Лабораторная работа №6

Дисциплина: Архитектура компьютера

Алиева Милена Арифовна

Содержание

# 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.

# 4 Выполнение лабораторной работы

1. Открываю Midnight Commander, введя в терминал mc (рис. 1)

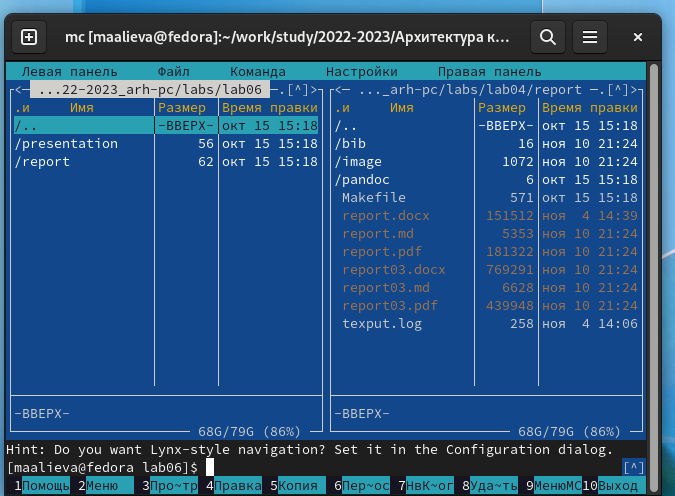


Рис. 1: Открытие Midnight Commander

1. Создаю каталог lab06\_nasm (рис. 2)

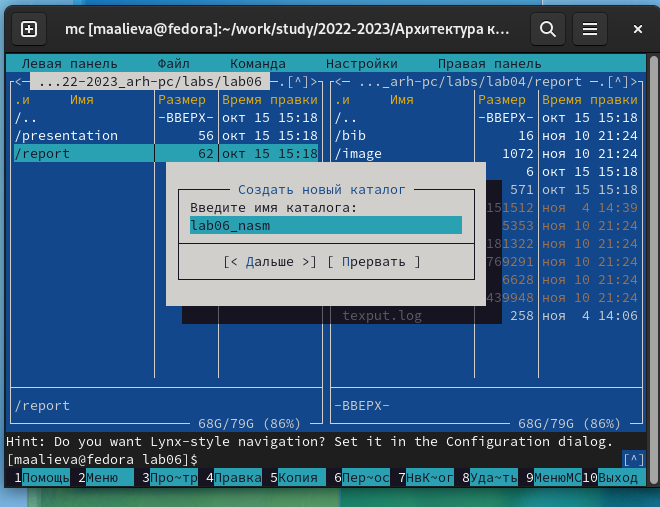


Рис. 2: Создание каталога lab06\_nasm

1. С помощью команды touch lab6-1.asm создаю файл, в котором буду работать. С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Ввожу в файл код программы для запроса строки у пользователя (рис. 3)

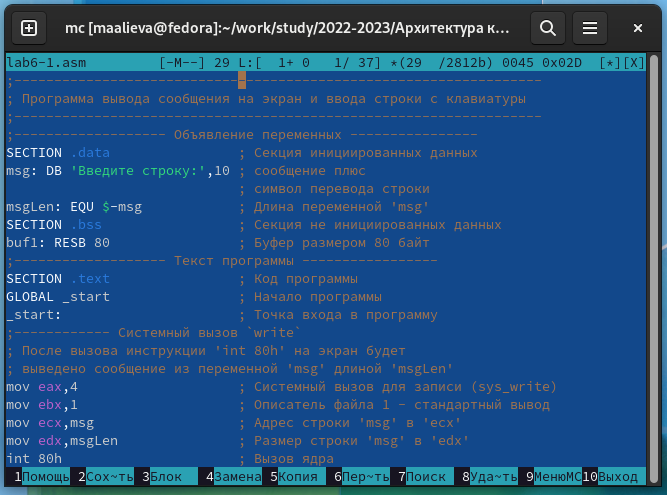


Рис. 3: Код программы для запроса строки у пользователя

1. Транслирую текст программы файла в объектный файл командой nasm -f elf lab6-1.asm, получаю объектный файл lab6-1.o, далее выполняю компоновку объектного файла с помощью команды ld -m elf\_i386 -o lab6-1 lab6-1.o, получаю исполняемый файл lab6-1 (рис. 4)

