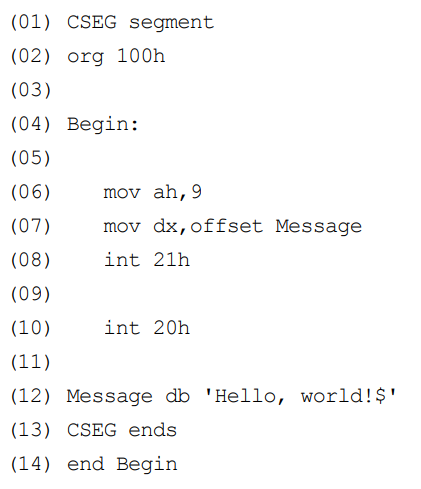
Mov (Move) – присвоить

Int (interrupt) – прерывание

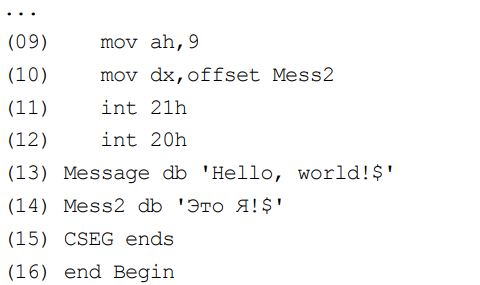


При выполнении строки (08) мы вызываем подпрограмму (в данном случае это называется прерыванием), которая выводит на экран строку. Для этого мы, собственно, и помещаем нужные значения в регистры, т. е. готовим для прерывания необходимые параметры. Всю работу (вывод строки, перемещение курсора) берет на себя эта процедура. Строку (08) следует читать так: "вызываем двадцать первое прерывание" (int от англ. interrupt — прерывание). Обратите внимание, что после числа 21 стоит буква h. Это, как мы уже знаем, шестнадцатеричное число (33 в десятичной системе). Конечно, нам ничего не мешает заменить строку int 21h строкой int 33. Программа будет работать корректно. Но в ассемблере принято указывать номера прерываний в шестнадцатеричной системе, да и все отладчики работают с этой системой.

В строке (10) мы, как вы уже догадались, вызываем прерывание 20h. Для его вызова не нужно указывать какие-либо значения в регистрах. Оно выполняет только одну задачу — выход из программы (выход в DOS). В результате выполнения прерывания 20h программа вернется туда, откуда ее запускали (загружали, вызывали). Например, в Norton Commander или DOS Navigator. Это что-то вроде оператора exit в некоторых языках высокого уровня.

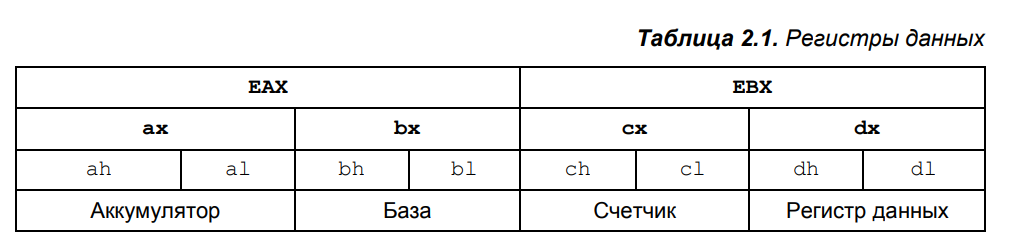
Строка (12) содержит сообщение для вывода. Первое слово (message — сообщение) — название этого сообщения. Оно может быть любым (например, mess или string и пр.). Обратите внимание на строку (07), в которой мы загружаем в регистр dx адрес этого сообщения.

Можно создать еще одну строку, которую назовем Mess2. Затем, начиная со строки (09), вставим в нашу программу следующие команды:



**Регистры**

Регистры данных могут использоваться программистом по своему усмотрению (за исключением некоторых случаев). В них можно хранить любые данные: числа, адреса и пр. В верхнем ряду табл. 2.1 находятся 32-разрядные регистры, которые могут хранить числа от 0 до 4 294 967 295 (0FFFFFFFFh). Их мы будем рассматривать позже. Во втором ряду — 16-разрядные, которые могут хранить числа от 0 до 65 535 или от 0h до FFFFh в шестнадцатеричной системе, что одно и то же. В следующей строке расположен ряд 8-разрядных регистров: ah, al, bh, bl, ch, cl, dh, dl. В эти регистры можно загружать максимальное число 255 (FFh). Это так называемые половинки (старшая или младшая) 16-разрядных регистров.

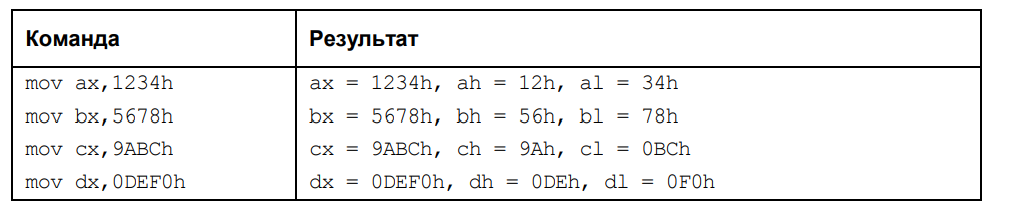


Мы уже изучили оператор mov, который предназначен для загрузки числа в регистр. Чтобы присвоить, к примеру, регистру al число 35h, нам необходимо записать так:

