

Глава 1

Предисловие

Знание языка ассемблера поможет программисту не только научиться программировать, но также лучше понять взаимодействие языков высокого уровня с аппаратным обеспечением, что в конечном итоге приводит к общему пониманию всего процесса выполнения программы на компьютере и способствует более осмысленному использованию конструкций языка высокого уровня. Язык ассемблера помогает раскрыть все секреты аппаратного и программного обеспечения, с его помощью можно получить представление о том, как аппаратная часть взаимодействует с операционной системой и как прикладные программы обращаются к операционной системе. Язык ассемблера — это язык низкого уровня, в котором каждая команда непосредственно интерпретируется в команду процессора. Поэтому изучение языка ассемблера позволит понять все нюансы обработки информации в процессоре и поможет программисту в управлении процессором и операционной системой, а также получении прямого доступа к аппаратуре.

Язык ассемблера незаменим при оптимизации критических блоков в прикладных программах с целью повышения их быстродействия. Например, многие фрагменты исходных кодов языка Delphi написаны именно на языке ассемблера, что и определяет быстродействие программ, разработанных на языке Delphi.

Язык ассемблера используется также при изучении курса аппаратного обеспечения компьютера и операционных систем, когда есть возможность проводить экспериментальную работу.

Возможно, вам придется создавать обслуживающие программы (утилиты) или писать драйверы для отдельных устройств. Во многих случаях это невозможно реализовать на языках высокого уровня и только язык ассемблера позволит это сделать.

Что такое язык ассемблера

Язык ассемблера — это специфический язык программирования со взаимно однозначным соответствием между его операторами и командами процессора. Язык ассемблера существует для каждого типа процессоров или целого семейства процессоров. Это связано с тем, что команды на языке ассемблера должны иметь взаимно однозначное соответствие с системой машинных команд и должны быть согласованы с архитектурой компьютера. Язык ассемблера для процессоров Intel имеет систему команд, работать с которыми могут различные процессоры производства Intel, такие как 8086, 8088, 80186, 80286, 80386, 80486, Celeron и Pentium.

Ассемблер — это программа, преобразовывающая исходные коды языка ассемблера в машинные команды. Ассемблер может создавать листинг программы с номерами строк, адресами памяти, исходными операторами и таблицей перекрестных ссылок символов и переменных, используемых в программе. Совместно используемая с ассемблером программа, называемая *компоновщик*, собирает отдельные файлы, созданные ассемблером, в единую исполняемую программу. В блок базовых программ входит также *отладчик*, позволяющий программисту проверять и трассировать исполняемую программу и контролировать содержимое памяти. Программы на языке ассемблера можно запускать либо в операционной

системе MS-DOS (или в режиме эмуляции MS-DOS в системах Windows), либо непосредственно в Windows. Наиболее популярны для семейства процессоров Intel ассемблеры MASM (Microsoft Assembler) и TASM (Borland Turbo Assembler). Язык ассемблера называется языком низкого уровня потому, что он очень близок к машинному языку по своей структуре и выполняемым функциям. Каждая команда языка ассемблера имеет взаимно однозначное соответствие с машинными командами, что отличает его от языков высокого уровня, для которых характерно то, что один оператор транслируется во множество машинных команд.

Прикладные программы на языке ассемблера

Может сложиться впечатление, что программы, написанные на языке ассемблера, не требуют особых усилий при их разработке — необходимо только освоить небольшое количество исходных команд и можно реализовывать логику программы. Однако даже при реализации простых программ следует вручную контролировать довольно много элементов, которые в языке высокого уровня контролируются автоматически. Язык ассемблера требует особого внимания к деталям и поэтому написание и отладка программы занимают очень много времени, но при этом получаются небольшие выполняемые модули, занимающие мало места в памяти и позволяющие достичь приемлемой скорости работы даже на медленных компьютерах. Значительные трудозатраты при написании программ на языке ассемблера не позволяют его использовать для создания больших и сложных программ, для этих целей существуют языки программирования высокого уровня, с помощью которых можно создавать структурированные программы, облегчающие их написание и отладку. На современном этапе развития такие объектно-ориентированные языки, как C++, Java, C# или Delphi, стали основным инструментом написания огромных программ, включающих миллионы строк кода.