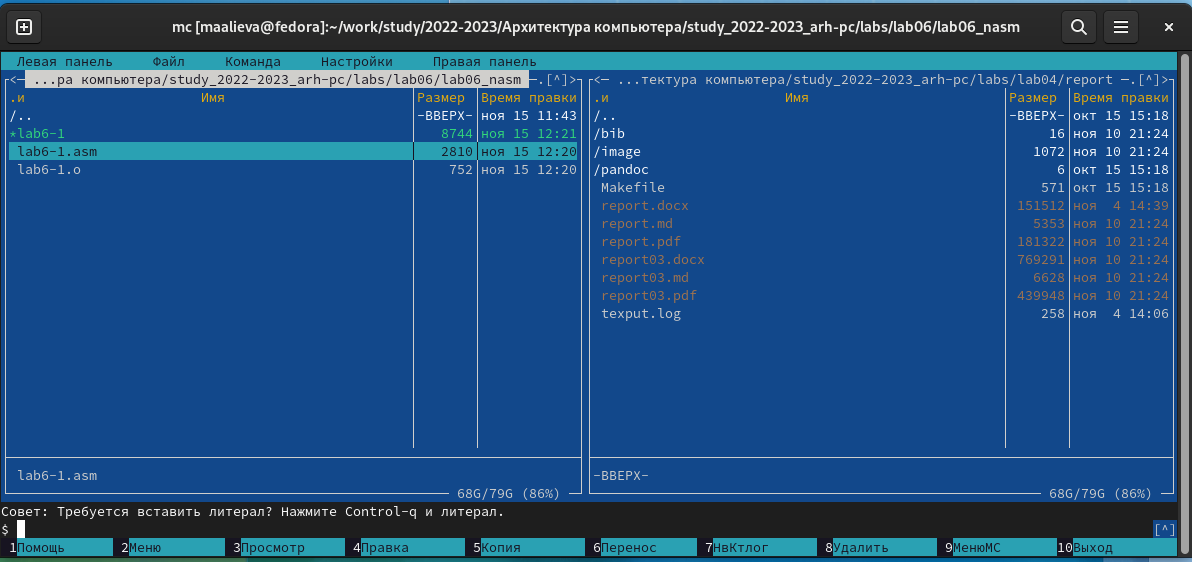


Рис. 4: Созданные файлы lab6-1.o и lab6-1

1. Запускаю исполняемый файл, программа просит ввести строку, ввожу имя и фамилию (рис. 5)



Рис. 5: Исполнение файла, ввод имени и фамилии

1. 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 lab06\_nasm (рис. 6)

