

18.18.1 Охарактеризувати процес обробки макрокоманд компілятором Асемблера
Макросы делятся на макроопределения (macro definition или просто macro) и макровыводы (macro instruction или macrocode).

Признаком макроопределения является директива MACRO, записанная в его первой строке. Перед ней обязательно указывается имя макроса, а после нее - список параметров, если они используются. В последующих строках располагается тело макроопределения, структура которого зависит от назначения макроса. В общем случае тело содержит директивы MASM и ассемблерные команды или их заготовки. В заготовках команд вместо имен операндов или операций указываются имена параметров макроопределения. Тело макроса обязательно заканчивается директивой ENDM.

Признаком макровывода является имя макроопределения и список параметров, если они используются. В процессе компиляции, обнаружив любое имя, MASM ищет его в таблицах имен, описанных в компилируемой программе и в своих таблицах, зарезервированных имен. Если имя соответствует макроопределению, то выполняется макроподстановка (macro substitution). При этом фрагменты ассемблерного кода, содержащиеся в теле макроопределения, включаются в текст программы, сразу компилируются и при отсутствии ошибок передаются в формируемый объектный модуль. В таких случаях макроподстановку иногда называют макрорасширением (macro expansion). Если же тело макроопределения не содержит ассемблерного кода, то в тексте программы и в объектном коде не остается никаких следов от его выполнения (пример 6 данной статьи).

По аналогии с подпрограммами можно считать, что в макроопределении описаны формальные параметры, а в макровыводе - фактические. Как обычно, должно соблюдаться строгое соответствие количества, типов и последовательности перечисления формальных и фактических параметров. Параметры, отсутствующие в макровыводе, заменяются двумя подряд расположенными запятыми. Фактический параметр может отсутствовать, только если это предусмотрено при разработке макроопределения, т. е. перед использованием параметра проверяется, определен он или нет (указан или отсутствует).

По функциональному назначению макроопределения можно разделить на две основные категории:

средства для упрощения программирования на ассемблере;
средства для расширения возможностей компилятора.

Принципиальное различие между ними заключается в следующем. Вызов макросов первой категории можно заменить группой команд, получаемых в результате подстановки фактических параметров. Макросы второй группы ничем заменить нельзя, поскольку они формируют объектные коды таких команд, который не обрабатывает конкретная версия компилятора.

Макроопределения первой категории чаще всего применяются для упрощения программирования запросов функций, выполняемых операционными системами и базовой системой ввода-вывода (BIOS). Разработчики MASM включают в состав дистрибутивных пакетов файлы, содержащие такие макроопределения, их можно (и нужно) использовать в прикладных программах различного назначения.

В дистрибутивном пакете MASM 6.0 имеется каталог INCLUDE, в котором собраны файлы с макроопределениями различного назначения. Например, файлы bios.inc и dos.inc содержат макроопределения запросов функций BIOS и DOS, соответственно, а файл macros.inc содержит макросы, выполняющие различные вспомогательные действия, в том числе и

обработку строк текста. В том же каталоге находятся файлы с другими макроопределениями.

В дистрибутивном пакете MASM32 имеется каталог MACROS, в котором находятся файлы `usemasco.inc` и `macros.asm`, содержащие макроопределениями запросов функций, выполняемых 32-х разрядными операционными системами семейства WINDOWS. Если запросы функций BIOS и DOS программируются сравнительно просто, то программировать запросы системных функций WINDOWS, без использования готовых макроопределений просто нецелесообразно.

Дистрибутивные пакеты содержат не только примеры макроопределений, но и описания зарезервированных имен, и исходные тексты программ выполняющих много полезных действий. Их анализ позволит вам найти ответ на многие вопросы, связанные с использованием системных функций, библиотек различного назначения и пр.

Макросредства, предназначенные для расширения возможностей компилятора, стали использоваться сравнительно недавно. Необходимость в них возникла потому, что система команд микропроцессоров семейства Pentium развивалась слишком быстро. Появлялись десятки и сотни принципиально новых команд, которые не могли компилировать существующие версии MASM. В этой ситуации сами разработчики стали публиковать макроопределения, позволяющие использовать новые команды при работе с устаревшими версиями MASM.

В заключение данного раздела следует отметить, что разработка макроопределений имеет мало общего с программированием на языке ассемблера. В этом случае программист решает совершенно другие задачи и использует другие средства. В двух следующих разделах данной статьи описаны некоторые приемы, применяемые при разработке макросов различного назначения.