Лабораторная работа №6

НКАбд-03-22

Шубнякова Дарья

Содержание

# 1 Цель работы

Приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера mov и int.

# 2 Задание

Создать два исполняемых файла. Один с использованием подпрограмм из внешнего файла, второй – без.

# 3 Теоретическое введение

Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управ- лению 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 неинициализирован- ных 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 ди- рективы DB, DW, DD, DQ и DT, которые резервируют память и указывают, какие значения должны храниться в этой памяти. 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. Для объявления неинициированных данных в секции .bss используются директивы resb, resw, resd и другие, которые сообщают ассемблеру, что необходимо зарезервировать заданное количество ячеек памяти.

Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике.

Инструкция языка ассемблера intпредназначена для вызова прерывания с указанным номером. В общем виде она записывается в виде int n.

Простейший диалог с пользователем требует наличия двух функций — выво- да текста на экран и ввода текста с клавиатуры. Простейший способ вывести строку на экран — использовать системный вызов write. Этот системный вызов имеет номер 4, поэтому перед вызовом инструкции int необходимо поместить значение 4 в регистр eax. Первым аргументом write, помещаемым в регистр ebx, задаётся дескриптор файла. Для вывода на экран в качестве дескриптора файла нужно указать 1 (это означает «стандартный вывод», т. е. вывод на экран). Вторым аргументом задаётся адрес выводимой строки (помещаем его в регистр ecx, например, инструкцией mov ecx, msg). Строка может иметь любую длину. Последним аргументом (т.е. в регистре edx) должна задаваться максимальная длина выводимой строки. Для ввода строки с клавиатуры можно использовать аналогичный системный вызов read. Его аргументы – такие же, как у вызова write, только для «чтения» с клавиатуры используется файловый дескриптор 0 (стандартный ввод). Системный вызов exit является обязательным в конце любой программы на языке ассемблер. Для обозначения конца программы перед вызовом инструк- ции int 80h необходимо поместить в регистр еах значение 1, а в регистр ebx код завершения 0.

# 4 Выполнение лабораторной работы

Открываем Midnight Commander с помощью команды mcedit. Создаем папку lab06, а в ней файл lab6-1.asm с помощью команды touch. (рис. 1)

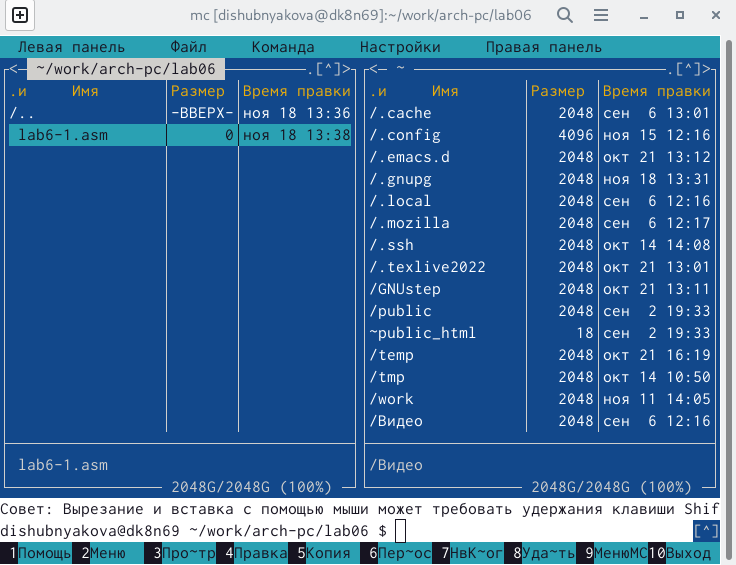


Рис. 1: Начало работы с Midnight Commander

Редактируем файл в соответствии с заданием в редакторе mcedit. Сохраняем файл и закрываем редактор. С помощью клавиши F3 проверяем правильность набранной программы. (рис. 2)

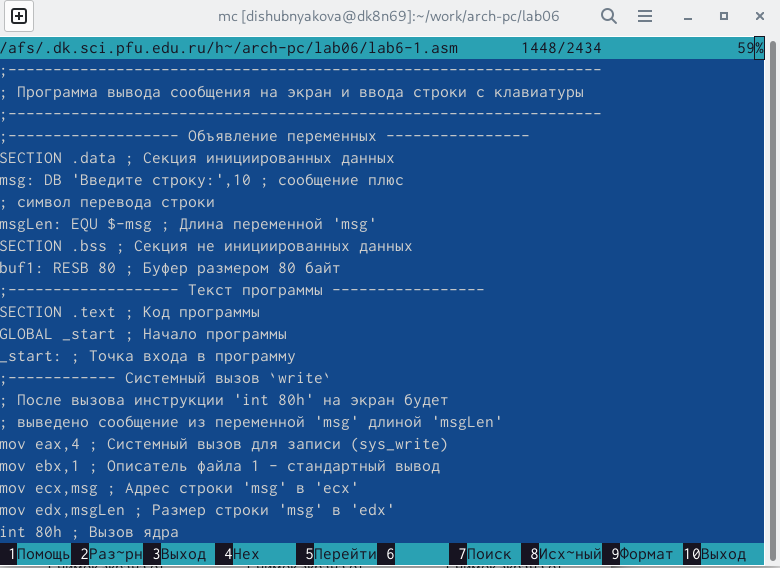


Рис. 2: Редактор mcedit

После того, как мы оттранслировали и провели компановку объектного файла, запускаем исполняемый файл. На запрос “Введите строку” вводим имя и фамилию. (рис. 3)

