Процедуры

процедура

Процедура представляет собой код, который может выполняться многократно и к которому можно обращаться из разных частей программы. Обычно процедуры предназначены для выполнения каких-то отдельных, законченных действий программы и поэтому их иногда называют подпрограммами.

Команды CALL и RET

Для работы с процедурами предназначены команды CALL и RET. С помощью команды CALL выполняется *вызов* процедуры. Эта команда работает почти также, как команда безусловного перехода (JMP), но с одним отличием — одновременно в стек сохраняется текущее значение регистра IP. Это позволяет потом вернуться к тому месту в коде, откуда была вызвана процедура. В качестве операнда указывается адрес перехода, который может быть непосредственным значением (меткой), 16-разрядным регистром (кроме сегментных) или ячейкой памяти, содержащей адрес.

ret

Возврат из процедуры выполняется командой RET. Эта команда восстанавливает значение из вершины стека в регистр IP. Таким образом, выполнение программы продолжается с команды, следующей сразу после команды CALL. Обычно код процедуры заканчивается этой командой. Команды CALL и RET не изменяют значения флагов .

Помещение параметров в стек

Перед вызовом процедуры параметры необходимо поместить в стек с помощью команды PUSH. Обычно используется обратный порядок. Параметры помещаются в стек, начиная с последнего, так что перед вызовом процедуры на вершине стека оказывается первый параметр:

Обращение к параметрам внутри процедуры

Для обращения к параметрам внутри процедуры обычно используют регистр BP. В самом начале процедуры содержимое регистра BP сохраняется в стеке и в него копируется значение регистра SP. Это позволяет «запомнить» положение вершины стека и адресовать параметры относительно регистра BP.

Извлечение параметров из стека

После того, как процедура выполнилась, необходимо очистить стек, вытолкнув из него параметры. Тут тоже существует 2 способа: стек может быть очищен самой процедурой или кодом, который эту процедуру вызывал. Для первого способа используется команда RET с одним операндом, который должен быть равен количеству байтов, выталкиваемых из стека. В нашем случае он должен быть равен количеству параметров, умноженному на 2.

регистры

Регистры-указатели BP и SP используются для работы со стеком. BP (*Base Pointer*) позволяет работать с переменными в стеке. SP (*Stack Pointer*) указывает на вершину стека. Он используется командами, которые работают со стеком.

Указатель команд

Указатель команд IP (*Instruction Pointer*) содержит адрес команды (в сегменте кода). Напрямую изменять его содержимое нельзя, но процессор делает это сам. При выполнении обычных команд значение IP увеличивается на размер выполненной команды. Существуют также команды передачи управления, которые изменяют значение IP для осуществления переходов внутри программы.

NEG

- Если необходимо в программе поменять знак числа на противоположный, можно использовать команду NEG. У этой команды всего один операнд.
- Neg ax