

Введение в информатику

Термин "**информатика**" (франц. *informatique*) происходит от французских слов *information* (информация) и *automatique* (автоматика) и дословно означает "**информационная автоматика**".

Широко распространён также англоязычный вариант этого термина — "**Computer science**", что означает буквально "**компьютерная наука**".

Информатика — это основанная на использовании компьютерной техники дисциплина, изучающая структуру и общие свойства информации, а также закономерности и методы её создания, хранения, поиска, преобразования, передачи и применения в различных сферах человеческой деятельности.

В 1978 году международный научный конгресс официально закрепил за понятием "**информатика**" области, связанные с **разработкой, созданием, использованием и материально-техническим обслуживанием систем обработки информации, включая компьютеры и их программное обеспечение, а также организационные, коммерческие, административные и социально-политические аспекты компьютеризации — массового внедрения компьютерной техники во все области жизни людей.**

Таким образом, информатика базируется на компьютерной технике и немыслима без нее.

Информатика — научная дисциплина с широчайшим диапазоном применения. Её **основные направления:**

- **разработка вычислительных систем и программного обеспечения;**
- **теория информации**, изучающая процессы, связанные с передачей, приёмом, преобразованием и хранением информации;
- **методы искусственного интеллекта**, позволяющие создавать программы для решения задач, требующих определённых интеллектуальных усилий при выполнении их человеком (логический вывод, обучение, понимание речи, визуальное восприятие, игры и др.);
- **системный анализ**, заключающийся в анализе назначения проектируемой системы и в установлении требований, которым она должна отвечать;
- **методы машинной графики, анимации, средства мультимедиа;**

- **средства телекоммуникации**, в том числе, **глобальные** компьютерные сети, объединяющие всё человечество в единое информационное сообщество;
- **разнообразные приложения**, охватывающие производство, науку, образование, медицину, торговлю, сельское хозяйство и все другие виды хозяйственной и общественной деятельности.

Информатику обычно представляют состоящей из двух частей:

- **технические средства;**
- **программные средства.**

Технические средства, то есть *аппаратура компьютеров*, в английском языке обозначаются словом *Hardware*, которое буквально переводится как "**твёрдые изделия**".

А для **программных средств** выбрано (а точнее, создано) очень удачное слово *Software* (буквально — "**мягкие изделия**"), которое подчёркивает равнозначность программного обеспечения и самой машины и вместе с тем подчёркивает способность программного обеспечения модифицироваться, приспосабливаться, развиваться.

Программное обеспечение — это совокупность всех программ, используемых компьютерами, а также вся область деятельности по их созданию и применению.

Помимо этих двух общепринятых ветвей информатики выделяют ещё одну существенную ветвь — **алгоритмические средства**. Для неё российский академик А.А. Дородницын предложил название *Brainware* (от англ. *brain* — интеллект). **Эта ветвь связана с разработкой алгоритмов и изучением методов и приёмов их построения.**

Алгоритмы — это правила, предписывающие выполнение последовательностей действий, приводящих к решению задачи.

Нельзя приступить к программированию, не разработав предварительно алгоритм решения задачи.

Роль информатики в развитии общества чрезвычайно велика. С ней связано начало революции в области накопления, передачи и обработки информации. Эта революция, следующая за революциями в овладении веществом и энергией, затрагивает и коренным образом преобразует не только сферу материального производства, но и интеллектуальную, духовную сферы жизни.

Рост производства компьютерной техники, развитие информационных сетей, создание новых информационных технологий приводят к значительным изменениям во всех сферах общества: в производстве, науке, образовании, медицине и т.д.

Термин "*информация*" происходит от латинского слова "*informatio*", что означает *сведения, разъяснения, изложение*.

Информация — это настолько общее и глубокое понятие, что его нельзя объяснить одной фразой. В это слово вкладывается различный смысл в технике, науке и в житейских ситуациях.

В обиходе информацией называют любые данные или сведения, которые кого-либо интересуют.

Например, сообщение о каких-либо событиях, о чьей-либо деятельности и т.п. "*Информировать*" в этом смысле означает "*сообщить нечто, неизвестное раньше*".

Информация — сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые воспринимают информационные системы (живые организмы, управляющие машины и др.) в процессе жизнедеятельности и работы.

Одно и то же информационное сообщение (статья в газете, объявление, письмо, телеграмма, справка, рассказ, чертёж, радиопередача и т.п.) *может содержать разное количество информации для разных людей — в зависимости от их предшествующих знаний, от уровня понимания этого сообщения и интереса к нему.*

Так, сообщение, составленное на японском языке, не несёт никакой новой информации человеку, не знающему этого языка, но может быть высокоинформативным для человека, владеющего японским. Никакой новой информации не содержит и сообщение, изложенное на знакомом языке, если его содержание непонятно или уже известно.

Информация есть характеристика не сообщения, а *соотношения между сообщением и его потребителем*. Без наличия потребителя, хотя бы потенциального, говорить об информации бессмысленно.

В случаях, когда говорят об автоматизированной работе с информацией посредством каких-либо технических устройств, обычно в первую очередь интересуются не содержанием сообщения, а тем, сколько символов это сообщение содержит.

Применительно к компьютерной обработке данных под информацией понимают некоторую последовательность символических обозначений (букв, цифр, закодирован-

ных графических образов и звуков и т.п.), несущую смысловую нагрузку и представленную в понятном компьютеру виде. Каждый новый символ в такой последовательности символов увеличивает информационный объём сообщения.

Информация может существовать в самых разнообразных формах:

- в виде текстов, рисунков, чертежей, фотографий;
- в виде световых или звуковых сигналов;
- в виде радиоволн;
- в виде электрических и нервных импульсов;
- в виде магнитных записей;
- в виде жестов и мимики;
- в виде запахов и вкусовых ощущений;
- в виде хромосом, посредством которых передаются по наследству

признаки и свойства организмов и т.д.

Предметы, процессы, явления материального или нематериального свойства, рассматриваемые с точки зрения их информационных свойств, называются информационными объектами.

В качестве единицы информации условились принять один бит (англ. bit — binary, digit — двоичная цифра).

Бит в теории информации — количество информации, необходимое для различения двух равновероятных сообщений.

А в вычислительной технике битом называют наименьшую "порцию" памяти, необходимую для хранения одного из двух знаков "0" и "1", используемых для внутримашинного представления данных и команд.

Бит — слишком мелкая единица измерения. На практике чаще применяется более крупная единица — ***байт***, равная **восьми битам**. Именно **восемь битов** требуется для того, чтобы закодировать любой из **256** символов алфавита клавиатуры компьютера ($256=2^8$).

Широко используются также ещё **более крупные производные единицы информации**:

- **1 Килобайт (Кбайт) = 1024 байт = 2^{10} байт,**
- **1 Мегабайт (Мбайт) = 1024 Кбайт = 2^{20} байт,**

- **1 Гигабайт (Гбайт) = 1024 Мбайт = 2^{30} байт.**

В последнее время в связи с увеличением объёмов обрабатываемой информации входят в употребление такие производные единицы, как:

- **1 Терабайт (Тбайт) = 1024 Гбайт = 2^{40} байт,**
- **1 Петабайт (Пбайт) = 1024 Тбайт = 2^{50} байт.**

За единицу информации можно было бы выбрать количество информации, необходимое для различения, например, десяти равновероятных сообщений. Это будет не двоичная (**бит**), а десятичная (**дит**) единица информации.

Свойства информации:

- **достоверность;**
- **полнота;**
- **точность;**

и др.

Информация достоверна, если она отражает истинное положение дел. Недостоверная информация может привести к неправильному пониманию или принятию неправильных решений.

Достоверная информация со временем может стать недостоверной, так как она обладает свойством **устаревать**, то есть **перестает отражать истинное положение дел.**

Информация полна, если её достаточно для понимания и принятия решений. Как неполная, так и избыточная информация **сдерживает принятие решений или может повлечь ошибки.**

Точность информации определяется степенью ее близости к реальному состоянию объекта, процесса, явления и т.п.

Информационные ресурсы – это идеи человечества и указания по их реализации, накопленные в форме, позволяющей их воспроизводство.

Это книги, статьи, патенты, диссертации, научно-исследовательская и опытно-конструкторская документация, технические переводы, данные о передовом производственном опыте и др.

Информатизация общества — организованный социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения инфор-

мационных потребностей и реализации прав граждан, органов государственной власти, органов местного самоуправления организаций, общественных объединений на основе формирования и использования информационных ресурсов.

Цель информатизации — улучшение качества жизни людей за счет увеличения производительности и облегчения условий их труда.

Информатизация — это сложный социальный процесс, связанный со значительными изменениями в образе жизни населения. Он требует серьёзных усилий на многих направлениях, включая ликвидацию компьютерной неграмотности, формирование культуры использования новых информационных технологий и др.

Ряд ученых выделяют характерные черты информационного общества:

- решена проблема информационного кризиса, т.е. разрешено противоречие между информационной лавиной и информационным голодом;
- обеспечен приоритет информации по сравнению с другими ресурсами;
- главной формой развития становится информационная экономика;
- в основу общества заложены автоматизированные генерация, хранение, обработка и использование знаний с помощью новейшей информационной техники и технологии;
- информационная технология приобретает глобальный характер, охватывая все сферы социальной деятельности человека;
- формируется информационное единство всей человеческой цивилизации;
- с помощью средств информатики реализован свободный доступ каждого человека к информационным ресурсам всей цивилизации;
- реализованы гуманистические принципы управления обществом и воздействия на окружающую среду.

Кроме положительных моментов, прогнозируются и отрицательные черты:

- все большее влияние на общество средств массовой информации;
- информационные технологии могут разрушить частную жизнь людей и организаций;
- существует проблема отбора качественной и достоверной информации;

- многим людям будет трудно адаптироваться к среде информационного общества. Существует опасность разрыва между "информационной элитой" и потребителями.

В настоящее время все страны мира в той или иной степени осуществляют процесс информатизации. Неправильно выбранная стратегия информатизации или ее недостаточные динамизм и мобильность могут привести к существенным, а подчас драматическим изменениям во всех сферах жизни страны. Первая страна, которая начала информатизацию - это США. Другие промышленно развитые страны мира, поняв перспективность и неизбежность этого направления, достаточно быстро сориентировались и стали наращивать темпы внедрения компьютеров и средств телекоммуникаций. В любой стране независимо от уровня ее развития понимают в той или иной мере неизбежность и необходимость претворения в жизнь идей информатизации общества. Многие страны имеют национальные программы информатизации с учетом местных особенностей и условий.

В период перехода к информационному обществу кроме решения описанных выше проблем необходимо подготовить человека к быстрому восприятию и обработке больших объемов информации, овладению им современными средствами, методами и технологией работы. Кроме того, новые условия работы порождают зависимость информированности одного человека от информации, приобретенной другими людьми. Поэтому уже недостаточно уметь самостоятельно осваивать и накапливать информацию, а надо научиться такой технологии работы с информацией, когда подготавливаются и принимаются решения на основе коллективного знания. Это говорит о том, что человек должен иметь определенный уровень культуры по обращению с информацией.

Для свободной ориентации в информационном потоке человек должен обладать информационной культурой как одной из составляющих общей культуры. Информационная культура связана с социальной природой человека. Она является продуктом разнообразных творческих способностей человека и проявляется в следующих аспектах:

- в конкретных навыках по использованию технических устройств (от телефона до персонального компьютера и компьютерных сетей);
- в способности использовать в своей деятельности компьютерную информационную технологию, базовой составляющей которой являются многочисленные программные продукты;

- в умении извлекать информацию из различных источников: как из периодической печати, так и из электронных коммуникаций, представлять ее в понятном виде и уметь ее эффективно использовать;
- во владении основами аналитической переработки информации;
- в умении работать с различной информацией;
- в знании особенностей информационных потоков в своей собственной области деятельности.