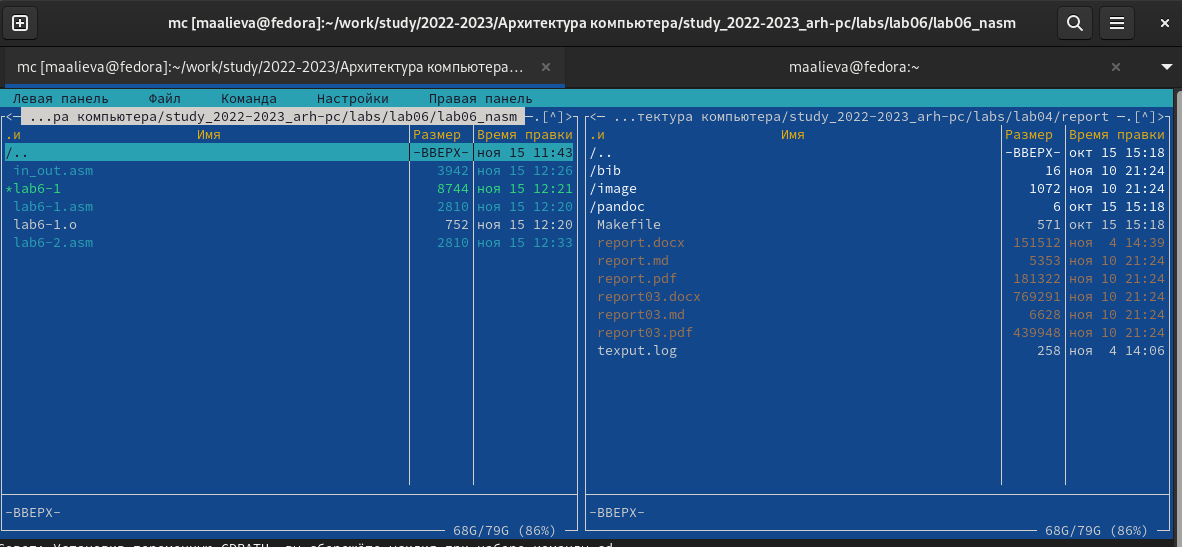


Рис. 6: Скачанный файл in\_out.asm

1. С помощью функциональной клавиши F6 создаю копию файла lab6-1.asm с именем lab6-2.asm. Исправляю текст программы в файле lab6-2.asm с использование подпрограмм из внешнего файла in\_out.asm (использую подпрограммы sprintLF, sread и quit) (рис. 7)

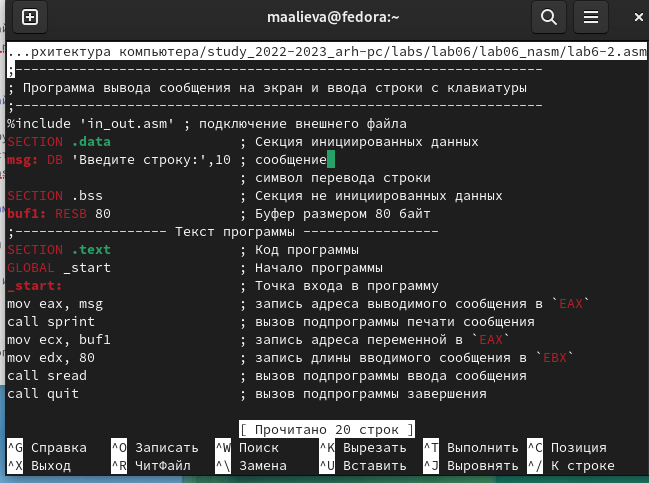


Рис. 7: Текст программы в файле lab6-2.asm

1. Транслирую текст программы файла в объектный файл командой nasm -f elf lab6-2.asm, получаю объектный файл lab6-2.o, далее выполняю компоновку объектного файла с помощью команды ld -m elf\_i386 -o lab6-2 lab6-2.o, получаю исполняемый файл lab6-2 (рис. 8)

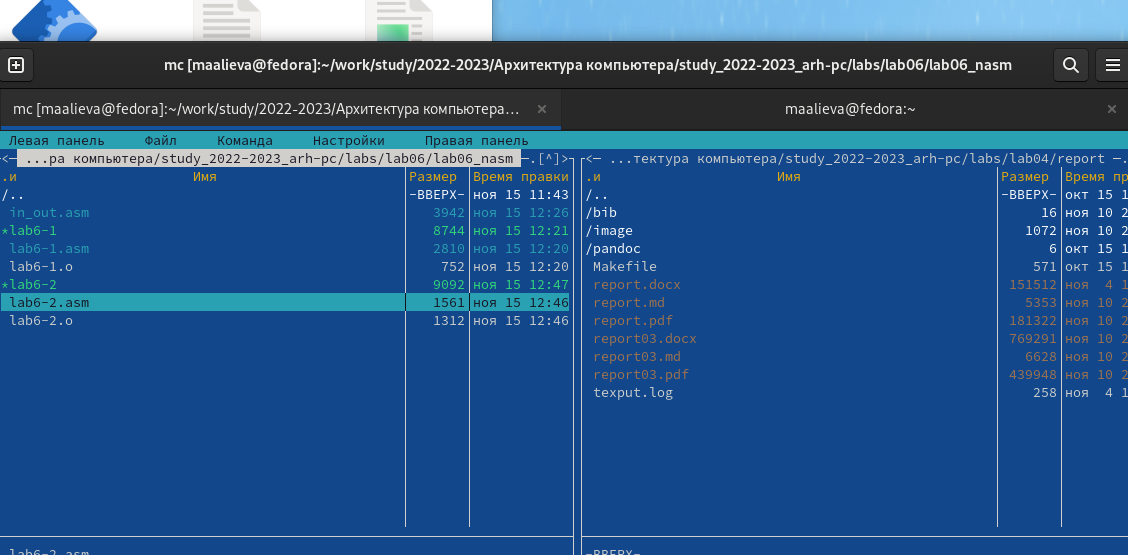


Рис. 8: Созданные файлы lab6-2.o и lab6-2

1. Запускаю исполняемый файл, программа просит ввести строку, ввожу имя и фамилию (рис. 9)

