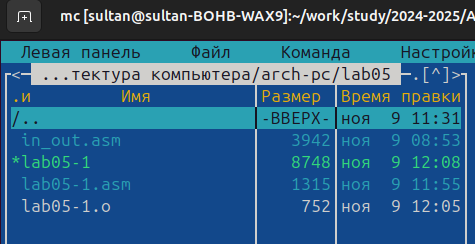


Рис. 4.9: Копирование файла

С помощью функциональной клавиши F5 копирую файл lab6-1 в тот же каталог, но с другим именем, для этого в появившемся окне mc прописываю имя для копии файла.

Изменяю содержимое файла lab6-2.asm во встроенном редакторе nano (рис. 4.10), чтобы в программе использовались подпрограммы из внешнего файла in\_out.asm.

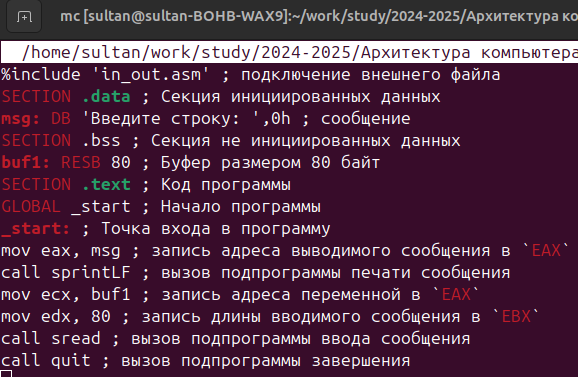


Рис. 4.10: Редактирование файла

Транслирую текст программы файла в объектный файл командой nasm -f elf lab05-2.asm. Создался объектный файл lab05-2.o. Выполняю компоновку объектного файла с помощью команды ld -m elf\_i386 -o lab05-2 lab05-2.o Создался исполняемый файл lab05-2. Запускаю исполняемый файл (рис.4.11).

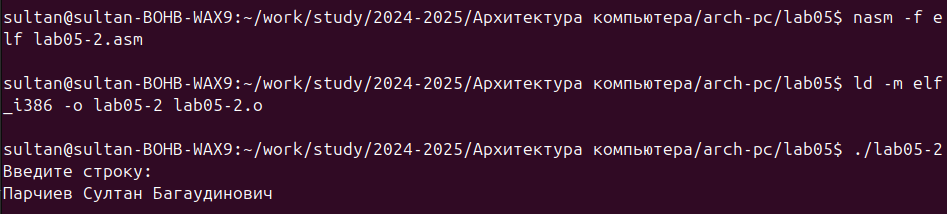
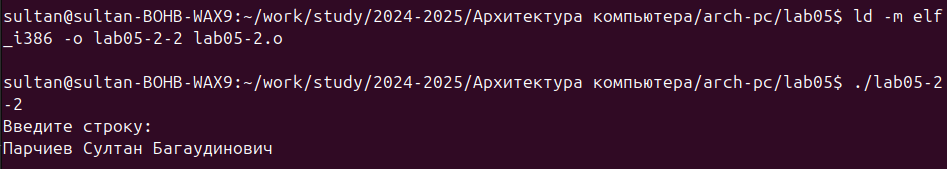


Рис. 4.11: Исполнение файла

Открываю файл lab6-2.asm для редактирования в nano функциональной клавишей F4. Изменяю в нем подпрограмму sprintLF на sprint. Сохраняю изменения. Снова транслирую файл, выполняю компоновку созданного объектного файла, запускаю новый исполняемый файл (рис. 4.12).

Рис. 4.12: Исполнение файла



Разница между первым исполняемым файлом lab05-2 и вторым lab05-2-2 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 lab05-1.asm с именем lab05-1-1.asm с помощью функциональной клавиши F5 (рис. 4.13).

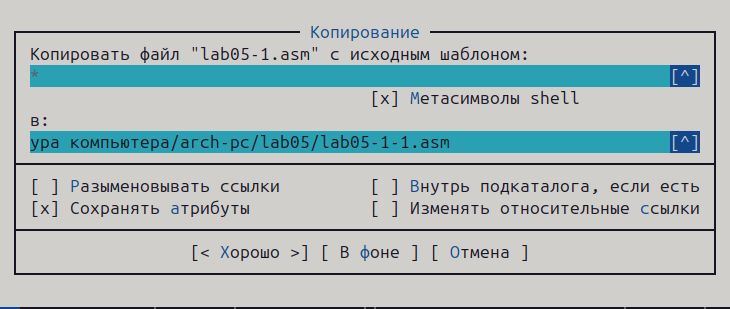


Рис. 4.13: Копирование файла

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. 4.14).