Язык ассемблера в основном используется при написании отдельных сегментов прикладных программ (для повышения скорости работы и прямого доступа к оборудованию), а также встроенных системных программ, которые хранятся в программируемой памяти отдельных устройств.

Некоторые языки высокого уровня, например C++ или Delphi, имеют уникальную способность совмещения языковых структур высокого уровня с низкоуровневыми деталями. При этом возможен прямой доступ к аппаратуре, хотя в результате может получиться *непереносимая программа* (программа, не работающая на других типах компьютеров, отличных от того, для которого она разрабатывалась). Компиляторы этих языков имеют возможность создавать ассемблерный код, с которым может работать программист перед тем, как получить исполняемый код.

Совмещение языков высокого уровня и языка ассемблера обычно используется при разработке приложений для встроенных систем, таких как картриджи для компьютерных игр, микроконтроллеры, телекоммуникационное оборудование, системы контроля окружающей среды, промышленные роботы, системы безопасности и др.

Можно написать подпрограммы на языке ассемблера и вызывать их из программ, написанных на языках высокого уровня. В отдельных случаях такие подпрограммы называют *интерфейсными программами*, поскольку они обеспечивают взаимодействие с оборудованием компьютера. В компьютерных играх, например, интерфейсные программы могут обеспечивать прямой доступ к аппаратным портам видеоадаптера и звуковой плате для достижения высочайшего уровня производительности.

Подпрограммы на языке ассемблера, вызываемые операционной системой для работы с отдельными устройствами, называют *драйверами устройств*. Как правило, драйверы обслуживают все прикладные программы. Например, когда программе необходимо считать

блок данных из файла, операционная система вызывает подпрограмму драйвера для устройства считывания.

Программируя на языке ассемблера, необходимо абсолютно точно представлять распределение данных, иначе ошибки неизбежны. Языки высокого уровня скрывают от программиста специфические детали для удобства использования и получения переносимого кода. С другой стороны, используя язык ассемблера, можно учитывать специфику отдельных устройств и почти не иметь ограничений при реализации интерфейса или драйвера.

Главный недостаток языка ассемблера состоит в том, что написанная для одного типа компьютеров программа не может быть перекомпилирована и использована на других типах компьютеров. Для каждого семейства процессоров используется свой язык ассемблера со своим синтаксисом — это следствие того, что язык ассемблера очень близок машинным командам и архитектуре процессора. Для того чтобы создаваемая программа могла работать на различных типах компьютеров, необходимо использовать языки высокого уровня, такие как C++, C#, Delphi или Java.

Машинный язык

Компьютер не воспринимает непосредственно команды на языке ассемблера, он понимает только машинные команды. Машинные команды состоят из чисел, которые интерпретируются процессором. Процессор обычно имеет встроенный интерпретатор, представляющий *микрпрограмму* и преобразовывающий машинные команды непосредственно в управляющие сигналы.

Машинный язык включает примитивные операторы, которые могут исполнить простые арифметические действия или переместить данные. Система команд процессора состоит из машинных команд, которые он может исполнить. Машинный язык для семейства процессоров Intel в своей основе имеет систему команд процессора 8086, которые могут выполняться процессорами всех последующих модификаций. Такая концепция называется *совместимостью сверху вниз*.

По мере совершенствования процессоров появляются новые команды, которые уже не могут использоваться при написании программ для старых моделей. Такие команды необходимо использовать для повышения скорости работы процессора и разработки программ, учитывающих специфику современных аппаратных устройств.

Конечно, можно написать целую программу на машинном языке, вводя в память двоичные числа машинных команд и заставляя процессор исполнять их. Но человеческий мозг не приспособлен для запоминания большого количества чисел и поэтому вместо ввода чисел используют мнемонически понятный набор команд — язык ассемблера, который преобразует мнемонические коды в машинные команды.