Лекция 7. Системы программирования

Даже при наличии десятков тысяч программ для IBM PC пользователям может потребоваться что-то такое, чего не делают (или делают, но не так) имеющиеся программы. В этих случаях следует использовать системы программирования, т.е. системы для разработки новых программ.

Современные системы программирования для персональных компьютеров обычно предоставляют пользователю весьма мощные и удобные средства для разработки программ. В них входят:

- компилятор, осуществляющий преобразование программ на языке программирования в программу машинных кодах, или интерпретатор, осуществляющий непосредственное выполнение текста программы на языке программирования высокого уровня;
- библиотеки программ, содержащие заранее подготовленные программы, которыми могут пользоваться программисты;
- различные вспомогательные программы, например отладчики, программы для получения перекрестных ссылок и т.д.

Системы программирования, прежде всего, различаются, естественно, по тому, кокой язык программирования они реализуют. Среди программистов пишущих программы для персональных компьютеров, наибольшей популярностью пользуются языки Си, Си++, Паскаль, Бейсик

К системам (или средствам) автоматизации программирования **(САП)** относят языки программирования, языковые трансляторы, редакторы, средства отладки и другие вспомогательные программы.

Языки программирования служат средством передачи информации, средством записи текстов исходных программ. Поэтому в состав программ общего ПО они не входят. Учитывая важность языковых средств, рассмотрим их состав более подробно.

В настоящее время известно несколько сотен языков программирования, которые используют пользователи при разработке своих заданий. Появление новых типов ЭВМ, например ПЭВМ, и новых областей их применения способствует появлению следующих поколений языковых средств, в большей степени отвечающих требованиям потребителей.

Вместе с тем число интенсивно применяемых языков программирования относительно невелико. Для каждого класса ЭВМ всегда существует несколько таких языков, ориентированных на определенные виды обработки информации, на уровень подготовки пользователей в области программирования. При выборе языка программирования пользователь должен учитывать, что описание алгоритма решаемой задачи можно выполнить на любом алгоритмическом языке в силу его универсальности. Однако изобразительные средства языков очень сильно отличаются, и задача выбора заключается в том, чтобы выбранный язык наилучшим образом соответствовал требуемым процедурам обработки данных в задании пользователя. Различают три уровня пользователей, работающих с языковыми средствами: *полъзователи-прикладники, системные программисты* и *инженерно-технический персонал,* обеспечивающий техническое обслуживание ЭВМ. Каждая категория пользователей использует определенный набор языков.

Важнейшими характеристиками языка являются трудоемкость программирования и качество получаемого программного продукта. Качество программ определяется длиной программ (количеством машинных команд или емкостью памяти, необходимой для хранения программ), а также временем выполнения этих программ. Для языков различного уровня эти характеристики взаимосвязаны. Чем выше уровень языка (рис. 1), тем меньше трудоемкость программирования, но тем сложнее средства САП (трансляторы, средства отладки и др.), привлекаемые для получения

машинных программ, тем ниже качество генерируемых программных продуктов.

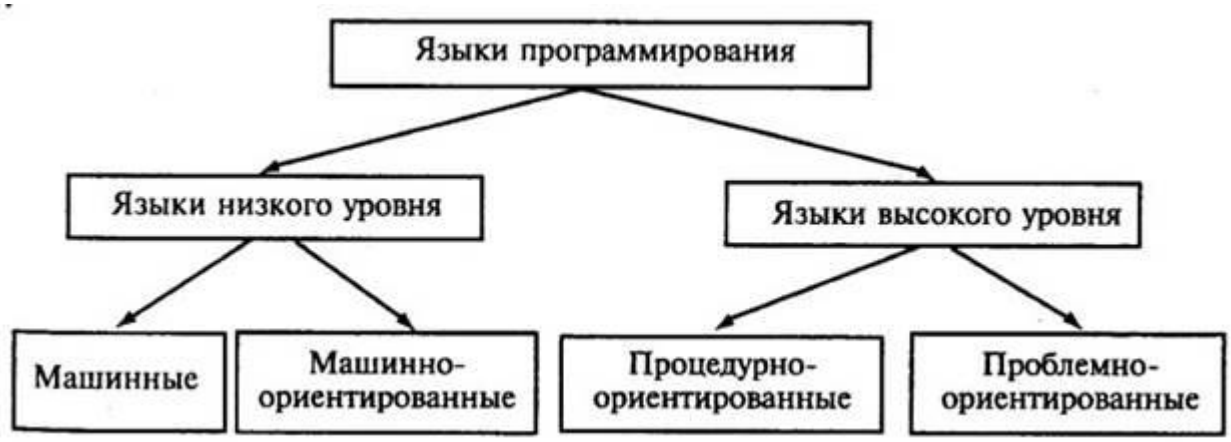

Рис. 1 Классификация языков программирования

Машинные языки современных ЭВМ практически не используются для программирования даже программистами-профессионалами из-за чрезмерной трудоемкости процесса разработки программ. В редких случаях их используют инженерно-технические работники вычислительных центров для проверок работы устройств и блоков ЭВМ, для выяснения нестандартных, нештатных ситуаций, когда другими средствами не удается выявить причины их появления. Применение машинных языков требует знания специфики представления и преобразования информации в ЭВМ.

Особое место имеют *машинно-ориентированные языки* (язык Ассемблер, автокоды, языки символического кодирования и др.). Несмотря на высокую трудоемкость, ими часто пользуются профессиональные системные программисты, например при разработке программ общего или специального ПО, особенно в тех случаях, когда эти программы должны быть максимально компактными и быстродействующими. Пользователям с недостаточной программистской подготовкой эти языки практически недоступны.

