Отчет по лабараторной работе №5

Дисциплина:архитектура компьютера

Никуленков Степан Сергеевич

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 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 лабараторной работы

## 4.1 Основы работы с mcг

Открываю Midnight Commander, введя в терминал mc.Перехожу в каталог ~/work/arch-pc и создаю (F7) там новый каталог lab05.

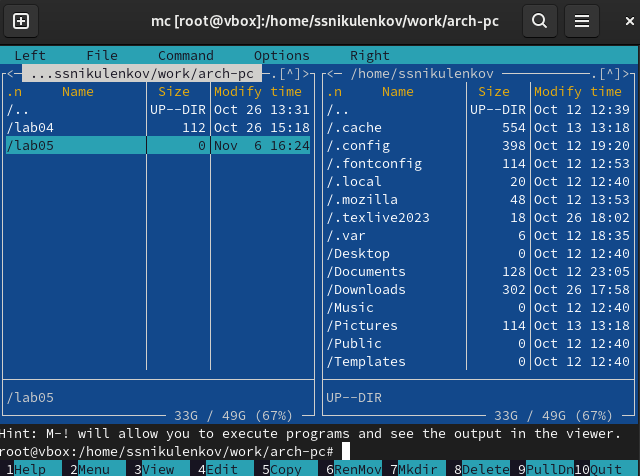


Рис. 1: Создан новый каталог lab05

Перехожу в каталог lab05 и с помощью команды touch создаю файл lab5-1.asm.

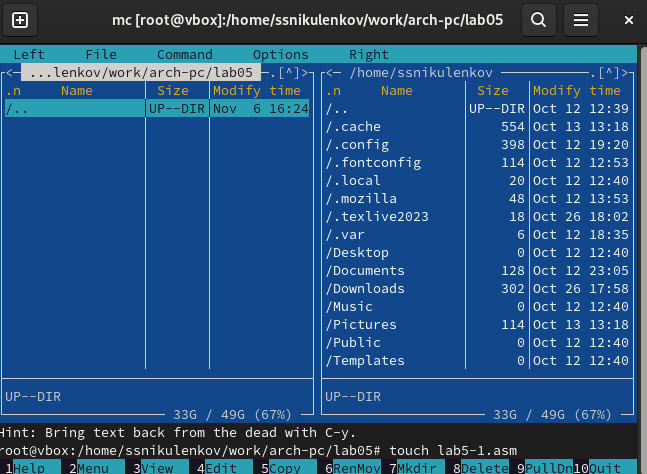


Рис. 2: Создаю lab5-1.asm

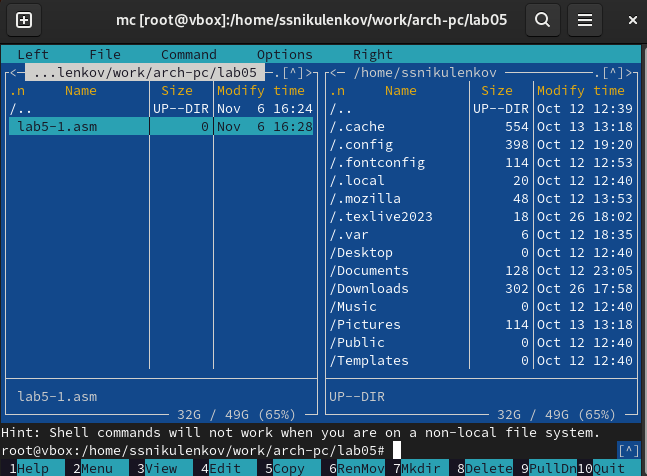


Рис. 3: Файл создан

## 4.2 Структура программы на языке ассемблера NASM

С помощью функциональной клавиши F4 открываю созданный файл для редактирования.