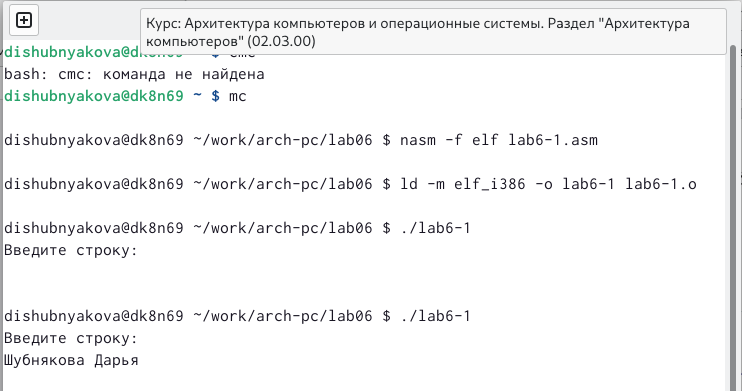


Рис. 3: Запуск файла lab6-1

Скачиваем внешний файл in\_out.asm с подпрограммами и копируем его в наш рабочий каталог. (рис. 4)

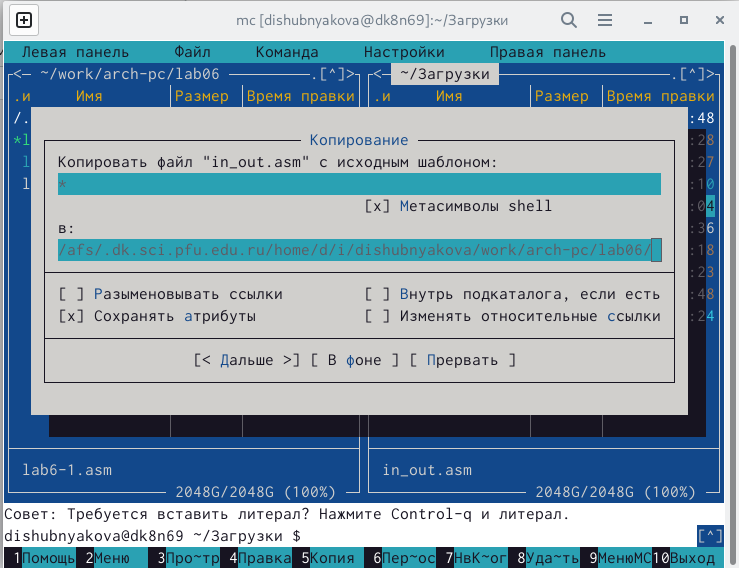


Рис. 4: Внешний файл с подпрограммами

Создаем копию нашего файла с названием lab6-2.asm. (рис. 5)

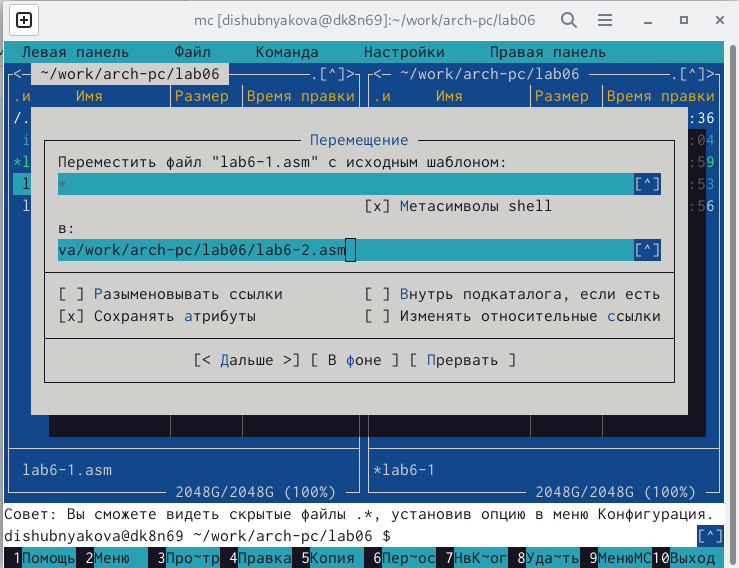


Рис. 5: Работа в Midnight Commander

Исправляем текст программы в nano на новый с использованием подпрограмм из скачанного ранее файла. Сохраним, оттранлируем и скомпонуем.(рис. 6)