Из *процедурно-ориентированных языков* широко известны языки Фортран, Алгол, Кобол, Basic, Pascal, Ада, Си и др. Спектр языков этой группы очень широк, и среди них существует определенная иерархия. Считается, что язык Basic предназначается для начинающих программистов, язык Pascal -язык для студентов, это язык "правильного", классического программирования, язык СИ - язык квалифицированных программистов и т.д. Существуют определенные соглашения в использовании языков программирования. Так, при создании программ для собственных работ пользователь может использовать любой язык, даже Basic. При разработке ПО для одного заказчика корректно использовать язык Pascal, при разработке программных средств для многих потребителей целесообразно использование языков Си и Ассемблер.

С появлением ПЭВМ наиболее распространенными языками являются Basic и Pascal. Первоначально они разрабатывались для целей обучения. Их применение обеспечивает быстрый и удобный перенос программ, написанных на этих языках, с одной ПЭВМ на другую. Наиболее простым языком является Basic. Трансляторы для этого языка имеются практически на всех ПЭВМ. Язык отличает простота и наличие средств интерактивной работы, что обеспечило ему популярность среди непрофессиональных программистов. Однако для построения сложных программ он в силу ограниченных возможностей (структурирование программ и данных, идентификация переменных и т.д.) подходит плохо.

Современный язык высокого уровня Pascal получил широкое распространение в силу ряда достоинств: простоты, ясности, сравнительно узкого набора возможных синтаксических конструкций наряду с семантическим их богатством. Общепризнанно, что он является наилучшим средством для обмена программами между различными типами ПЭВМ. На основе разработки языка Pascal предложен ряд новых языков, например язык Модула-2, в котором особое внимание уделяется построению программы как

набора независимых модулей. На базе языка Pascal создан достаточно мощный язык Ада, который задумывался как универсальный и наиболее перспективный язык программирования. К нему было приковано внимание разработчиков всех новых типов ЭВМ. Однако широкого распространения этот язык до сих пор не получил.

Для разработки коммерческих программ больше используется язык Си, который удачно сочетает в себе средства языка высокого уровня и языка Ассемблер, что позволяет разрабатывать компактные, быстродействующие, высокоэффективные программные продукты.

Все описанные выше языки программирования используют так называемые пошаговые описания алгоритмов. Именно в этом и заключается источник большой трудоемкости подготовки задач к решению. Несомненно, что для машин будущих поколений будут предложены более эффективные средства программирования. Так, например, все больше внимания уделяется разработке *проблемно-ориентированных языков* программирования (Симула, GPSS и др.). В этих языках имеется возможность описывать специфические алгоритмы обработки информации более крупными конструкциями. Это делает программы пользователей более наглядными, так как каждая используемая конструкция соответствует вполне определенному объекту, исследуемому пользователем.

Другой интересной тенденцией является появление *непроцедурных описательных языков.* Конструкции этих языков констатируют, какой результат желателен пользователю, не указывая, каким образом это сделать. Примером такого языка служит язык ПРОЛОГ (Программирование ЛОГики), который широко используется специалистами в области искусственного интеллекта. Конструкции языка соответствуют не математическим формулам, а определяют отношения между объектами и величинами. Язык состоит только из описаний и не имеет как таковых команд-инструкций.

В заключение необходимо отметить, что в машинах будущих поколений будут использоваться языки программирования, имеющие средства распараллеливания вычислительных работ для многомашинных и многопроцессорных вычислительных систем. Проблемы построения таких языков еще полностью не разрешены и находятся в стадии исследования.

В состав САП включаются также *языковые трансляторы* для всех языков, которые используют пользователи при разработке своих программ. В зависимости от специфики вычислительного центра и контингента пользователей их состав формируется эмпирически. Обычно же он включает трансляторы процедурно-ориентированных языков высокого уровня (Pascal, Basic, Си) и машинно-ориентированных языков (Ассемблер).

Различают трансляторы двух типов: интерпретаторы и компиляторы.

Трансляторы-интерпретаторы предназначаются для последовательного пооператорного преобразования каждого предложения исходного модуля программы в блок машинных команд с одновременным их выполнением. Машинная программа в полном объеме при этом не создается, решение задач пользователей происходит замедленными темпами. Этот вид трансляции рекомендуется использовать при отладке новых программных продуктов.

Трансляторы-компиляторы, напротив, предназначаются для формирования полного загрузочного модуля по исходным программам пользователя. Это позволяет отделить полученный программный продукт от среды его разработки и в последующем использовать его автономно.

Из ***системных обслуживающих программ,*** широко используемых при подготовке вычислений, следует выделить *редактор, загрузчик, библиотекарь, средства отладки* и другие вспомогательные программы. Назначение программ вытекает из их названия.

Программы пользователей после обработки их транслятором (трансляторами) представляются в виде набора программных блоков, имеющих промежуточный формат, общий для всех трансляторов. Специфика исходных языков программирования при этом теряется. Объединение программных блоков в единую программу выполняет *редактор.* В зависимости от того, в какой стадии Подготовки к решению находятся программы абонентов, они могут размещаться в различных библиотеках. Управляет размещением программ, последующей идентификацией и выборкой *библиотекарь.* Вызов готовых к решению программ в оперативную память, активизацию их с учетом их места размещения выполняет *загрузчик.*

Средства отладки обеспечивают проверку заданий пользователей, поиск в них различного рода ошибок, вывод на печать запрашиваемой отладочной информации, распечатку содержимого зон оперативной памяти, выдачу различных управляющих блоков и таблиц и т.п.

Вспомогательные программы (утилиты) служат для перемещения информации с одного носителя на другой, разметки накопителей, редактирования информации в наборах данных, сбора информации об ошибках.