# LABEL

(стр 107)

Когда метка стоит перед директивой ассемблера, она обычно оказывается одним из операндов этой директивы и двоеточие не ставится. Рассмотрим директивы, работающие напрямую с метками и их значениями, - LABEL, EQU и =.

метка label тип

Директива LABEL определяет метку и задает ее тип: BYTE (байт), WORD

(слово), DWORD (двойное слово), FWORD (6 байт), QWORD (учетверенное слово), TBYTE (10 байт), NEAR (ближняя метка), FAR (дальняя метка). Метка получает значение, равное адресу следующей команды или следующих данных, и тип,

указанный явно. В зависимости от типа команда

mov метка,О

запишет в память байт (слово, двойное слово и т. д.), заполненный нулями, а команда

call метка

выполнит ближний или дальний вызов подпрограммы.

С помощью директивы LABEL удобно организовывать доступ к одним и тем же

данным, как к байтам, так и к словам, определив перед данными две метки с разными типами

# EXTRN и PUBLIC

EXTRN - объявляет соответствующее имя внешним для текущего ассемблирования

PUBLIC, извещающего ассемблер о том, что данное символическое имя доступно другим программам.

<https://www.i-assembler.ru/182/5-10-2.html>

## Процедуры NEAR и FAR

Процедуры **NEAR** вызываются с помощью вызова ближнего типа и содержат ближний возврат управления. Вы должны вызывать их только в том же сегменте, в котором они определены. Вызов ближнего типа заносит адрес возврата в стек и устанавливает указатель инструк- тор (IP) в значение смешения процедуры. Поскольку сегмент кода (CS) не изменяется, процедура должна находиться в том же сегмен- те, что и вызывающая программа. Когда процессор обнаруживает возврат ближнего типа, он извлекает из стека адрес возврата и снова устанавливает в него IP. Сегмент кода не изменяется.

Процедура **FAR** вызывается с помощью вызова дальнего типа и содержит возврат дальнего типа. Процедуры FAR вы можете вызывать вне сегмента, в котором они определяются. Вызов FAR заносит в стек адрес в виде сегмента и смещения, а затем устанавливает CS:IP в адрес процедуры. Когда процессор обнаруживает возврат дальнего типа, он извлекает из стека сегмент и смещение адреса возврата и устанавливает в него CS:IP.

Расстояние (NEAR или FAR), используемое в процедуре по умол- чанию, определяется текущей выбранной моделью. Для моделей TINY, SMALL и COMPACT по умолчанию процедура будет ближней (NEAR). Для всех других моделей по умолчанию выбирается расстояние FAR. Если вы не используете упрощенные директивы определения сегментов, то по умолчанию процедура всегда будет ближней (NEAR).

Индексный регистр SI DI— это регистр процессора в современных процессорах, используемый для автоматического изменения адреса операнда во время исполнения программы.

В архитектуре x86 индексные регистры называются SI и DI. При базово-индексной адресации их содержимое может суммироваться с содержимым регистра BX.