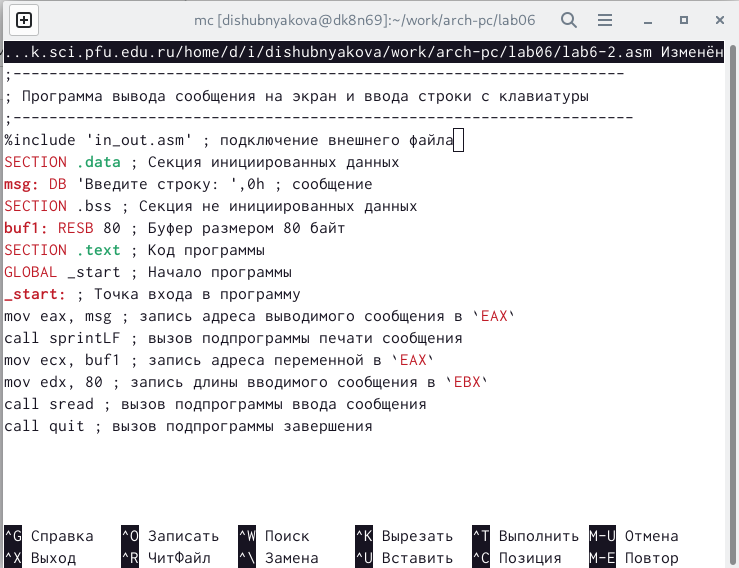


Рис. 6: Редактор nano

Запускаем исполняемый файл и проверяем, что все правильно работает. (рис. 7)

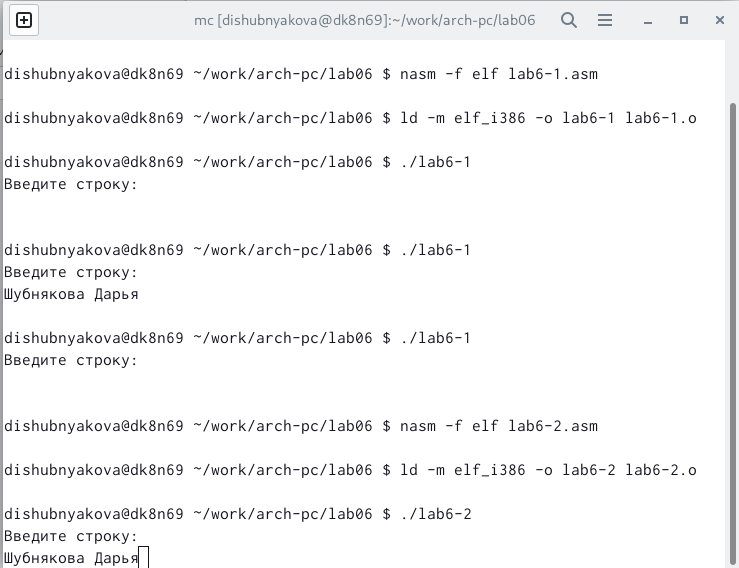


Рис. 7: Запуск lab6-2 с подпрограммой sprintLF

В этом файле меняем подпрограмму sprintLF на sprint. Выполняем те же действия для получения исполняемого файла и проверяем его работу. Замечаем небольшую разницу между двумя этими подпрограммами. При использовании sprintLF после приглашения “Введите строку” был совершен переход на новую строку для пользовательского ввода данных. В случае sprint мы вводим имя и фамилию на этой же строке. (рис. 8)

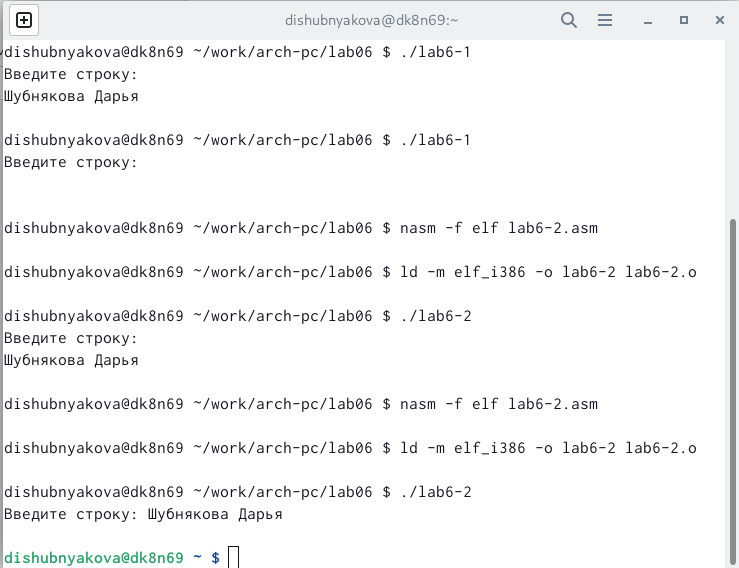


Рис. 8: Запуск lab6-2 с подпрограммой sprint

# 5 Выводы

Мы познакомились с Midnight Commander (или просто mc). Узнали про структуру программы на языке ассемблера NASM. Прочитали про инструкции mov и int. Использовали внешний файл с подпрограммами для более удоной работы с NASM.

# Список литературы