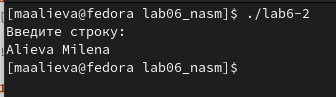


Рис. 9: Исполнение файла, ввод имени и фамилии

1. Начинаю выполнять задания для самостоятельной работы. Создаю копию файла lab6-1.asm с именем lab6-1-1.asm с помощью функциональной клавиши F5 (рис. 10)

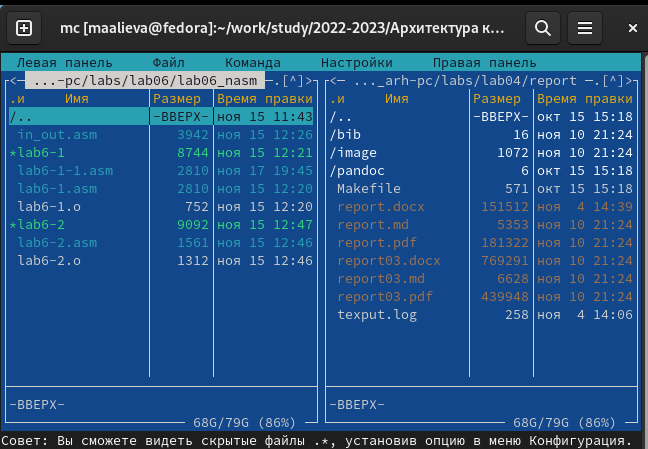


Рис. 10: Созданный файл с именем lab6-1-1.asm

1. С помощью функциональной клавиши F4 открываю созданный файл и изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. 11)

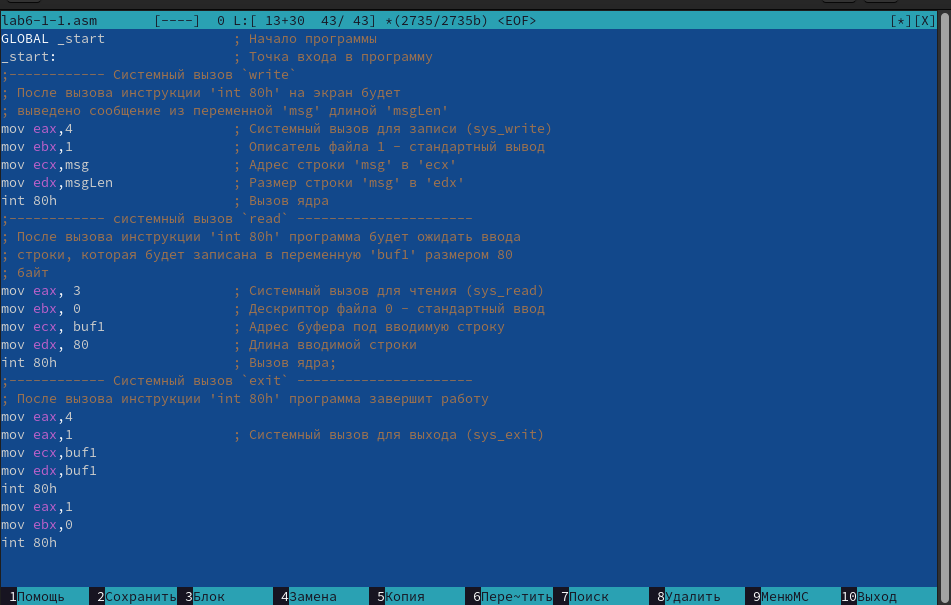


Рис. 11: Изменённая программа

1. Транслирую текст программы файла в объектный файл командой nasm -f elf lab6-1-1.asm, получаю объектный файл lab6-1-1.o, далее выполняю компоновку объектного файла с помощью команды ld -m elf\_i386 -o lab6-1-1 lab6-1-1.o, получаю исполняемый файл lab6-1-1. Программа просит ввести строку, ввожу свои ФИО, далее программа выводит введенные данные (рис. 12)



Рис. 12: Исполнение файла, ввод имени, фамилии, отчества и их вывод

1. Создаю копию файла lab6-2.asm с именем lab6-2-1.asm с помощью функциональной клавиши F5 (рис. 13)