ЭКОНОМИЧЕСКАЯ ИНФОРМАЦИЯ: ОСОБЕННОСТИ, ВИДЫ, СТРУКТУРА

Каждая область человеческой деятельности связана со "своей" информацией. Экономическая наука, экономическая деятельность общества оперируют информацией, которая называется *экономической*. Как категория экономическая информация, с одной стороны, соответствует общему понятию "информация", с другой - неразрывно связана с экономикой и управлением народным хозяйством. Поэтому на экономическую информацию распространяются различные толкования и свойства: присущие информации вообще и в то же время отражающие ее характерные особенности и отличия, вытекающие из ее природы. Таким образом, экономическая информация представляет собой лишь одну из разновидностей информации, хотя и достаточно важную.

В общем случае под *термином "экономическая информация"* понимается информация, отражающая и обслуживающая процессы производства, распределения, обмена и потребления материальных благ. Более строго: экономическая информация - это объективизированное воплощение - с помощью знаков и сигналов - знаний о материальных, трудовых и стоимостных аспектах воспроизводимых в экономике процессов, устраняющих неопределенность в отношении исходов этих процессов.

Экономическая информация служит инструментом управления и одновременно принадлежит к его элементам. Ее необходимо рассматривать как одну из разновидностей *управленческой информации*, которая обеспечивает решение задач организационно-экономического управления народным хозяйством. Экономическая информация представляет собой совокупность сведений (данных), отражающих состояние и определяющих направление развития народного хозяйства и его отдельных звеньев. В информационных процессах, осуществляемых в управлении, информация играет роль предме-

та труда (исходная, "сырая" информация) и продукта труда (результатная, "обработанная" информация). Говоря о понятии "экономическая информация" с кибернетических позиций, информационный процесс управления можно охарактеризовать как превращение сведений (исходных данных) в экономическую информацию, необходимую для принятия решений, направленных на обеспечение заданного состояния народного хозяйства и его оптимального развития.

Экономическая информация неотделима от информационного процесса управления, осуществляемого в производственной и непроизводственной сферах, она используется во всех отраслях народного хозяйства и во всех органах общегосударственного управления.

Экономическая информация насчитывает много разновидностей (типов), которые выделяются на основе соответствующих классификационных критериев, например:

- **по принадлежности к сфере материального производства и непроизводственной сфере**, а внутри - по отраслям и подотраслям народного хозяйства в соответствии с принятой их группировкой;

- **по стадиям воспроизводства и элементам производственного процесса**. В силу этого выделяется информация, отражающая снабжение, производство, распределение и потребление, а также материальные, трудовые и финансовые ресурсы;

- **по временным стадиям управления**. С этих позиций выделяются: прогнозная информация, плановая, учетная, информация для анализа хозяйственной деятельности, оперативного управления, составления отчетности;

- **по критериям соответствия отражаемым явлениям** экономической информация делится на достоверную и недостоверную;

- **по полноте отражения событий** экономическая информация подразделяется на достаточную (полную), недостаточную и избыточную. Для решения задач экономического управления необходима вполне конкретная по содержательности минимальная информация, т.е. достаточная. Избыточная информация содержит излишние данные, которые либо вообще не используются для решения задач (составляют «информационный шум» с его разновидностью для компьютерных сетей - «электронным мусором», или «спэмом» (spam - англ.), либо выполняют контрольно-дублирующие функции;

- **по стадии возникновения** бывает исходная (первичная) и производная (вторичная) информация. *Исходная информация* возникает в результате действия источников

информации (министерства, ведомства, предприятия и других организаций и подразделений), и по этим источникам исходная информация делится на планово-директивную и учетно-отчетную. *Планово-директивная информация* перемещается ("спускается") вниз по уровням иерархии управления, при этом она разукрупняется, детализируется, "расширяется". *Учетно-отчетная информация*, наоборот, перемещается вверх по уровням иерархии управления, причем по мере своего продвижения эта информация укрупняется, агрегируется, "сжимается". Производная информация возникает в результате обработки исходной и другой вторичной - либо только исходной, либо только вторичной. Среди производной информации различают *промежуточную* и *окончательную* (результатную) информацию;

• *по стабильности во времени* экономическая информация делится на *постоянную* (условно-постоянную) и *переменную*. При этом важно отметить, что период стабильности носит конкретный характер для определенных задач управления. Как правило, он составляет один год, а для оценки уровня стабильности информации используется коэффициент стабильности K , рассчитываемый по формуле

$$K_{ст} = \frac{ИС_{общ} - ИС_{изм}}{ИС_{общ}},$$

Если $K \geq 0,85$, то информационную совокупность принято считать условно-постоянной, и в условиях автоматизированной обработки информации ее следует организовывать и хранить в виде самостоятельных массивов нормативно-справочной информации (НСИ) или файлов баз данных;

• *по технологии решения* экономических задач в системах управления различают *входную*, *промежуточную* и *выходную* информацию.

Экономической информации свойственны некоторые особенности, вытекающие из ее сущности. Принципиальное значение для создания систем обработки экономической информации и формирования информационных технологий имеют следующие ее свойства:

- преобладание алфавитно-цифровых знаков;

- необходимость оформления результатов обработки данных в форме, удобной для восприятия человеком;

- широкое распространение документов как носителей исходных данных и результатов их обработки;

- значительный объем переменных и постоянных (условно-постоянных) данных;

- дискретность, объясняющаяся тем, что экономическая информация характеризует состояние объекта или процесса либо на определенный момент времени, либо за определенный интервал времени;

- организованность, вытекающая из того, что экономическая информация отражает результат интеллектуальной деятельности человека;

- неоднородность - в силу основного назначения информации:

различать элементы и свойства отражаемых процессов;

- рассредоточенность источников и принципиальная невозможность концентрации и централизации процессов сбора данных;

- сохраняемость (неиссякаемость) при ее использовании (потреблении);

- возможность многократного использования одних и тех же данных, в том числе и одновременно разными потребителями;

- возможность сохранения переработанной информации у отправителя;

- возможность длительного хранения с воспроизведением и обновлением;

- способность к преобразованию, агрегированию по определенным признакам, детализации (разукрупнению) и сжатию (укрупнению);

- определенная самостоятельность данных по отношению к своему носителю.

В условиях выполнения функций управления теми или иными объектами экономическая информация должна отвечать определенным требованиям. Наиболее существенные из них:

- быть достоверной, правдивой;

- быть своевременной, так как запоздалое поступление нужной информации часто оказывается бесполезным;

- быть документальной: юридически подтвержденной в документах соответствующими подписями (визами) соответствующих должностных лиц;

- быть актуальной, нужной для подразделения и лиц, принимающих решения.

Важной характеристикой экономической информации является ее структура. **Структура информации** играет ту же роль, что синтаксис любого языка. Говоря о структуре информации, различают два взаимосвязанных между собой аспекта:

- состав элементов, образующих структуру информации;

- взаимосвязь между элементами этой структуры.

Рассматривая с этих позиций структуру информации, выделяют простые и составные (сложные) единицы информации.

Составной единицей информации (СЕЙ) называют единицу информации, состоящую из совокупности других единиц информации, ассоциативно связанных между собой, т.е. связанных по смыслу.

Единицу информации, входящую в СЕЙ, называют *составляющей единицей информации*.

Простой, элементарной составляющей единицей экономической информации **является референт**. Референтам присущи два свойства, важных с точки зрения их обработки:

- отдельно взятый референт не может полностью характеризовать экономический процесс или объект;
- отдельный референт может входить в состав различных экономических показателей.

Каждый референт характеризуется именем (наименованием), типом и значением.

В зависимости от характера отображаемого **ими** свойства референты делятся на референты-признаки и референты-основания.

Референты-признаки отражают качественные свойства экономического объекта, процесса или явления. Они могут быть выражены в алфавитном, цифровом или алфавитно-цифровом виде. Референты-признаки служат для логической обработки составных единиц, т.е. для поиска, сортировки, группировки, выборки и т.д.

Референты-основания характеризуют количественную сторону процесса или объекта. Они чаще всего выражаются в цифровой форме. Над ними могут выполняться логические и арифметические операции.

Референты можно расчленивать и дальше - на символы и биты, но при этом теряется смысловое содержание референтов. Для исчерпывающей характеристики экономического процесса, объекта или явления необходима определенная совокупность референтов, описывающих качественные и количественные свойства отображаемого объекта.

Совокупность референтов-признаков и референтов-оснований представляет собой *сообщение* об объекте.

Понятие баз данных, их состав, назначение, организация.

Наряду с понятием «информация» широко употребляется термин «данные» как синоним сведений. Это могут быть отдельные их отношения, словосочетания, факты, преобразование и обработка которых позволяет извлечь информацию. Иными словами, данные служат сырьем для создания информации.

Встречается и иное толкование термина «данные». Согласно этому толкованию информация может быть представлена в специфической форме, позволяющей осуществить ее дальнейшую обработку, в том числе и техническими средствами. Результат такого представления и называют данными.

Смысл указанного толкования термина «данные» заключается в том, что данные представляют собой закодированную информацию.

База данных – поименованная совокупность структурированных данных, относящихся к определенной предметной области.

В настоящее время в мире широко представлены компьютерные информационные системы, которые характеризуются огромными объемами хранимых данных, сложной организацией, необходимостью удовлетворять разнообразные требования многочисленных пользователей.

В основе каждой информационной системы лежит база данных. Хорошо спроектированная база данных содержит совокупность избыточных, непротиворечивых данных, защищенных от несанкционированного доступа.

Пользователями компьютерной базы данных могут быть различные прикладные программы, программные комплексы, специалисты предметной области, выступающие в роли потребителей или источников информации.

Логическая организация базы данных – представление пользователя (проектировщика) о той предметной области, информация из которой должна храниться в базе данных.

В результате создается концептуальная (внешняя, информационно—логическая) схема базы данных.

Известны несколько логических моделей: иерархическая, сетевая, реляционная, объектно—ориентированная. Наиболее широкое применение на персональных компьютерах нашли реляционные модели.

Реляционная модель ориентирована на организацию данных в виде двумерных таблиц, строки которых соответствуют записям, а столбцы – полям.

Понятие баз знаний, их состав, назначение, организация.

Для решения сложных трудно формализуемых научных, производственных и экономических задач и тиражирования профессионального опыта применяются системы искусственного интеллекта.

Основное направление искусственного интеллекта – представление знаний и разработка систем, основанных на знаниях.

Знания – это выявленные закономерности предметной области (принципы, связи, позволяющие решать задачи в этой области.

Переход от данных к знаниям – следствие развития и усложнения информационно—логических структур, обрабатываемых с помощью компьютера.

Особенности знаний. в которых заключается их отличие от данных(при обработке на компьютере):

- интерпретируемость – знания всегда содержательны и понятны человеку, в отличие от данных, которые не имеют смысла без обрабатывающих их алгоритмов;
- структурированность – выполняется декомпозиция сложных объектов на более простые и установление связей между ними;
- связанность – отражаются закономерности относительно фактов, явлений и причинно-следственные отношения между ними;
- активность – знания предполагают целенаправленное использование информации, способность управлять информационными процессами по решению определенных задач.

Преобразование знаний в объект обработки на компьютере – это задача информационной технологии инженерии знаний.

Инженер по знаниям (аналитик) является главной фигурой при извлечении знаний из источника знаний. Результат его работы отражает совокупность представлений и рассуждений специалистов. В результате появляется программа – экспертная система, которая решает задачи так же как эксперты—люди. Т.о. можно сделать вывод:

- экспертная система включает в себя базу знаний с набором правил и механизм вывода;

- база знаний – систематизированная совокупность знаний предметной области, описанных с использованием выбранной формы их представления.

Общие принципы организации и работы компьютеров

Компьютер (англ. *computer* — вычислитель) представляет собой программируемое электронное устройство, способное обрабатывать данные и производить вычисления, а также выполнять другие задачи манипулирования символами.

Существует два основных класса компьютеров:

- **цифровые компьютеры**, обрабатывающие данные в виде числовых двоичных кодов;
- **аналоговые компьютеры**, обрабатывающие непрерывно меняющиеся физические величины (электрическое напряжение, время и т.д.), которые являются аналогами вычисляемых величин.