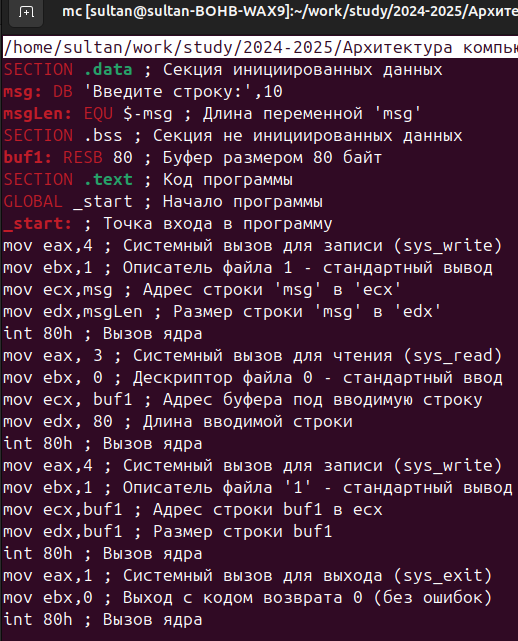
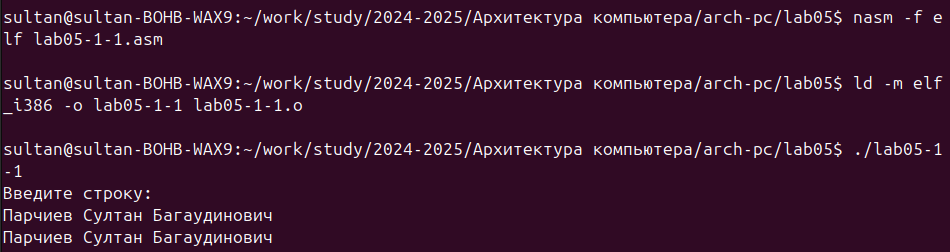


Рис. 4.14: Отредактированный файл

2. Создаю объектный файл lab05-1-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab05-1-1, запускаю полученный исполняемый файл. Программа запрашивает ввод, ввожу свои ФИО, далее программа выводит введенные мною данные (рис. 4.15).

Рис. 4.15: Исполнение файла



3. Создаю копию файла lab05-2.asm с именем lab05-2-1.asm с помощью функциональной клавиши F5. 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. 4.16).

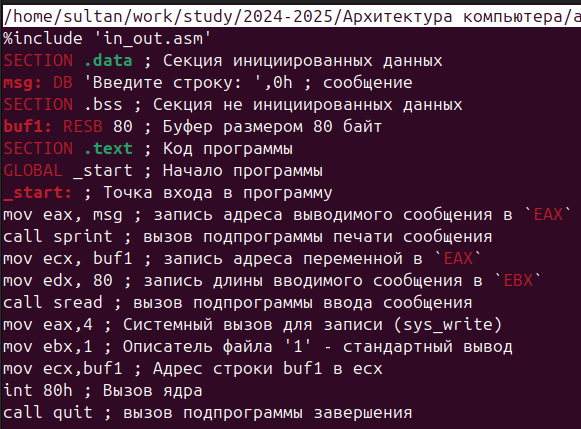


Рис. 4.16: Отредактированный файл

4. Создаю объектный файл lab05-2-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab05-2-1, запускаю полученный исполняемый файл. Программа запрашивает ввод без переноса на новую строку, ввожу свои ФИО, далее программа выводит введенные мною данные (рис. 4.17).

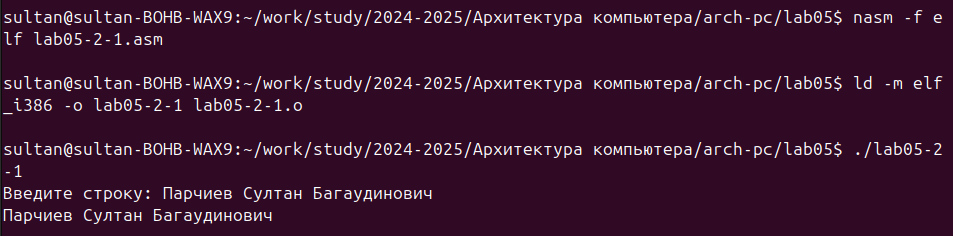


Рис. 4.17: Исполнение файла

**5 Выводы**

При выполнении данной лабораторной работы я приобрел практические навыки работы в Midnight Commander и освоил инструкции языка ассемблера mov и int

**Список литературы**

1. Архитекутра ЭВМ