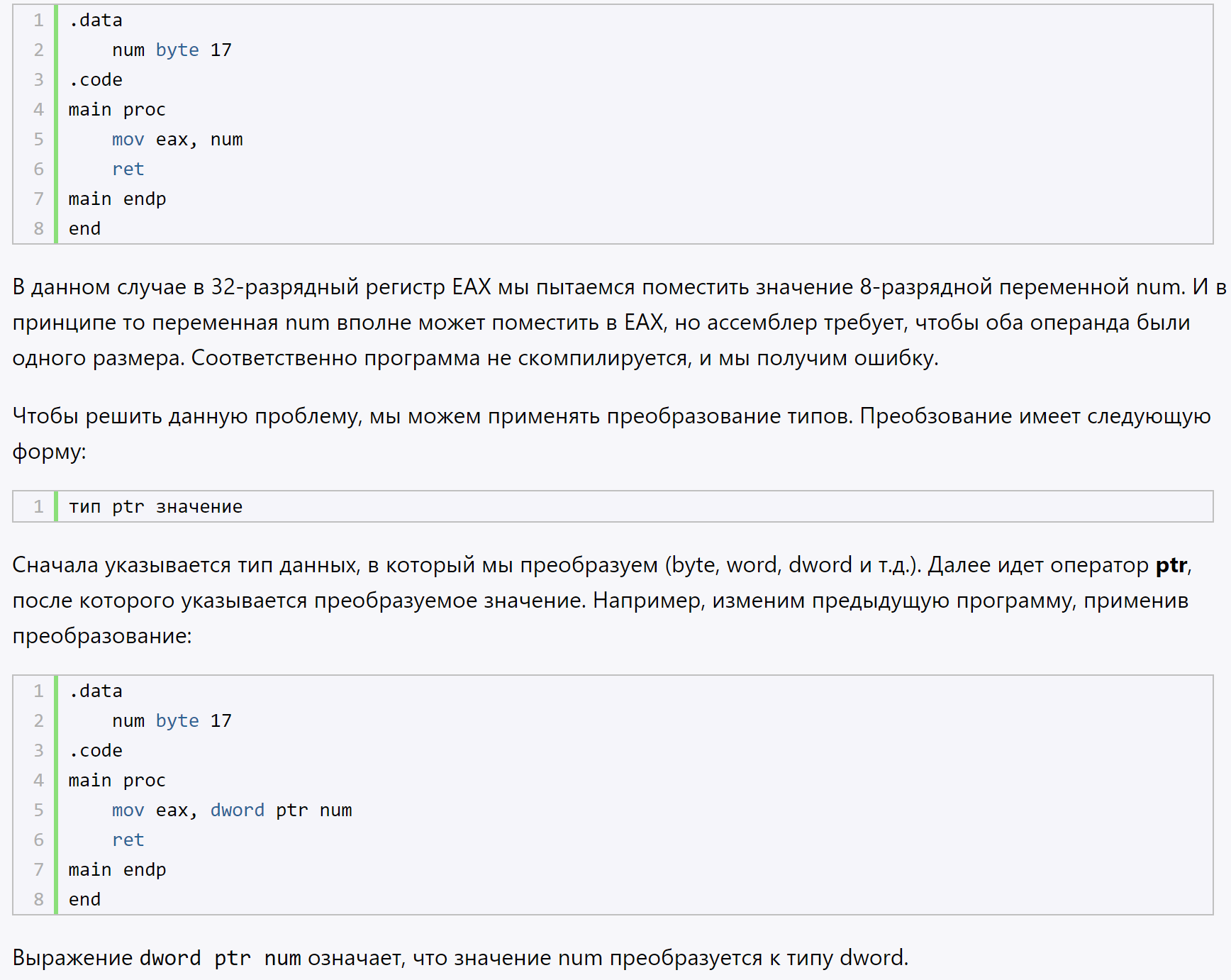
Индексный регистр-источник SI применяется в качестве указателя адреса байта или слова в таких строковых командах, как LODS (загрузить строку), CMPS (сравнить строку), MOVS (переслать строку).

Индексный регистр-приёмник DI используется как указатель назначения для адреса байта или слова в строковых командах, таких как SCAS (сканировать строку), CMPS, MOVS, STOS (записать строку).

Базово-индексная адресация — это распространённый способ адресации, особенно при работе с массивами.

В нём используются два регистра: один из них должен быть базовым (ВХ или ВР), а другой — индексным (SI или DI). Как правило, в одном из регистров находится адрес массива, а в другом — индекс в нём.

# ptr – преобразование типов



<https://metanit.com/assembler/tutorial/3.1.php>

# Резидентные программы

#### **INT 21h функция 35h**

После вызова этой функции в регистрах **ES** и **BX** окажутся, соответственно, сегмент и смещение обработчика прерывания. Это нам пригодится для корректной передачи управления в ДОС после выполнения нашего алгоритма.

ВХОД: **AH** – **35h** **AL** – номер прерывания ВЫХОД: **ES** – сегмент обработчика прерывания **BX** – смещение обработчика прерывания

#### **INT 21h функция 25h**

Эта функция устанавливает новый обработчик прерывания. Пользоваться надо очень осторожно и обдуманно, дабы не отстрелить себе очередную ногу.

ВХОД: **AH** – **25h** **AL** – номер прерывания **DS** – сегмент обработчика прерывания **DX** – смещение обработчика прерывания