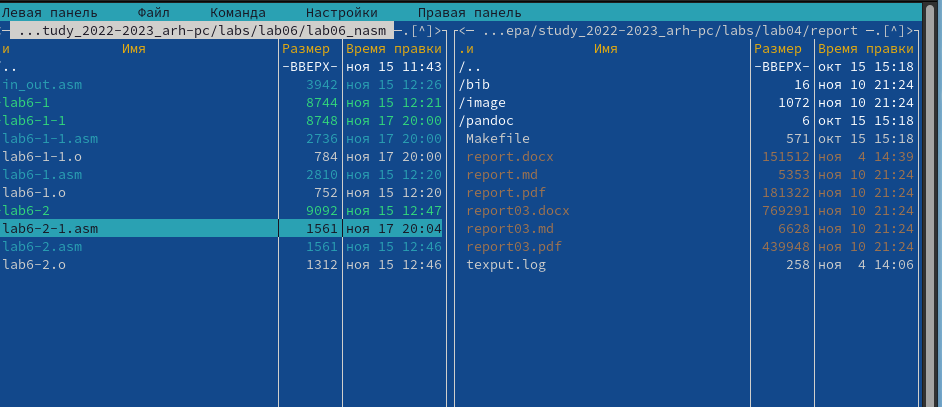


Рис. 13: Создание файла lab6-2-1.asm

1. С помощью функциональной клавиши F4 открываю созданный файл и изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. 14)

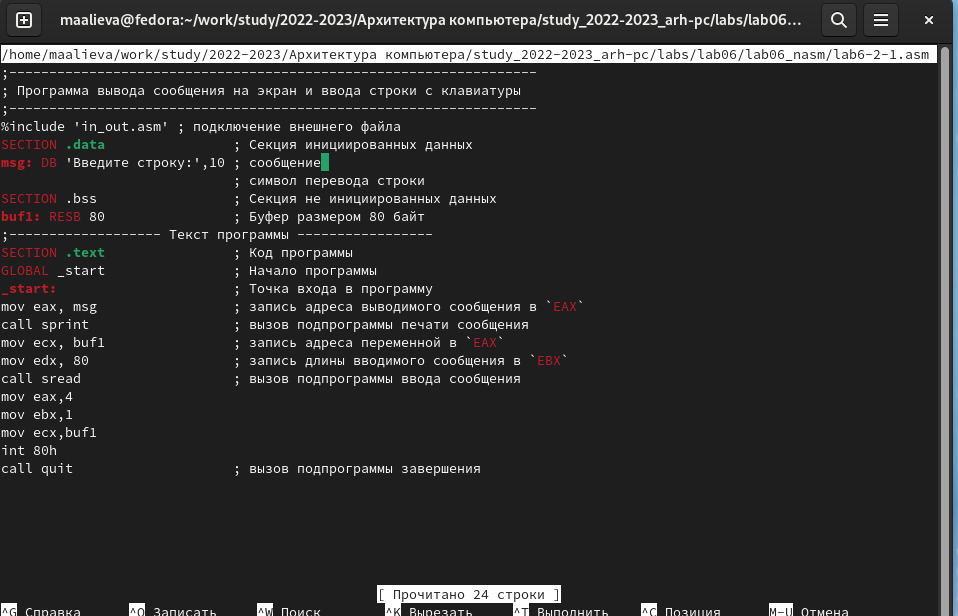


Рис. 14: Изменённый код программы

1. Транслирую текст программы файла в объектный файл командой nasm -f elf lab6-2-1.asm, получаю объектный файл lab6-2-1.o, далее выполняю компоновку объектного файла с помощью команды ld -m elf\_i386 -o lab6-2-1 lab6-2-1.o, получаю исполняемый файл lab6-2-1. Программа просит ввести строку, ввожу свои ФИО, далее программа выводит введенные данные (рис. 15)

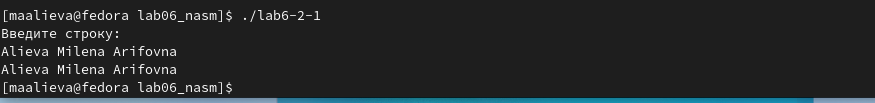


Рис. 15: Исполнение файла, ввод имени, фамилии, отчества и их вывод

# 5 Выводы

Я приобрела практические навыки работы в Midnight Commander, а также освоила инструкции языка ассемблера mov и int.

# Список литературы