Поскольку в настоящее время подавляющее большинство компьютеров являются цифровыми, далее будем рассматривать только этот класс компьютеров и слово "*компьютер*" употреблять в значении "*цифровой компьютер*".

Основу компьютеров образует аппаратура (*HardWare*), построенная, в основном, с использованием электронных и электромеханических элементов и устройств. Принцип действия компьютеров состоит в выполнении программ (*SoftWare*) — заранее заданных, четко определённых последовательностей арифметических, логических и других операций.

Любая компьютерная программа представляет собой последовательность отдельных команд.

Команда — это описание операции, которую должен выполнить компьютер. Как правило, у команды есть свой *код* (условное обозначение), *исходные данные* (операнды) и *результат*.

Например, у команды "*сложить два числа*" операндами являются слагаемые, а результатом — их сумма. А у команды "*стоп*" операндов нет, а результатом является прекращение работы программы.

Результат команды вырабатывается по точно определенным для данной команды правилам, заложенным в конструкцию компьютера.

Совокупность команд, выполняемых данным компьютером, называется системой команд этого компьютера.

Компьютеры работают с очень высокой скоростью, составляющей миллионы - сотни миллионов операций в секунду.

Структурная организация ЭВМ

Под структурой ЭВМ понимается совокупность входящих в ее состав устройств и связей между ними. Основными структурными составляющими являются:

- **память** (запоминающее устройство, ЗУ), состоящую из перенумерованных ячеек;
- **процессор**, включающий в себя **устройство управления (УУ)** и **арифметико-логическое устройство (АЛУ)**;
- **устройство ввода**;
- **устройство вывода**.

Эти устройства соединены *каналами связи*, по которым передается информация.

Функции памяти:

- *приём информации* из других устройств;
- *запоминание информации*;
- *выдача информации* по запросу в другие устройства машины.

Функции процессора:

- *обработка данных по заданной программе* путем выполнения арифметических и логических операций;
- *программное управление работой устройств* компьютера.

Та часть процессора, которая выполняет команды, называется арифметико-логическим устройством (АЛУ), а другая его часть, выполняющая функции управления устройствами, называется устройством управления (УУ).

Обычно эти два устройства выделяются чисто условно, *конструктивно они не разделены*.

В составе процессора имеется ряд специализированных дополнительных ячеек памяти, называемых *регистрами*.

Регистр выполняет функцию кратковременного хранения числа или команды. Над содержимым некоторых регистров специальные электронные схемы могут выполнять некоторые манипуляции. Например, "вырезать" отдельные части команды для последующего их использования или выполнять определенные арифметические операции над числами.

Основным элементом регистра является электронная схема, называемая триггером, которая способна хранить одну двоичную цифру (*разряд*).

Регистр представляет собой совокупность триггеров, связанных друг с другом определённым образом общей системой управления.

Существует несколько типов регистров, отличающихся видом выполняемых операций.

Некоторые важные регистры имеют свои названия, например:

- сумматор — регистр АЛУ;
- счетчик команд — регистр УУ, содержимое которого соответствует адресу очередной выполняемой команды; служит для автоматической выборки программы из последовательных ячеек памяти;

- **регистр команд** — регистр УУ для хранения кода команды на период времени, необходимый для ее выполнения. Часть его разрядов используется для хранения *кода операции*, остальные — для хранения *кодов адресов операндов*.

Принципы построения компьютера

В основу построения подавляющего большинства компьютеров положены следующие общие принципы, сформулированные в 1945 г. американским ученым **Джоном фон Нейманом**.

1. Принцип программного управления. Из него следует, что программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности.

Выборка программы из памяти осуществляется с помощью **счетчика команд**. Этот регистр процессора **последовательно увеличивает хранимый в нем адрес очередной команды на длину команды**.

А так как команды программы расположены в памяти друг за другом, то тем самым организуется выборка цепочки команд из последовательно расположенных ячеек памяти.

Если же нужно после выполнения команды перейти не к следующей, а к какой-то другой, используются команды **условного** или **безусловного переходов**, которые **заносят в счетчик команд номер ячейки памяти, содержащей следующую команду**. Выборка команд из памяти прекращается после достижения и выполнения команды *“стоп”*.

Таким образом, **процессор исполняет программу автоматически, без вмешательства человека**.

2. Принцип однородности памяти. Программы и данные хранятся в одной и той же памяти. Поэтому компьютер не различает, что хранится в данной ячейке

памяти — число, текст или команда. Над командами можно выполнять такие же действия, как и над данными.

Это открывает целый ряд возможностей. Например, программа в процессе своего выполнения также может подвергаться переработке, что позволяет задавать в самой программе правила получения некоторых ее частей (так в программе организуется выполнение циклов и подпрограмм).

Более того, команды одной программы могут быть получены как результаты исполнения другой программы. На этом принципе основаны методы трансляции — перевода текста программы с языка программирования высокого уровня на язык конкретной машины.

3. Принцип адресности. Структурно основная память состоит из перенумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка.

Отсюда следует возможность давать имена областям памяти, так, чтобы к запомненным в них значениям можно было впоследствии обращаться или менять их в процессе выполнения программ с использованием присвоенных имен.

Компьютеры, построенные на этих принципах, относятся к типу **фон-неймановских**.

- Но существуют компьютеры, принципиально отличающиеся от фон-неймановских. Для них, например, может *не выполняться принцип программного управления*, т.е. они могут работать без “счетчика команд”, указывающего текущую выполняемую команду программы. Для обращения к какой-либо переменной, хранящейся в памяти, этим компьютерам *не обязательно давать ей имя*. Такие компьютеры называются **не-фон-неймановскими**.

Архитектура и структура компьютера

При рассмотрении компьютерных устройств принято различать их архитектуру и структуру.

Архитектурой компьютера называется его описание на некотором общем уровне, включающее описание пользовательских возможностей программирования, системы команд, системы адресации, организации памяти и т.д. Архитектура определяет принципы действия, информационные связи и взаимное соединение основных логических узлов компьютера: процессора, оперативного ЗУ, внешних ЗУ и периферийных устройств. Общность архитектуры разных компьютеров обеспечивает их совместимость с точки зрения пользователя.

Структура компьютера — это совокупность его функциональных элементов и связей между ними. Элементами могут быть самые различные устройства — от основных логических узлов компьютера до простейших схем. Структура компьютера графически представляется в виде структурных схем, с помощью которых можно дать описание компьютера на любом уровне детализации.

Наиболее распространены следующие архитектурные решения.

- **Классическая архитектура** (архитектура фон Неймана) — *одно арифметико-логическое устройство (АЛУ), через которое проходит поток данных, и одно устройство управления (УУ), через которое проходит поток команд — программа. Это одно-процессорный компьютер.*

К этому типу архитектуры относится и архитектура персонального компьютера с **общей шиной**, подробно рассмотренная в разделе. Все функциональные блоки здесь связаны между собой общей шиной, называемой также ***системной магистралью***.

Физически **магистраль** представляет собой многопроводную линию с гнездами для подключения электронных схем. Совокупность проводов магистрали разделяется на отдельные группы: шину адреса, шину данных и шину управления.

Периферийные устройства (принтер и др.) подключаются к аппаратуре компьютера через специальные **контроллеры** — **устройства управления периферийными устройствами**.

Контроллер — устройство, которое связывает периферийное оборудование или каналы связи с центральным процессором, освобождая процессор от непосредственного управления функционированием данного оборудования.

- **Многопроцессорная архитектура.** Наличие в компьютере нескольких процессоров означает, что **параллельно** может быть **организовано много потоков данных и много потоков команд**. Таким образом, **параллельно могут выполняться несколько фрагментов одной задачи**. Структура такой машины, имеющей общую оперативную память и несколько процессоров, представлена на рис. 1.

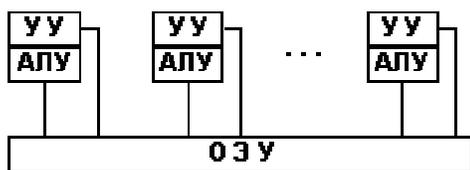


Рис. 1. Архитектура многопроцессорного компьютера

- **Многомашинная вычислительная система.** Здесь **несколько процессоров**, входящих в вычислительную систему, **не имеют общей оперативной памяти, а имеют каждый свою (локальную)**. Каждый компьютер в многомашинной системе имеет классическую архитектуру, и такая система применяется достаточно широко.

Однако эффект от применения такой вычислительной системы может быть получен только при решении задач, **имеющих очень специальную структуру: она должна разбиваться на столько слабо связанных подзадач, сколько компьютеров в системе.**

Преимущество в быстродействии многопроцессорных и многомашинных вычислительных систем перед однопроцессорными очевидно.

- **Архитектура с параллельными процессорами.** Здесь **несколько АЛУ работают под управлением одного УУ**. Это означает, что множество данных может обрабатываться по одной программе — то есть по одному потоку команд.

Высокое быстродействие такой архитектуры можно получить только на задачах, в которых одинаковые вычислительные операции выполняются одновременно на различных однотипных наборах данных. Структура таких компьютеров представлена на рис. 2.

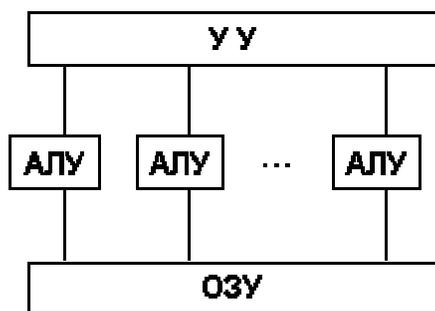


Рис. 2. Архитектура с параллельным процессором

В современных машинах часто присутствуют элементы различных типов архитектурных решений. Существуют и такие архитектурные решения, которые радикально отличаются от рассмотренных выше.

Центральный процессор

Центральный процессор (CPU, от англ. *Central Processing Unit*) — это основной рабочий компонент компьютера, который выполняет арифметические и логические операции, заданные программой, управляет вычислительным процессом и координирует работу всех устройств компьютера.

Центральный процессор в общем случае содержит в себе:

- арифметико-логическое устройство;
- шины данных и шины адресов;
- регистры;
- счетчики команд;
- кэш — очень быструю память малого объема (от 8 до 512 Кбайт);
- математический сопроцессор чисел с плавающей точкой.

Современные процессоры выполняются в виде микروпроцессоров. **Физически микропроцессор представляет собой интегральную схему** — тонкую пластинку кристаллического кремния прямоугольной формы площадью всего несколько квадратных миллиметров, на которой размещены схемы, реализующие все функции процессора. Кристалл-пластинка обычно помещается в пластмассовый или керамический плоский корпус и соединяется золотыми проводками с металлическими штырьками, чтобы его можно было присоединить к системной плате компьютера.

В вычислительной системе может быть несколько параллельно работающих процессоров; такие системы называются **многопроцессорными**.

Память

Память компьютера построена из двоичных запоминающих элементов — *битов*, объединенных в группы по 8 битов, которые называются *байтами*. (Единицы измерения памяти совпадают с единицами измерения информации). Все байты пронумерованы. Номер байта называется его *адресом*.

Байты могут объединяться в ячейки, которые называются также *словами*. Для каждого компьютера характерна определенная длина слова — два, четыре или восемь байтов. Это не исключает использования ячеек памяти другой длины (например, Широко используются и более крупные производные единицы объема памяти: *Килобайт*, *Мегабайт*, *Гигабайт*, а также, в последнее время, *Терабайт* и *Петабайт*.

Современные компьютеры имеют много разнообразных запоминающих устройств, которые сильно отличаются между собой по назначению, временным характеристикам, объёму хранимой информации и стоимости хранения одинакового объёма информации.

Различают два основных вида памяти — *внутреннюю* и *внешнюю*.

Внутренняя память

В состав внутренней памяти входят *оперативная память*, *кэш-память* и *специальная память*.