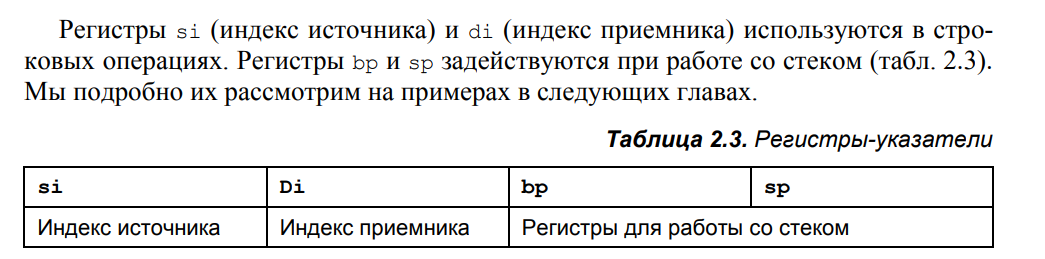


а регистру ax — число 346Ah, так:

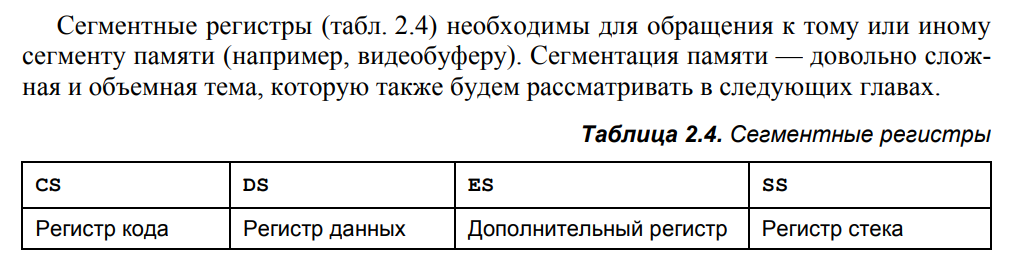




**Регистры-указатели**

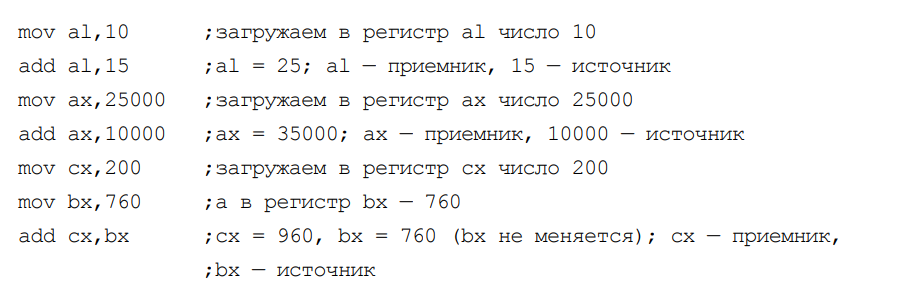


**Сегментные регистры**

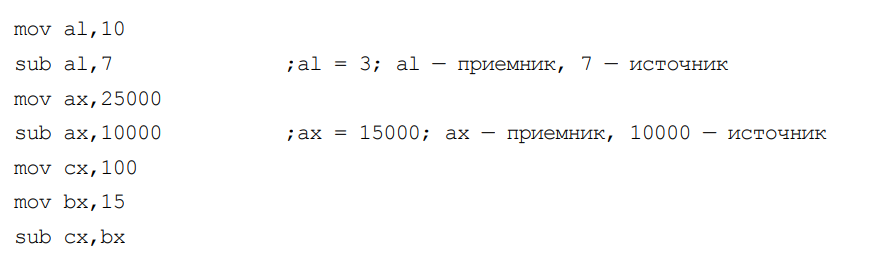


**Команды сложения и вычитания**

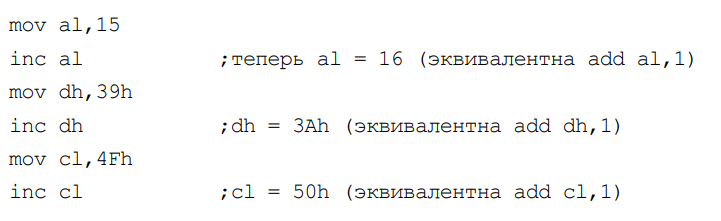
Add (addition) – сложение двух чисел.



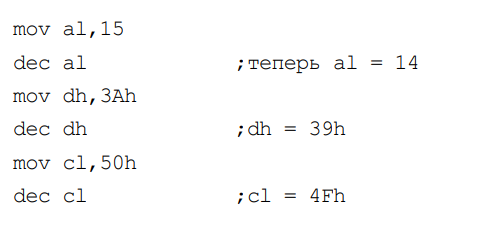
Sub (subtraction) – вычитание двух чисел.



Inc (increment) – увеличивает содержимое приемника (регистра или ячейки памяти) на 1.



Dec (decrement) – уменьшает содержимое приемника (регистра или ячейки памяти) на 1.



Рассмотрим одну небольшую программу, которая выводит на экран сообщение и ждет, когда пользователь нажмет любую клавишу. После чего возвращается в DOS. Работать с клавиатурой позволяет прерывание BIOS (ПЗУ) 16h, которое можно вызывать даже до загрузки операционной системы, в то время как прерывания 20h, 21h и пр. доступны только после загрузки IO.SYS/MSDOS.SYS — определенной части ОС MS-DOS. Чтобы заставить программу ждать нажатия пользователем любой клавиши, следует вызвать функцию 10h прерывания 16h. После нажатия любой клавиши компьютер продолжит выполнять программу, а регистр ax будет содержать код клавиши, которую нажал пользователь. Следующая программа выводит на экран сообщение и ждет нажатия любой клавиши, что равнозначно команде PAUSE в BAT-файлах.

