

Команды ассемблера и машинные команды

Прежде всего нужно не забывать, что команды на языке ассемблера и команды на машинном языке - это две разные вещи. Хотя несомненно, что эти два понятия тесно связаны между собой.

Команда ассемблера это некоторое мнемоническое имя. Для процессоров семейства x86 это имя записывается на английском языке. Например, у команды сложения имя **ADD**, а у команды вычитания имя **SUB**.

В таблицах настоящего справочника в колонке **Команда** показано имя команды на языке ассемблера.

Основой машинной команды является код операции, который является просто некоторым числом. Для процессоров x86 (впрочем, и для других процессоров тоже) принято использовать шестнадцатиричные числа. (Попутно заметим, что для советских ЭЦВМ были приняты восьмеричные числа, с ними было меньше путаницы, поскольку такие числа состоят только из цифр и не содержат букв).

В таблицах настоящего справочника в колонке **Код** показан код операции машинной команды, а в колонке **Формат** показан формат машинной команды.

Можно считать, что количество разных машинных команд у данного процессора равно количеству возможных кодов операции. По формату можно узнать, из каких составных частей состоит данная машинная команда. Разные машинные команды могут иметь разный формат. Код операции машинной команды полностью определяет ее формат.

Часто одна ассемблерная команда имеет несколько разных вариантов машинных команд. Причем форматы этих машинных команд могут быть разными для разных вариантов.

Например, ассемблерная команда **ADD** имеет десять вариантов машинных команд с разными кодами операций. Но разных форматов оказывается меньше, только три. И каждый из этих трех форматов требует свои типы операндов при записи команды на языке ассемблера.

См. страницу [Основные команды арифметики](#).

Здесь важно обратить внимание, что все эти десять машинных команд выполняют одну и ту же элементарную операцию, которая на языке ассемблера называется **ADD**.

И, значит, получается, что вроде бы можно рассуждать так: процессор может выполнять столько разных элементарных операций, сколько есть разных команд ассемблера. Однако, этот простой принцип все равно нуждается в оговорках и примечаниях. Так как у части ассемблерных команд бывают еще и синонимы.

Общий список всех команд процессора можно построить разными способами, выбирая разный порядок расположения команд. Основные два способа такие.

Способ (1). Взять за основу команды языка ассемблера и расположить команды по алфавиту.

Тогда могут получиться вот такие таблицы.

[Все команды по алфавиту \(кратко\)](#)

[Все команды по алфавиту \(подробно\)](#)

Способ (2). Взять за основу код операции машинной команды и расположить команды в порядке кодов операций. При этом будет лучше, если общий список разделить на две части, сделать отдельные списки для команд с однобайтным кодом операции и для команд с двухбайтным кодом операции.

[Первый байт кода операций](#)

[Второй байт кода операций](#)

Конечно, возможен еще и третий способ, который обычно применяется в учебниках. Разделить все команды на группы по смыслу и изучать их по группам, начиная с более простых.

Основной байт кода операции

В системе команд x86 одного байта (256 разных комбинаций) оказалось недостаточно для кодирования всех команд. Поэтому код операции в машинной команде занимает либо один байт, либо два байта.

Если первый байт содержит код **0F**, то код операции состоит из двух байтов.

Если в машинной команде код операции состоит из одного байта, то этот единственный байт и является основным байтом кода операции. И содержимое этого байта определяет, что это за операция.

Если в машинной команде код операции состоит из двух байт, то уже не первый, а второй байт будет основным и определяющим в коде операции.

В таблицах справочника, в которых показано кодирование машинных команд, основной байт кода операции обычно бывает показан дважды, сначала в колонке "Код" в виде шестнадцатиричного числа, а затем в колонке "Формат" в виде условных восьми черточек, на которых отмечены особые биты, если таковые есть в основном байте кода операции.