- **Оперативная память**

Оперативная память (*ОЗУ*, англ. *RAM, Random Access Memory* — *память с произвольным доступом*) — это быстрое запоминающее устройство не очень большого объёма, непосредственно связанное с процессором и предназначенное для записи, считывания и хранения выполняемых программ и данных, обрабатываемых этими программами.

Оперативная память используется только для *временного хранения данных и программ*, так как, **когда машина выключается, все, что находилось в ОЗУ, пропадает**. Доступ к элементам оперативной памяти *прямой* — это означает, что **каждый байт памяти имеет свой индивидуальный адрес**.

Для эффективной работы современного программного обеспечения желательно иметь не менее 64 Мбайт ОЗУ. Обычно ОЗУ **исполняется из интегральных микросхем памяти DRAM** (*Dynamic RAM* — динамическое ОЗУ). Микросхемы DRAM работают медленнее, чем другие разновидности памяти, но стоят дешевле.

Каждый информационный бит в DRAM запоминается в виде электрического заряда крохотного конденсатора, образованного в структуре полупроводникового кристалла. Из-за токов утечки такие конденсаторы быстро разряжаются, и их периодически (примерно каждые 2 миллисекунды) подзаряжают специальные устройства. Этот процесс называется **регенерацией памяти** (Refresh Memory).

Современные микросхемы имеют **ёмкость более 16 Мбит**. Они устанавливаются в корпуса и собираются в **модули памяти**.

Наиболее распространены модули типа SIMM (*Single In-Line Memory Module* — *модуль памяти с однорядным расположением микросхем*).

Важная характеристика модулей памяти — **время доступа** к данным, которое обычно составляет 60 – 80 наносекунд.

- **Кэш-память**

Кэш (англ. *cache*), или **сверхоперативная память** — очень быстрое ЗУ небольшого объёма, которое используется при обмене данными между микропроцессором и оперативной памятью для компенсации разницы в скорости обработки информации процессором и несколько менее быстросействующей оперативной памятью.

Кэш-памятью управляет специальное устройство — **контроллер**, который, анализируя выполняемую программу, пытается ***предвидеть, какие данные и команды вероятнее всего понадобятся в ближайшее время процессору, и подкачивает их в кэш***.

память. При этом возможны как "**попадания**", так и "**промахи**". В случае *попадания*, то есть, *если в кэш подкачаны нужные данные*, извлечение их из памяти происходит без задержки. Если же требуемая информация в кэше отсутствует, то процессор считывает её непосредственно из оперативной памяти. Соотношение числа попаданий и промахов определяет эффективность кэширования.

Кэш-память реализуется на микросхемах статической памяти SRAM (*Static RAM*), более быстродействующих, дорогих и малоёмких, чем DRAM.

Современные микропроцессоры имеют **встроенную кэш-память**, так называемый **кэш первого уровня**. Кроме того, на системной плате компьютера может быть установлен **кэш второго уровня** ёмкостью от 64 Кбайт до 256 Кбайт и выше.

- **Специальная память**

К устройствам специальной памяти относятся **постоянная память (ROM)**, **перепрограммируемая постоянная память (Flash Memory)**, **память CMOS RAM**, питаемая от батарейки, **видеопамять** и некоторые другие виды памяти.

Постоянная память (ПЗУ, *англ. ROM, Read Only Memory* — память только для чтения) — энергонезависимая память, используется для хранения данных, которые никогда не потребуют изменения. Содержание памяти специальным образом "*зашифтуется*" в устройстве при его изготовлении для постоянного хранения. Из ПЗУ можно только читать.

Перепрограммируемая постоянная память (Flash Memory) — энергонезависимая память, допускающая многократную перезапись своего содержимого с дискеты.

Прежде всего в постоянную память записывают программу управления работой самого процессора. В ПЗУ находятся программы управления дисплеем, клавиатурой, принтером, внешней памятью, программы запуска и остановки компьютера, тестирования устройств.

Важнейшая микросхема постоянной или Flash-памяти — модуль BIOS.

BIOS (*Basic Input/Output System* — базовая система ввода-вывода) — совокупность программ, предназначенных для:

- автоматического тестирования устройств после включения питания компьютера;
- загрузки операционной системы в оперативную память.

Роль BIOS двоякая: с одной стороны это неотъемлемый элемент аппаратуры (*Hardware*), а с другой стороны — важный модуль любой операционной системы (*Software*).

Разновидность постоянного ЗУ — **CMOS RAM**.

CMOS RAM — это память с невысоким быстродействием и минимальным энергопотреблением от батарейки. Используется для хранения информации о конфигурации и составе оборудования компьютера, а также о режимах его работы.

Содержимое CMOS изменяется специальной программой *Setup*, находящейся в BIOS (*англ. Set-up* — устанавливать, читается "сетап").

Для хранения графической информации используется **видеопамять**.

Видеопамять (VRAM) — разновидность оперативного ЗУ, в котором хранятся закодированные изображения. Это ЗУ организовано так, что его содержимое доступно сразу двум устройствам — процессору и дисплею. Поэтому изображение на экране меняется одновременно с обновлением видеоданных в памяти.

Внешняя память

Внешняя память (ВЗУ) предназначена для длительного хранения программ и данных, и целостность её содержимого не зависит от того, включен или выключен компьютер. В отличие от оперативной памяти, внешняя память не имеет прямой связи с процессором. Информация от ВЗУ к процессору и наоборот циркулирует примерно по следующей цепочке:



В состав внешней памяти компьютера входят:

- накопители на *жёстких* магнитных дисках;
- накопители на *гибких* магнитных дисках;
- накопители на *компакт-дисках*;
- накопители на *магнито-оптических* компакт-дисках;
- накопители на *магнитной ленте* (стримеры) и др.

Накопители на гибких магнитных дисках

Дискета состоит из круглой полимерной подложки, покрытой с обеих сторон магнитным окислом и помещенной в пластиковую упаковку, на внутреннюю поверхность которой нанесено очищающее покрытие. В упаковке сделаны с двух сторон радиальные прорезы, через которые головки считывания/записи накопителя получают доступ к диску.

Способ записи двоичной информации на магнитной среде называется магнитным кодированием. Он заключается в том, что магнитные домены в среде выстраиваются вдоль дорожек в направлении приложенного магнитного поля своими северными и южными полюсами. Обычно устанавливается однозначное соответствие между двоичной информацией и ориентацией магнитных доменов.

Информация записывается по концентрическим *дорожкам* (*трекам*), которые делятся на *секторы*. Количество дорожек и секторов зависит от типа и формата дискеты. Сектор хранит минимальную порцию информации, которая может быть записана на диск или считана. Ёмкость сектора постоянна и составляет 512 байтов.



Рис. 3. Поверхность магнитного диска

3.

Поверхность

На дискете можно хранить до 2,88 Мегабайт информации.

В настоящее время наибольшее распространение получили **дискеты со следующими характеристиками**: диаметр 3,5 дюйма (89 мм), ёмкость 1,44 Мбайт, число дорожек 80, количество секторов на дорожках 18.

Дискета устанавливается в накопитель на гибких магнитных дисках (англ. *floppy-disk drive*), автоматически в нем фиксируется, после чего механизм накопителя раскручивается до частоты вращения 360 мин^{-1} . В накопителе вращается сама дискета, магнитные головки остаются неподвижными. Дискета вращается только при обращении к ней.

Накопитель связан с процессором через контроллер гибких дисков.

Накопители на жестких магнитных дисках

Если гибкие диски — это средство переноса данных между компьютерами, то **жесткий диск — информационный склад компьютера**.

Накопитель на жёстких магнитных дисках (англ. HDD — *Hard Disk Drive*) или винчестерский накопитель — это наиболее массовое запоминающее устройство большой ёмкости, в котором носителями информации являются круглые алюминиевые пластины — *платтеры*, обе поверхности которых покрыты слоем магнитного материала. Используется для постоянного хранения информации — программ и данных.

Как и у дискеты, рабочие поверхности платтеров разделены на кольцевые концентрические дорожки, а дорожки — на секторы. Головки считывания-записи вместе с их несущей конструкцией и дисками заключены в герметически закрытый корпус, называемый *модулем данных*. При установке модуля данных на дисковод он автоматически соединяется с системой, подкачивающе очищенный охлажденный воздух.

Поверхность платтера имеет *магнитное покрытие* толщиной всего лишь в 1,1 мкм, а также *слой смазки* для предохранения головки от повреждения при опускании и подъёме на ходу. При вращении платтера над ним образуется *воздушный слой*, который

обеспечивает *воздушную подушку* для зависания головки на высоте 0,5 мкм над поверхностью диска.

Винчестерские накопители имеют *очень большую ёмкость*: до десятков Гбайт. У современных моделей скорость вращения шпинделя достигает 10000 оборотов в минуту, среднее время поиска данных — 4 мс, максимальная скорость передачи данных до 40 Мбайт/с.

В отличие от дискеты, винчестерский диск *вращается непрерывно*.

Винчестерский накопитель связан с процессором через *контроллер жесткого диска*.

Все современные накопители снабжаются *встроенным кэшем* (256 Кбайт и более), который существенно повышает их производительность.

Накопители на компакт-дисках

CD-ROM состоит из прозрачной полимерной основы диаметром 12 см и толщиной 1,2 мм. Одна сторона покрыта тонким алюминиевым слоем, защищенным от повреждений слоем лака. **Двоичная информация представляется последовательным чередованием углублений (*pits* — ямки) и основного слоя (*land* — земля).**

На одном дюйме (2,54 см) по радиусу диска размещается 16 тысяч дорожек с информацией. Для сравнения — на дюйме по радиусу дискеты всего лишь 96 дорожек. Ёмкость CD до 800 Мбайт. *Информация заносится на диск на заводе и не может быть изменена.*

Достоинства CD-ROM:

- При малых физических размерах **CD-ROM** обладают *высокой информационной ёмкостью*, что позволяет использовать их в справочных системах и в учебных комплексах с богатым иллюстративным материалом; *один CD, имея размеры примерно дискеты, по информационному объёму равен почти 500 таким дискетам;*

- **Считывание информации с CD происходит с высокой скоростью, сравнимой со скоростью работы винчестера;**

- CD просты и удобны в работе, практически не изнашиваются;
- CD не могут быть поражены вирусами;
- На CD-ROM **невозможно случайно стереть информацию;**
- Стоимость хранения данных (в расчете на 1 Мбайт) низкая.

В отличие от магнитных дисков, компакт-диски имеют не множество кольцевых дорожек, а *одну* — **спиральную**, как у грампластинок. В связи с этим, угловая скорость вращения диска не постоянна. Она линейно уменьшается в процессе продвижения читающей магнитной головки к центру диска.

Для работы с CD ROM нужно подключить к компьютеру **накопитель CD-ROM (CD-ROM Drive)**, в котором компакт-диски сменяются как в обычном проигрывателе. Накопители CD-ROM часто называют **проигрывателями CD-ROM** или **приводами CD-ROM**.

Участки CD, на которых записаны символы "0" и "1", отличаются коэффициентом отражения лазерного луча, посылаемого накопителем CD-ROM. Эти отличия улавливаются фотоэлементом, и общий сигнал преобразуется в соответствующую последовательность нулей и единиц.

Многие накопители CD-ROM способны воспроизводить обычные аудио-CD. Это позволяет пользователю, работающему за компьютером, слушать музыку в фоновом режиме.

На смену CD-ROM приходят **цифровые видеодиски DVD** (читается "ди-ви-ди"). Эти диски имеют тот же размер, что и обычные CD, но вмещают **4,7 Гбайт** данных и более, т.е. по объёму заменяют *семь стандартных дисков CD-ROM*. На таких дисках удобно выпускать полноэкранные видеофильмы отличного качества, программы-тренажёры, мультимедийные игры и многое другое.

Записывающие оптические и магнитооптические накопители

Накопитель на магнито-оптических компакт-дисках CD-МО (Compact Disk-Magneto Optical). Диски CD-МО можно многократно использовать для записи, но они не читаются на традиционных дисководов CD-ROM. Ёмкость от 128 Мбайт до 2,6 Гбайт.