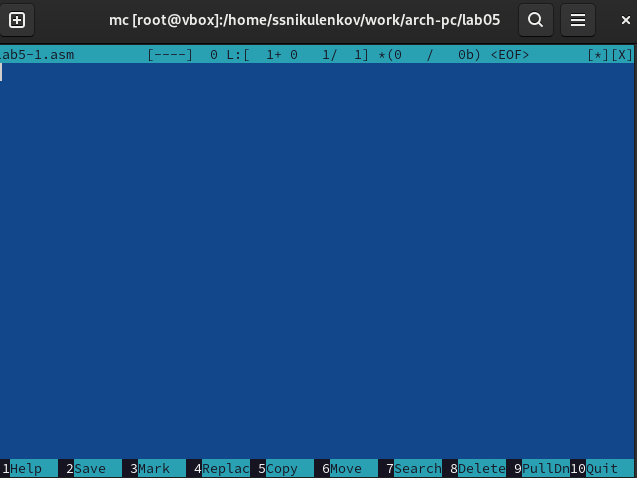


Рис. 4: Файл открыт

Пишу код программы.

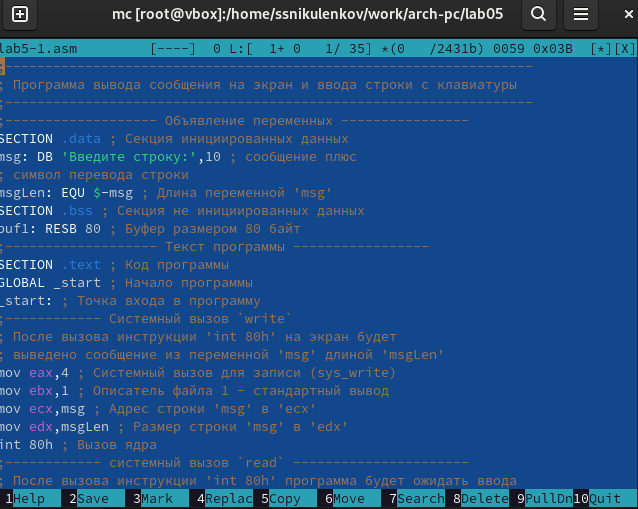


Рис. 5: Код

Открываю файл для просмотра.

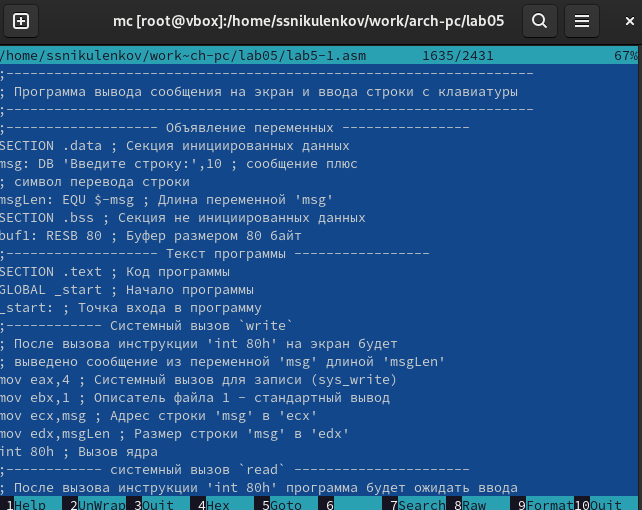


Рис. 6: Проверка

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.Запускаю программу.

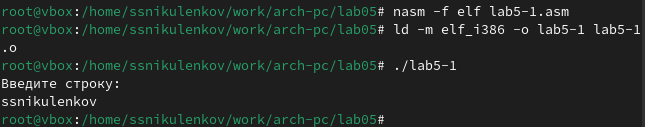


Рис. 7: Исполнение программы

## 4.3 Подключение внешнего файла

Скачиваю файл in\_out.asm со страницы курса в ТУИС. Он сохранился в каталог “Downloads”

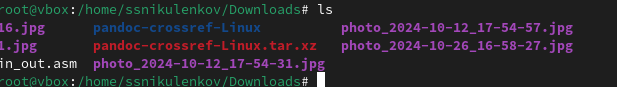


Рис. 8: in\_out.asm в папке downloads

С помощью функциональной клавиши F5 копирую файл in\_out.asm из каталога Downloads в созданный каталог lab05.