- **Записывающий накопитель CD-R** (Compact Disk Recordable) способен, наряду с прочтением обычных компакт-дисков, **записывать информацию на специальные оптические диски**. Ёмкость от 650 Мбайт.

- **Накопитель WARM** (Write And Read Many times), позволяет производить многократную запись и считывание.

- **Накопитель WORM** (Write Once, Read Many times), позволяет производить однократную запись и многократное считывание.

Накопители на магнитной ленте (стримеры) и накопители на сменных дисках

Стример (англ. *tape streamer*) — устройство для резервного копирования больших объёмов информации. В качестве носителя здесь применяются кассеты с магнитной лентой ёмкостью 1 - 2 Гбайта и больше.

Стримеры позволяют записать на небольшую кассету с магнитной лентой огромное количество информации. **Встроенные в стример средства аппаратного сжатия позволяют автоматически уплотнять информацию перед её записью и восстанавливать после считывания, что увеличивает объём сохраняемой информации.**

Недостатком стримеров является их сравнительно *низкая скорость* записи, поиска и считывания информации.

В последнее время всё шире используются **накопители на сменных дисках**, которые позволяют не только увеличивать объём хранимой информации, но и переносить информацию между компьютерами. Объём сменных дисков — от сотен Мбайт до нескольких Гигабайт.

Устройство компьютера

Рассмотрим устройство компьютера на примере самой распространенной компьютерной системы — персонального компьютера.

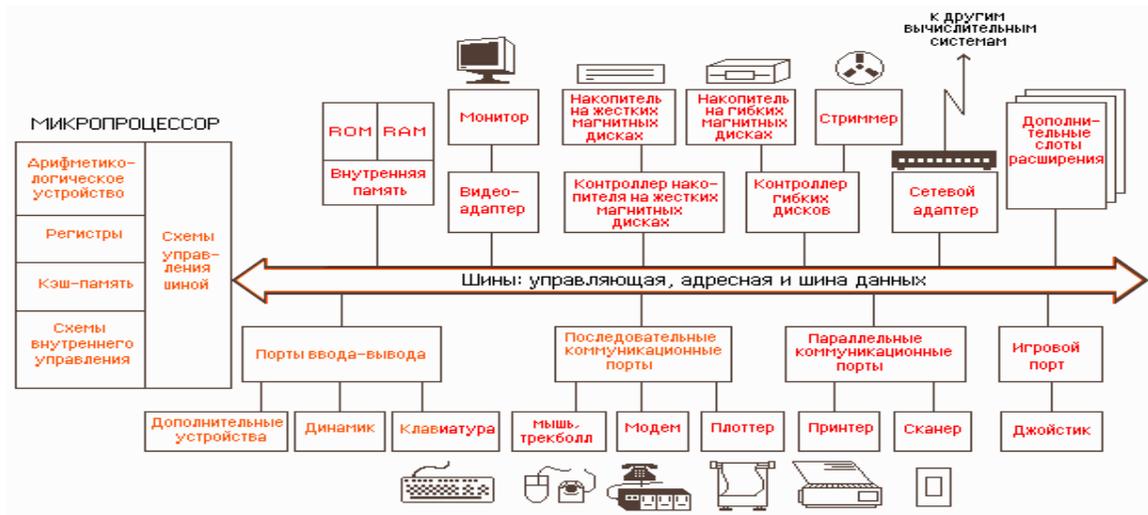
Персональным компьютером (ПК) называют сравнительно недорогой универсальный микрокомпьютер, рассчитанный на одного пользователя.

Персональные компьютеры обычно проектируются на основе принципа открытой архитектуры.

Принцип открытой архитектуры заключается в следующем:

- Регламентируются и стандартизируются только *описание принципа действия компьютера и его конфигурация* (определенная совокупность аппаратных средств и соединений между ними). Таким образом, *компьютер можно собирать из отдельных узлов и деталей, разработанных и изготовленных независимыми фирмами-изготовителями.*
- Компьютер *легко расширяется и модернизируется* за счёт наличия внутренних расширительных гнезд, в которые пользователь может вставлять разнообразные устройства, удовлетворяющие заданному стандарту, и тем самым *устанавливать конфигурацию своей машины в соответствии со своими личными предпочтениями.*

Упрощённая блок-схема, отражающая основные функциональные компоненты компьютерной системы в их взаимосвязи, изображена на рисунке 4.



Для того, чтобы соединить друг с другом различные устройства компьютера, они должны иметь одинаковый **интерфейс** (англ. *interface* от *inter* — между, и *face* — лицо).

Интерфейс — это средство сопряжения двух устройств, в котором все физические и логические параметры согласуются между собой.

Если интерфейс является общепринятым, например, утверждённым на уровне международных соглашений, то он называется **стандартным**.

Каждый из функциональных элементов (память, монитор или другое устройство) связан с шиной определённого типа — адресной, управляющей или шиной данных.

Для согласования интерфейсов периферийные устройства подключаются к шине не напрямую, а через свои **контроллеры** (адаптеры) и **порты** примерно по такой схеме:



Контроллеры и адаптеры представляют собой наборы электронных цепей, которыми снабжаются устройства компьютера с целью совместимости их интерфейсов. Контроллеры, кроме этого, осуществляют непосредственное управление периферийными устройствами по запросам микропроцессора.

Порты устройств представляют собой некие электронные схемы, содержащие один или несколько *регистров ввода-вывода* и позволяющие подключать периферийные устройства компьютера к внешним шинам микропроцессора.

Портами также называют *устройства стандартного интерфейса*: последовательный, параллельный и игровой порты (или интерфейсы).

Последовательный порт обменивается данными с процессором побайтно, а с внешними устройствами — побитно. *Параллельный порт* получает и посылает данные побайтно.

К *последовательному* порту обычно подсоединяют медленно действующие или достаточно удалённые устройства, такие, как мышь и модем. К *параллельному* порту подсоединяют более "быстрые" устройства — принтер и сканер. Через *игровой* порт подсоединяется джойстик. Клавиатура и монитор подключаются к своим *специализированным* портам, которые представляют собой просто *разъёмы*.

Основные электронные компоненты, определяющие архитектуру процессора, размещаются на основной плате компьютера, которая называется *системной* или *материнской (MotherBoard)*. А контроллеры и адаптеры дополнительных устройств, либо сами эти устройства, выполняются в виде *плат расширения (DaughterBoard* — дочерняя плата) и подключаются к шине с помощью *разъёмов расширения*, называемых также *слотами расширения* (англ. *slot* — щель, паз).

Основные блоки компьютера

Современный персональный компьютер состоит из нескольких основных конструктивных компонент:

- системного блока;
- монитора;
- клавиатуры;
- манипуляторов.

В системном блоке размещаются:

- блок питания;
- накопитель на жёстких магнитных дисках;
- накопитель на гибких магнитных дисках;

- системная плата;
- платы расширения;
- накопитель CD-ROM;
- и др.

Корпус системного блока может иметь горизонтальную (*DeskTop*) или вертикальную (*Tower* — башня) компоновку.

Вместо термина "*системный блок*" иногда употребляют термин "*платформа*".

Системная плата

Системная плата является основной в системном блоке. Она содержит компоненты, определяющие архитектуру компьютера:

- центральный процессор;
- постоянную (**ROM**) и оперативную (**RAM**) память, кэш-память;
- интерфейсные схемы шин;
- гнезда расширения;
- обязательные системные средства ввода-вывода и др.

Системные платы исполняются на основе наборов микросхем, которые называются **чипсетам** (*ChipSets*). Часто на системных платах устанавливают и контроллеры дисковых накопителей, видеоадаптер, контроллеры портов и др.

В гнезда расширения системной платы устанавливаются платы таких периферийных устройств, как модем, сетевая плата, видеоплата и т.п.

Аудиоадаптер

Аудиоадаптер (*Sound Blaster* или звуковая плата) это специальная электронная плата, которая позволяет записывать звук, воспроизводить его и создавать программными средствами с помощью микрофона, наушников, динамиков, встроенного синтезатора и другого оборудования.

Аудиоадаптер содержит в себе два преобразователя информации:

- **аналого-цифровой**, который преобразует непрерывные (то есть, аналоговые) звуковые сигналы (речь, музыку, шум) в цифровой двоичный код и записывает его на магнитный носитель;
- **цифро-аналоговый**, выполняющий обратное преобразование сохранённого в цифровом виде звука в аналоговый сигнал, который затем воспроизводится с помощью акустической системы, синтезатора звука или наушников.

Профессиональные звуковые платы позволяют выполнять сложную обработку звука, обеспечивают стереозвучание, имеют собственное ПЗУ с хранящимися в нём сотнями тембров звучаний различных музыкальных инструментов.

Звуковые файлы обычно имеют очень большие размеры. Так, трёхминутный звуковой файл со стереозвучанием занимает примерно 30 Мбайт памяти. Поэтому **платы Sound Blaster, помимо своих основных функций, обеспечивают автоматическое сжатие файлов.**

Область применения звуковых плат — компьютерные игры, обучающие программные системы, рекламные презентации, "голосовая почта" (*voice mail*) между компьютерами, озвучивание различных процессов, происходящих в компьютерном оборудовании, таких, например, как отсутствие бумаги в принтере и т.п.

Видеоадаптер и графический акселератор

Видеоадаптер — это электронная плата, которая обрабатывает видеоданные (текст и графику) и управляет работой дисплея. Содержит *видеопамять, регистры ввода вывода и модуль BIOS*. Посылает в дисплей сигналы управления яркостью лучей и сигналы развертки изображения.

Наиболее распространенный видеоадаптер на сегодняшний день — **адаптер SVGA** (*Super Video Graphics Array* — супервидеографический массив), **который может отображать на экране дисплея 1280x1024 пикселей при 256 цветах и 1024x768 пикселей при 16 миллионах цветов.**

С увеличением числа приложений, использующих сложную графику и видео, наряду с традиционными видеоадаптерами широко используются разнообразные **устройства компьютерной обработки видеосигналов**:

- **Графические акселераторы** (ускорители) — специализированные графические *сопроцессоры*, увеличивающие эффективность видеосистемы. Их применение освобождает центральный процессор от большого объема операций с видеоданными, так как акселераторы самостоятельно вычисляют, какие пиксели отображать на экране и каковы их цвета.

- **Фрейм-грабберы**, которые позволяют отображать на экране компьютера видеосигнал от видеомагнитофона, камеры, лазерного проигрывателя и т. п., с тем, чтобы **захватить нужный кадр в память и впоследствии сохранить его в виде файла**.

- **TV-тюнеры** — видеоплаты, превращающие компьютер в телевизор. TV-тюнер позволяет выбрать любую нужную телевизионную программу и отображать ее на экране в масштабируемом окне. Таким образом можно следить за ходом передачи, не прекращая работу.

Клавиатура

Клавиатура служит для ввода информации в компьютер и подачи управляющих сигналов. Она содержит стандартный набор алфавитно-цифровых клавиш и некоторые дополнительные клавиши — управляющие и функциональные, клавиши управления курсором, а также малую цифровую клавиатуру.

Курсор — светящийся символ на экране монитора, указывающий *позицию*, на которой будет отображаться следующий вводимый с клавиатуры знак.

Все символы, набираемые на клавиатуре, немедленно отображаются на мониторе в позиции курсора.

Наиболее распространена сегодня **101-клавишная клавиатура с раскладкой клавиш QWERTY** (читается “кверти”), названная так по клавишам, расположенным в верхнем левом ряду алфавитно-цифровой части клавиатуры:

Такая клавиатура имеет *12 функциональных клавиш*, расположенных вдоль верхнего края. **Нажатие функциональной клавиши приводит к посылке в компьютер не одного символа, а целой совокупности символов.**

Функциональные клавиши могут программироваться пользователем. Например, во многих программах для получения помощи (подсказки) задействована клавиша **F1**, а для выхода из программы — клавиша **F10**.

Управляющие клавиши имеют следующее назначение:

- **Enter** — клавиша **ввода**;
- **Esc** (Escape — выход) клавиша **для отмены** каких-либо действий, выхода из программы, из меню и т.п.;
- **Ctrl** и **Alt** — эти клавиши самостоятельного значения не имеют, но при нажатии совместно с другими управляющими клавишами изменяют их действие;
- **Shift** (регистр) — обеспечивает **смену регистра клавиш** (верхнего на нижний и наоборот);
- **Insert** (вставлять) — **переключает режимы вставки** (новые символы вводятся посреди уже набранных, раздвигая их) и **замены** (старые символы замещаются новыми);
- **Delete** (удалять) — **удаляет символ** с позиции курсора;
- **Back Space** или ← — **удаляет символ** перед курсором;
- **Home** и **End** — обеспечивают **перемещение курсора в первую и последнюю позицию строки**, соответственно;
- **Page Up** и **Page Down** — обеспечивают **перемещение по тексту на одну страницу** (один экран) назад и вперед, соответственно;
- **Tab** — **клавиша табуляции**, обеспечивает перемещение курсора вправо сразу на несколько позиций до очередной позиции табуляции;
- **Caps Lock** — фиксирует верхний регистр, обеспечивает **ввод прописных букв вместо строчных**;
- **Print Screen** — обеспечивает **печать информации**, видимой в текущий момент на экране.

- **Длинная нижняя клавиша** без названия — предназначена для **ввода пробелов**.

- Клавиши **↑**, **↓**, **←** и **→** служат для перемещения курсора **вверх, вниз, влево и вправо** на одну позицию или строку.

Малая цифровая клавиатура используется в двух режимах — **ввода чисел и управления курсором**. Переключение этих режимов осуществляется клавишей **Num Lock**.

Клавиатура содержит встроенный **микроконтроллер** (местное устройство управления), который выполняет следующие функции:

- **последовательно опрашивает клавиши, считывая введенный сигнал и вырабатывая двоичный *скан-код* клавиши;**
- **управляет световыми индикаторами клавиатуры;**
- **проводит внутреннюю диагностику неисправностей;**
- **осуществляет взаимодействие с центральным процессором через порт ввода-вывода клавиатуры.**

Клавиатура имеет **встроенный буфер** — **промежуточную память** малого размера, куда помещаются **введённые символы**. В случае переполнения буфера нажатие клавиши будет сопровождаться звуковым сигналом — это означает, что символ не введён (отвергнут).

Работу клавиатуры поддерживают специальные программы, "защитые" в **BIOS**, а также **драйвер клавиатуры**, который обеспечивает возможность ввода русских букв, управление скоростью работы клавиатуры и др.

Видеосистема компьютера

Видеосистема компьютера состоит из трех компонент:

- **монитор** (называемый также дисплеем);
- **видеоадаптер;**
- **программное обеспечение** (драйверы видеосистемы).

Видеоадаптер посылает в монитор сигналы управления яркостью лучей и синхросигналы строчной и кадровой развёрток. **Монитор** преобразует эти сигналы в зрительные образы. А **программные средства** обрабатывают видеоизображения — выполняют кодирование и декодирование сигналов, координатные преобразования, сжатие изображений и др.

Монитор — устройство визуального отображения информации (в виде текста, таблиц, рисунков, чертежей и др.).

подавляющее большинство мониторов сконструированы на базе **электронно-лучевой трубки** (ЭЛТ), и принцип их работы аналогичен принципу работы телевизора. Мониторы бывают алфавитно-цифровые и графические, монохромные и цветного изображения. Современные компьютеры комплектуются, как правило, цветными графическими мониторами.

Люминофор наносится в виде наборов точек трёх основных цветов — **красного, зелёного и синего**. Эти цвета называют основными, потому что их сочетаниями (в различных пропорциях) можно представить **любой цвет спектра**.

Наборы точек люминофора располагаются по треугольным триадам. Триада образует **пиксел** — точку, из которых формируется изображение (*англ. pixel — picture element*, элемент картинки).

Расстояние между центрами пикселов называется точечным шагом монитора. Это расстояние существенно влияет на чёткость изображения. **Чем меньше шаг, тем выше чёткость**. Обычно в цветных мониторах шаг составляет 0,28 мм. При таком шаге **глаз человека воспринимает точки триады как одну точку "сложного" цвета**.

На противоположной стороне трубки расположены три (по количеству основных цветов) **электронные пушки**. Все три пушки "нацелены" на один и тот же пиксел, но каждая из них излучает поток электронов в сторону "своей" точки люминофора.

Наряду с традиционными ЭЛТ-мониторами все шире используются **плоские жидкокристаллические** (ЖК) мониторы.

Жидкие кристаллы — это особое состояние некоторых органических веществ, в котором они обладают текучестью и свойством образовывать пространственные структуры, подобные кристаллическим. Жидкие кристаллы могут *изменять свою структуру и оптические свойства под действием электрического напряжения*. Меняя с помощью электрического поля ориентацию групп кристаллов и используя введённые в жидкокристаллический раствор вещества, способные излучать свет под воздействием электрического поля, можно создать высококачественные изображения, передающие более 15 миллионов цветовых оттенков.

Ранее для ЖК-мониторов использовали тонкую плёнку из жидких кристаллов, помещённую между двумя стеклянными пластинами. Заряды передавались через так называемую **пассивную матрицу** — сетку невидимых нитей, горизонтальных и вертикальных, создавая в месте пересечения нитей точку изображения (несколько размытого из-за того, что заряды проникают в соседние области жидкости).

Активные матрицы вместо нитей используют прозрачный экран из транзисторов и обеспечивают яркое, практически не имеющее искажений изображение. Панель состоит из независимых ячеек, каждая из которых состоит из четырех частей (для трёх основных цветов и одна резервная), каждая из которых управляется собственным транзистором.

По компактности такие мониторы не знают себе равных. Они занимают в 2 – 3 раза меньше места, чем мониторы с ЭЛТ и во столько же раз легче; потребляют гораздо меньше электроэнергии и **не излучают электромагнитных волн, воздействующих на здоровье людей**.

Разновидность монитора — **сенсорный экран**. Здесь **общение с компьютером осуществляется путём прикосновения пальцем к определённому месту чувствительного экрана. Этим выбирается необходимый режим из меню, показанного на экране монитора.**

Меню — это выведенный на экран монитора список различных вариантов работы компьютера, по которому можно сделать конкретный выбор.

Сенсорными экранами оборудуют рабочие места операторов и диспетчеров, их используют в информационно-справочных системах и т.д.

Принтер, плоттер, сканер

Принтер — печатающее устройство. Осуществляет вывод из компьютера закодированной информации в виде печатных копий текста или графики.

Существуют тысячи наименований принтеров. Но основных видов принтеров три: матричные, лазерные и струйные.

- **Матричные принтеры** используют комбинации маленьких штырьков, которые бьют по красящей ленте, благодаря чему на бумаге остаётся отпечаток символа. Каждый символ, печатаемый на принтере, формируется из набора 9, 18 или 24 игл, сформированных в виде вертикальной колонки. Недостатками этих недорогих принтеров являются их шумная работа и невысокое качество печати, приемлемое, в основном, для домашних целей.

- **Лазерные принтеры** работают примерно так же, как ксероксы. Компьютер формирует в своей памяти "образ" страницы текста и передает его принтеру. Информация о странице проецируется с помощью лазерного луча на вращающийся барабан со светочувствительным покрытием, меняющим электрические свойства в зависимости от освещённости.

После засветки на барабан, находящийся под электрическим напряжением, наносится красящий порошок — *тонер*, частицы которого налипают на засвеченные участки поверхности барабана. Принтер с помощью специального горячего валика протягивает бумагу под барабаном; тонер переносится на бумагу и "вплавляется" в неё, оставляя стойкое высококачественное изображение. *Цветные* лазерные принтеры пока очень дороги.

- **Струйные принтеры** генерируют символы в виде последовательности *чернильных точек*. Печатающая головка принтера имеет крошечные *сопла*, через которые на страницу выбрызгиваются быстросохнущие чернила. Эти принтеры требовательны к

качеству бумаги. **Цветные** струйные принтеры создают цвета, комбинируя чернила **четырёх** основных цветов — **ярко-голубого, пурпурного, желтого и черного**.

Принтер связан с компьютером посредством **кабеля** принтера, один конец которого вставляется своим разъёмом в **гнездо** принтера, а другой — в **порт** принтера компьютера. **Порт** — это разъём, через который можно соединить процессор компьютера с внешним устройством.

Каждый принтер обязательно имеет свой **драйвер** — программу, которая способна переводить (транслировать) стандартные команды печати компьютера в специальные команды, требующиеся для каждого принтера.

Плоттер (графопостроитель) — устройство, которое чертит графики, рисунки или диаграммы под управлением компьютера.

Плоттеры используются для получения сложных конструкторских чертежей, архитектурных планов, географических и метеорологических карт, деловых схем. **Плоттеры рисуют изображения с помощью пера**.

Роликовые плоттеры прокручивают бумагу под пером, а **планшетные плоттеры** перемещают перо через всю поверхность горизонтально лежащей бумаги.

Плоттеру, так же, как и принтеру, обязательно нужна специальная программа — **драйвер**, позволяющая прикладным программам передавать ему инструкции: поднять и опустить перо, провести линию заданной толщины и т.п.

Сканер — устройство для ввода в компьютер графических изображений. Создает оцифрованное изображение документа и помещает его в память компьютера.

Если принтеры выводят информацию из компьютера, то сканеры, наоборот, **переносят информацию с бумажных документов в память компьютера**. Существуют **ручные сканеры**, которые прокатывают по поверхности документа рукой, и **планшетные сканеры**, по внешнему виду напоминающие копировальные машины.

Если при помощи сканера вводится текст, компьютер воспринимает его как картинку, а не как последовательность символов. Для преобразования такого графического

ческого текста в обычный символьный формат используют **программы оптического распознавания образов.**

Модем и факс-модем

Модем — устройство для передачи компьютерных данных на большие расстояния по телефонным линиям связи.

Цифровые сигналы, вырабатываемые компьютером, нельзя напрямую передавать по телефонной сети, потому что она предназначена для передачи человеческой речи — непрерывных сигналов звуковой частоты.

Модем обеспечивает преобразование цифровых сигналов компьютера в переменный ток частоты звукового диапазона — этот процесс называется модуляцией, а также **обратное преобразование**, которое называется демодуляцией. Отсюда название устройства: *модем* — модулятор/демодулятор.

Для осуществления связи один модем вызывает другой по номеру телефона, а тот отвечает на вызов. Затем модемы посылают друг другу сигналы, согласуя подходящий им обоим *режим связи*. После этого передающий модем начинает *посылать модулированные данные* с согласованной скоростью (количеством бит в секунду) и форматом. Модем на другом конце *преобразует полученную информацию в цифровой вид* и передает её своему компьютеру. Закончив сеанс связи, модем отключается от линии.

Управление модемом осуществляется с помощью специального *коммутационного программного обеспечения*.

Модемы бывают внешние, выполненные в виде отдельного устройства, и внутренние, представляющие собой электронную плату, устанавливаемую внутри компьютера. Почти все модемы поддерживают и функции факсов.

Факс — это устройство факсимильной передачи изображения по телефонной сети. Название "*факс*" произошло от слова "*факсимиле*" (лат. *fac simile* — сделай по-

добное), означающее точное воспроизведение графического оригинала (подписи, документа и т.д.) средствами печати.

Модем, который может передавать и получать данные как факс, называется факс-модемом.

Манипуляторы

Манипуляторы (мышь, джойстик и др.) — это специальные устройства, которые используются для **управления курсором**.

Мышь имеет вид небольшой коробки, полностью уместящейся на ладони. Мышь связана с компьютером кабелем через специальный блок — *адаптер*, и её движения преобразуются в соответствующие перемещения курсора по экрану дисплея. В верхней части устройства расположены управляющие кнопки (обычно их три), позволяющие задавать начало и конец движения, осуществлять выбор меню и т.п.