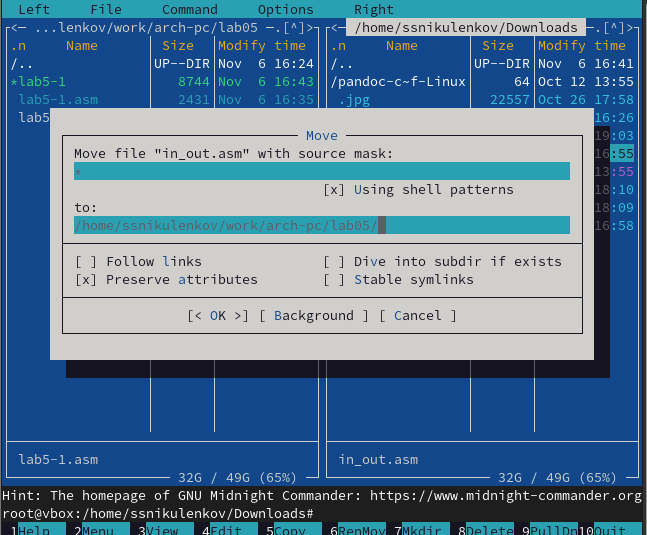


Рис. 9: Копирование в lab05

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

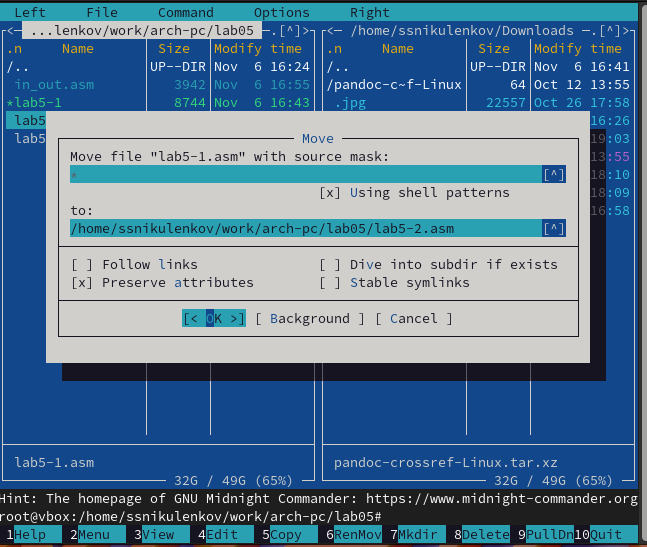


Рис. 10: Переименование файла

Изменяю содержимое файла lab5-2.asm.Создаю объектный файл и исполняю его.

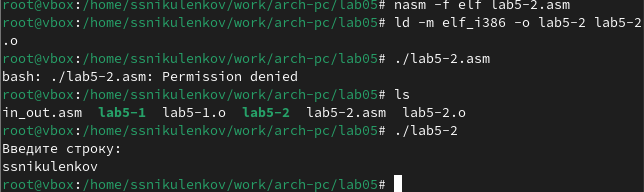


Рис. 11: Исполнение файла

Снова изменяю файл lab5-2.asm.Изменяю в нем подпрограмму sprintLF на sprint.Создаю объектный файл и исполняю его.

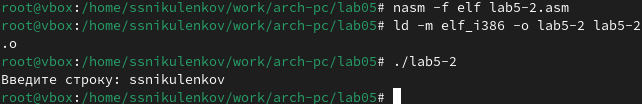


Рис. 12: Исполнение файла

Разница между первым исполняемым файлом lab6-2 и вторым lab6-2-2 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 Выполнение заданий для самостоятельной работы

Создаю копии файлов lab5-1.asm и lab5-2.asm редактирую их.Создаю объектные файлы для каждого из них и исполняю их.

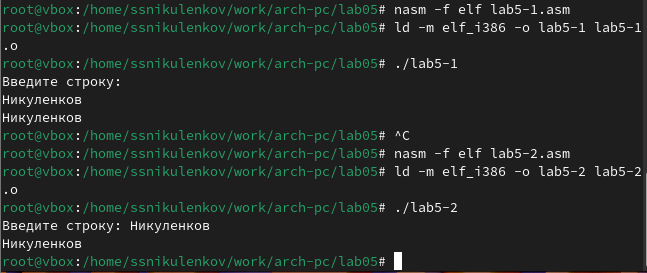


Рис. 13: Исполнени файлов

# 6 Выводы

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