Джойстик — обычно это стержень-ручка, отклонение которой от вертикального положения приводит к передвижению курсора в соответствующем направлении по экрану монитора. Часто применяется в компьютерных играх. В некоторых моделях в джойстик монтируется датчик давления. В этом случае, чем сильнее пользователь нажимает на ручку, тем быстрее движется курсор по экрану дисплея.

Трекбол — небольшая коробка с шариком, встроенным в верхнюю часть корпуса. Пользователь *рукой вращает шарик и перемещает, соответственно, курсор*. В отличие от мыши, трекбол не требует свободного пространства около компьютера, его можно встроить в корпус машины.

Дигитайзер — устройство для преобразования готовых изображений (чертежей, карт) в цифровую форму. Представляет собой плоскую панель — *планшет*, располагаемую на столе, и специальный инструмент — *перо*, с помощью которого указывается позиция на планшете. При перемещении пера по планшету фиксируются его координаты в близко расположенных точках, которые затем преобразуются в компьютере в требуемые единицы измерения.

Классификация компьютеров

Существуют различные классификации компьютерной техники:

- по этапам развития (по поколениям);
- по архитектуре;
- по производительности;
- по условиям эксплуатации;
- по количеству процессоров;
- по потребительским свойствам и т.д.

Четких границ между классами компьютеров не существует. По мере совершенствования структур и технологии производства, появляются новые классы компьютеров, границы существующих классов существенно изменяются. Деление компьютерной техники на поколения — весьма условная, нестрогая классификация вычислительных систем по степени развития аппаратных и программных средств, а также способов общения с компьютером.

Идея делить машины на поколения вызвана к жизни тем, что за время короткой истории своего развития компьютерная техника проделала большую эволюцию как в смысле **элементной базы** (*лампы, транзисторы, микросхемы* и др.), так и в смысле **изменения её структуры, появления новых возможностей, расширения областей применения и характера использования.**

К **первому поколению** обычно относят машины, созданные на рубеже 50-х годов. В их схемах использовались **электронные лампы**. Эти компьютеры были **огромными, неудобными и слишком дорогими машинами**, которые могли приобрести только крупные корпорации и правительства. Лампы потребляли огромное количество электроэнергии и выделяли много тепла.

Набор команд был небольшой, схема арифметико-логического устройства и устройства управления достаточно проста, программное обеспечение практически отсутствовало. Показатели объема оперативной памяти и быстродействия были

низкими. Для ввода-вывода использовались перфоленты, перфокарты, магнитные ленты и печатающие устройства.

Быстродействие порядка 10-20 тысяч операций в секунду.

Но это только техническая сторона. Очень важна и другая — способы использования компьютеров, стиль программирования, особенности математического обеспечения.

Программы для этих машин писались **на языке конкретной машины**. Математик, составивший программу, садился за пульт управления машины, вводил и отлаживал программы и производил по ним счет. Процесс отладки был наиболее длительным по времени.

Несмотря на ограниченность возможностей, эти машины позволили выполнить сложнейшие расчёты, необходимые для прогнозирования погоды, решения задач атомной энергетики и др.

Опыт использования машин первого поколения показал, что существует огромный разрыв между временем, затрачиваемым на разработку программ, и временем счета.

Эти проблемы начали преодолевать путем интенсивной **разработки средств автоматизации программирования, создания систем обслуживающих программ, упрощающих работу на машине и увеличивающих эффективность её использования**. Это, в свою очередь, потребовало значительных изменений в структуре компьютеров, направленных на то, чтобы приблизить её к требованиям, возникшим из опыта эксплуатации компьютеров.

Отечественные машины первого поколения: МЭСМ (малая электронная счётная машина), БЭСМ, Стрела, Урал, М-20.

Второе поколение компьютерной техники — машины, сконструированные примерно в 1955-65 гг. Характеризуются использованием в них как **электронных ламп**, так и **дискретных транзисторных логических элементов**. Их оперативная память была

построена на магнитных сердечниках. В это время стал расширяться диапазон применяемого оборудования ввода-вывода, появились высокопроизводительные **устройства для работы с магнитными лентами, магнитные барабаны и первые магнитные диски.**

Быстродействие — до сотен тысяч операций в секунду, **ёмкость памяти** — до нескольких десятков тысяч слов.

Появились так называемые **языки высокого уровня**, средства которых допускают описание всей необходимой последовательности вычислительных действий **в наглядном, легко воспринимаемом виде.**

Программа, написанная на алгоритмическом языке, непонятна компьютеру, воспринимающему только язык своих собственных команд. Поэтому специальные программы, которые называются **трансляторами**, переводят программу с языка высокого уровня на машинный язык.

Появился широкий набор библиотечных программ для решения разнообразных математических задач. Появились **мониторные системы**, управляющие режимом трансляции и исполнения программ. Из мониторных систем в дальнейшем выросли современные операционные системы.

Операционная система — важнейшая часть программного обеспечения компьютера, предназначенная для автоматизации планирования и организации процесса обработки программ, ввода-вывода и управления данными, распределения ресурсов, подготовки и отладки программ, других вспомогательных операций обслуживания.

Таким образом, ***операционная система является программным расширением устройства управления компьютера.***

Для некоторых машин второго поколения уже были созданы операционные системы с ограниченными возможностями.

Машинам второго поколения была свойственна **программная несовместимость**, которая затрудняла организацию крупных информационных систем. Поэтому в середи-

не 60-х годов наметился переход к созданию компьютеров, программно совместимых и построенных на микроэлектронной технологической базе.

Машины третьего поколения созданы примерно после 60-х годов. Поскольку процесс создания компьютерной техники шел непрерывно, и в нём участвовало множество людей из разных стран, имеющих дело с решением различных проблем, трудно и бесполезно пытаться установить, когда "поколение" начиналось и заканчивалось. Возможно, наиболее важным критерием различия машин второго и третьего поколений является критерий, основанный на понятии архитектуры.

Машины третьего поколения — это семейства машин с единой архитектурой, т.е. программно совместимых. В качестве элементной базы в них используются интегральные схемы, которые также называются микросхемами.

Машины третьего поколения имеют развитые операционные системы. Они обладают возможностями мультипрограммирования, т.е. одновременного выполнения нескольких программ. Многие задачи управления памятью, устройствами и ресурсами стала брать на себя операционная система или же непосредственно сама машина.

Примеры машин третьего поколения — семейства IBM-360, IBM-370, ЕС ЭВМ (Единая система ЭВМ), СМ ЭВМ (Семейство малых ЭВМ) и др.

Быстродействие машин внутри семейства изменяется от нескольких десятков тысяч до миллионов операций в секунду. Ёмкость оперативной памяти достигает нескольких сотен тысяч слов.

Четвёртое поколение — это теперешнее поколение компьютерной техники, разработанное после 1970 года.

Наиболее важный в концептуальном отношении критерий, по которому эти компьютеры можно отделить от машин третьего поколения, состоит в том, что машины четвёртого поколения проектировались в расчете на эффективное использование современных высокоуровневых языков и упрощение процесса программирования для конечного пользователя.

В аппаратном отношении для них характерно широкое использование **интегральных схем** в качестве элементной базы, а также наличие быстродействующих запоминающих устройств с произвольной выборкой ёмкостью в десятки мегабайт.

С точки зрения структуры машины этого поколения представляют собой **многопроцессорные** и **многомашинные комплексы**, работающие на общую память и общее поле внешних устройств. Быстродействие составляет до нескольких десятков миллионов операций в секунду, ёмкость оперативной памяти порядка 1 - 64 Мбайт.

Для них характерны:

- применение персональных компьютеров;
- телекоммуникационная обработка данных;
- компьютерные сети;
- широкое применение систем управления базами данных;
- элементы интеллектуального поведения систем обработки данных и устройств.

Разработка последующих поколений компьютеров производится на основе **больших интегральных схем повышенной степени интеграции**, использования оптоэлектронных принципов (**лазеры, голография**).

Развитие идет также по пути "**интеллектуализации**" компьютеров, устранения барьера между человеком и компьютером. Компьютеры будут способны воспринимать информацию с рукописного или печатного текста, с бланков, с человеческого голоса, узнавать пользователя по голосу, осуществлять перевод с одного языка на другой.

В компьютерах пятого поколения произойдет качественный переход от обработки данных к обработке знаний.

Архитектура компьютеров будущего поколения будет содержать два основных блока. Один из них — это **традиционный** компьютер. Но теперь он лишён связи с пользователем. Эту связь осуществляет блок, называемый термином "**интеллектуальный интерфейс**". Его задача — понять текст, написанный на естественном языке и содержащий условие задачи, и перевести его в работающую программу для компьютера.

Будет также решаться проблема децентрализации вычислений с помощью компьютерных сетей, как больших, находящихся на значительном расстоянии друг от друга, так и миниатюрных компьютеров, размещённых на одном кристалле полупроводника.

По условиям эксплуатации компьютеры делятся на два типа:

- офисные (универсальные);
- специальные.

Офисные предназначены для решения широкого класса задач при нормальных условиях эксплуатации.

Специальные компьютеры служат для решения более узкого класса задач или даже одной задачи, требующей многократного решения, и функционируют в особых условиях эксплуатации.

Машинные ресурсы специальных компьютеров часто ограничены. Однако их узкая ориентация позволяет реализовать заданный класс задач наиболее эффективно.

Специальные компьютеры управляют технологическими установками, работают в операционных или машинах скорой помощи, на ракетах, самолётах и вертолётах, вблизи высоковольтных линий передач или в зоне действия радаров, радиопередатчиков, в неотапливаемых помещениях, под водой на глубине, в условиях пыли, грязи, вибраций, взрывоопасных газов и т.п. Существует много моделей таких компьютеров.

Важное направление в создании промышленных компьютеров — разработка "*операторского интерфейса*" — пультов управления, дисплеев, клавиатур и указательных устройств во всевозможных исполнениях. От этих изделий напрямую зависит комфортность и результативность труда операторов.

По производительности и характеру использования компьютеры можно условно подразделить на:

- микрокомпьютеры, в том числе — персональные компьютеры;
- миникомпьютеры;
- мэйнфреймы (универсальные компьютеры);

- суперкомпьютеры.

Микрокомпьютеры — это компьютеры, в которых центральный процессор выполнен в виде микропроцессора.

Продвинутые модели микрокомпьютеров имеют несколько микропроцессоров. Производительность компьютера определяется не только характеристиками применяемого микропроцессора, но и ёмкостью оперативной памяти, типами периферийных устройств, качеством конструктивных решений и др.

Микрокомпьютеры представляют собой инструменты для решения разнообразных сложных задач. Их микропроцессоры с каждым годом увеличивают мощность, а периферийные устройства — эффективность. Быстродействие — порядка 1 - 10 миллионов операций в сек.

Разновидность микрокомпьютера — **микроконтроллер**. Это основанное на микропроцессоре специализированное устройство, встраиваемое в систему управления или технологическую линию.

Персональные компьютеры (ПК) — это микрокомпьютеры универсального назначения, рассчитанные на одного пользователя и управляемые одним человеком.

В класс персональных компьютеров входят различные машины — от **дешёвых** домашних и игровых с небольшой оперативной памятью, с памятью программы на кассетной ленте и обычным телевизором в качестве дисплея, до **сверхсложных** машин с мощным процессором, винчестерским накопителем ёмкостью в десятки Гигабайт, с цветными графическими устройствами высокого разрешения, средствами мультимедиа и другими дополнительными устройствами.

Персональный компьютер должен удовлетворять следующим требованиям:

- стоимость от нескольких сотен до 5-10 тысяч долларов;
- наличие внешних ЗУ на магнитных дисках;
- объём оперативной памяти не менее 4 Мбайт;
- наличие операционной системы;

- способность работать с программами на языках высокого уровня;
- ориентация на пользователя-непрофессионала (в простых моделях).

Миникомпьютерами и **суперминикомпьютерами** называются машины, конструктивно выполненные в одной стойке, т.е. занимающие объём порядка половины кубометра. Сейчас компьютеры этого класса вымирают, уступая место микрокомпьютерам.

Мэйнфреймы предназначены для решения широкого класса научно-технических задач и являются сложными и дорогими машинами. Их целесообразно применять в больших системах при наличии не менее 200 - 300 рабочих мест.

Централизованная обработка данных на мэйнфрейме обходится примерно в 5 - 6 раз дешевле, чем распределённая обработка при клиент-серверном подходе.

Известный мэйнфрейм *S/390* фирмы IBM обычно оснащается не менее чем тремя процессорами. Максимальный объём оперативного хранения достигает 342 Терабайт.

Производительность его процессоров, пропускная способность каналов, объём оперативного хранения позволяют наращивать число рабочих мест в диапазоне от 20 до 200000 с помощью простого добавления процессорных плат, модулей оперативной памяти и дисковых накопителей.

Десятки мэйнфреймов могут работать совместно под управлением одной операционной системы над выполнением единой задачи.

Суперкомпьютеры — это очень мощные компьютеры с производительностью свыше 100 мегафлопов (1 мегафлоп — миллион операций с плавающей точкой в секунду). Они называются **сверхбыстродействующими**. Эти машины представляют собой **многопроцессорные** и (или) **многомашинные** комплексы, работающие на общую память и общее поле внешних устройств. Различают суперкомпьютеры *среднего класса*, класса *выше среднего* и *переднего края (high end)*.

Архитектура суперкомпьютеров основана на идеях **параллелизма** и **конвейеризации вычислений**.

В этих машинах параллельно, то есть одновременно, выполняется множество похожих операций (это называется *мультипроцессорной обработкой*). Таким образом, сверхвысокое быстродействие обеспечивается *не для всех задач*, а только для задач, поддающихся распараллеливанию.

Что такое **конвейерная обработка**? Приведем сравнение — на каждом рабочем месте конвейера выполняется один шаг производственного процесса, а на всех рабочих местах в одно и то же время обрабатываются различные изделия на всевозможных стадиях. По такому принципу устроено арифметико-логическое устройство суперкомпьютера.

Отличительной особенностью суперкомпьютеров являются **векторные процессоры**, оснащенные аппаратурой для параллельного выполнения операций с многомерными цифровыми объектами — векторами и матрицами. В них встроены **векторные регистры** и **параллельный конвейерный механизм обработки**. Если на обычном процессоре программист выполняет операции над каждым компонентом вектора по очереди, то на векторном — выдаёт сразу векторные команды.

Векторная аппаратура очень дорога, в частности, потому, что требуется много сверхбыстродействующей памяти под векторные регистры.

Наиболее распространённые суперкомпьютеры — **массово-параллельные** компьютерные системы. Они имеют **десятки тысяч процессоров**, взаимодействующих через сложную, иерархически организованную систему памяти.

В качестве примера рассмотрим характеристики *многоцелевого массово-параллельного суперкомпьютера среднего класса Intel Pentium Pro 200*. Этот компьютер содержит 9200 процессоров Pentium Pro на 200 МГц, в сумме (теоретически) обеспечивающих производительность **1,34 Терафлон** (1 Терафлон равен 10^{12} операций с плавающей точкой в секунду), имеет 537 Гбайт памяти и диски ёмкостью 2,25 Терабайт. Система весит 44 тонны (кондиционеры для неё — целых 300 тонн) и потребляет мощность 850 кВт.

Супер-компьютеры используются для решения сложных и больших научных задач (метеорология, гидродинамика и т. п.), в управлении, разведке, в качестве централизованных хранилищ информации и т. д.

Элементная база — микросхемы сверхвысокой степени интеграции.

Портативные компьютеры обычно нужны руководителям предприятий, менеджерам, учёным, журналистам, которым приходится работать вне офиса — дома, на презентациях или во время командировок.

Основные разновидности портативных компьютеров:

Laptop (наколенник, от *lap* — колено и *top* — верх). По размерам близок к обычному портфелю. По основным характеристикам (быстродействие, память) примерно соответствует настольным ПК. Сейчас компьютеры этого типа уступают место ещё меньшим.

Notebook (блокнот, записная книжка). По размерам он ближе к книге крупного формата. Имеет вес около 3 кг. Помещается в портфель-дипломат. Для связи с офисом его обычно комплектуют *модемом*. Ноутбуки зачастую снабжают *приводами CD-ROM*.

Многие современные ноутбуки включают *взаимозаменяемые блоки со стандартными разъёмами*. Такие модули предназначены для очень разных функций. В одно и то же гнездо можно по мере надобности вставлять привод компакт-дисков, накопитель на магнитных дисках, запасную батарею или съёмный винчестер. Ноутбук *устойчив к сбоям в энергопитании*. Даже если он получает энергию от обычной электросети, в случае какого-либо сбоя он мгновенно переходит на питание от аккумуляторов.

Palmtop (наладонник) — самые маленькие современные персональные компьютеры. Умещаются на ладони. Магнитные диски в них заменяет энергонезависимая электронная память. Нет и накопителей на дисках — обмен информацией с обычными компьютерами идет линиям связи. Если Palmtop дополнить набором деловых программ, записанных в его постоянную память, получится *персональный цифровой помощник (Personal Digital Assistant)*.

Как выбрать компьютер

Чтобы купить компьютер и затем успешно им пользоваться, необязательно досконально знать принципы его работы и из каких частей он состоит. Главное — представлять, каковы возможности ПК и как ими пользоваться. Покупая компьютер, решите для себя, для чего он вам нужен и какой уровень качества может вас удовлетворить.

Главные параметры компьютера

Какие же компоненты компьютера определяют его качество и возможности? Если вы покупаете новый, современный ПК, к которому не предъявляется особых требований (это не сервер и не графическая станция), то его главные параметры — качество изображения монитора, объем жесткого диска и марка видеокарты. Остальное — тип и частота процессора, модель системной платы, марка CD-дисковода и т. д. — тоже имеет значение, но характеристики современных комплектующих достигли на сегодня таких высот, когда большинство покупателей могут удовлетворить даже недорогие компоненты начального уровня. Например, скорость самого простого процессора Intel Celeron 1000 достаточна для любого делового и домашнего использования. Дорогие процессоры вроде Pentium 4 и Athlon XP работают намного быстрее, но их производительность во многих случаях оказывается избыточной.

Конфигурация

Итак, каков же “джентльменский набор” современного компьютера? Сначала о том, какие устройства нужны для выполнения разных функций, а потом о конкретных марках этих устройств. Любой полноценный персональный компьютер должен состоять как минимум из следующих компонентов: процессора, материнской платы, оперативной памяти, винчестера, видеокарты, корпуса с блоком питания, клавиатуры и, конечно же, монитора. Без остальных устройств компьютер, конечно, работать будет, но вот пользоваться им вряд ли будет удобно.

Поэтому прибавим к списку манипулятор “мышь”, дисковод флоппи-дисков, дисковод CD-ROM, а также сетевую карту (для офисных ПК) и модем (если у вас не выделенная линия). С такой комплектацией уже можно работать, а вот чтобы полноценно от-

дышать, например, играть и слушать музыку, потребуется как минимум звуковая карта и акустические системы.

Для более комфортного времяпровождения и максимально плодотворной деятельности существуют такие, например, устройства, как DVD-ROM, пишущий CD-дисковод, TV-тюнер (просмотр телепередач, возможность “оцифровывать” изображение), FM-тюнер для приема радиопередач и пр.

Характеристики комплектующих

Процессор. Наилучшими параметрами обладают сейчас недорогие процессоры Intel Celeron IV, более быстрые, но и более дорогие Intel Pentium III и Pentium 4. Желательно, чтобы тактовая частота процессора была не ниже 1000 МГц.

Материнская плата. Отличные характеристики у плат на основе микросхем Intel (i815, i845, i845D, i850) и VIA(КТ266, КТ333, КТ400), SiS (645 ,648DX). Лучше всего среди производителей системных плат зарекомендовали себя ASUS, ECS, Intel, Gigabyte.

Оперативная память. Для работы с офисными программами и не самыми требовательными играми достаточно 128 Мб памяти. Более высокая скорость достигается при объеме 256 Мб, а в некоторых случаях (сложные игры, графика) рекомендуется иметь 512 Мб или больше. Самые известные производители модулей памяти: Samsung (SEC), Micron, Siemens, LG (LGS), Hitachi, Hynix.

Винчестер. Современные модели имеют емкость как минимум 40 Гбайт, а средняя емкость 30-80 Гб. Если вы не собираетесь записывать музыку, фото или - видеоизображения, то вам хватит и 20 Гб. Если не будете экономить место на винчестере, то лучше выбрать HDD емкостью не меньше 40 Гбайт. Среди производителей винчестеров лучше всего зарекомендовали себя IBM и Maxtor, Seagate, WD.

Дисковод CD-ROM. Скорость достаточно высока у всех современных моделей, причем у самых быстрых (48-х и более) иногда наблюдается разрушение “пиратских”

дисков внутри дисководов. Так что за скоростью сильно гнаться не стоит. Наиболее качественные CD-приводы выпускают TEAC, NEC, Samsung, LG, Pioneer, Sony, Toshiba и др.

Монитор. От выбора монитора зависит не только качество изображения, но и здоровье работающего с ним человека. Наиболее четкое и контрастное изображение имеют мониторы с кинескопами типа Sony Trinitron, или похожие на них LG Flatron, ViewSonic, SonicTron, Mitsubishi Diamondtron, NEC CromaClear. Однако цена у мониторов с такими кинескопами довольно высокая, а разрешающая способность по горизонтали не самая лучшая. Кинескопы другого типа — с теневой маской — стоят дешевле, но тоже имеют весьма качественное изображение. Для недорогого компьютера обычно приобретаются мониторы с диагональю 15". Для игрового ПК или работы с графикой лучше 17" модель. Мониторы большого размера — 19", 20" или 21" — используются в основном в сферах компьютерного дизайна и автоматизированного проектирования. Если хотите быть уверены в безопасности монитора, покупайте модели, соответствующие самому жесткому стандарту — TCO-99.

И последний совет: если вы все же чувствуете неуверенность при выборе компьютера, если вам что-то непонятно — не стесняйтесь, спрашивайте у менеджеров и консультантов компьютерного салона. Компьютер — непростое устройство. А изменения в этой сфере происходят так быстро, что даже не все специалисты разбираются в текущей ситуации на рынке.

Программное обеспечение компьютеров

Под программным обеспечением (Software) понимается совокупность программ, выполняемых вычислительной системой.

К программному обеспечению (ПО) относится также вся **область деятельности по проектированию и разработке ПО:**

- технология проектирования программ (например, нисходящее проектирование, структурное и объектно-ориентированное проектирование и др.);
- методы тестирования программ ;

- методы доказательства правильности программ;
- анализ качества работы программ;
- документирование программ;
- разработка и использование программных средств, облегчающих процесс проектирования программного обеспечения, и многое другое.

Программное обеспечение – *неотъемлемая часть компьютерной системы*. Оно является логическим продолжением технических средств. Сфера применения конкретного компьютера определяется созданным для него ПО.

Сам по себе компьютер не обладает знаниями ни в одной области применения. Все эти знания сосредоточены в выполняемых на компьютерах программах.

Программное обеспечение современных компьютеров включает миллионы программ — от игровых до научных.

В первом приближении все программы, работающие на компьютере, можно условно разделить на **три категории** (рис. 6.1):

1. **прикладные программы**, непосредственно обеспечивающие выполнение необходимых пользователям работ;
2. **системные программы**, выполняющие различные вспомогательные функции, например:
 - управление ресурсами компьютера;
 - создание копий используемой информации;
 - проверка работоспособности устройств компьютера;
 - выдача справочной информации о компьютере и др.;
3. **инструментальные программные системы**, облегчающие процесс создания новых программ для компьютера.

При построении классификации ПО нужно учитывать тот факт, что стремительное развитие вычислительной техники и расширение сферы приложения компьютеров резко ускорили процесс эволюции программного обеспечения.

Если раньше можно было по пальцам перечислить основные категории ПО — операционные системы, трансляторы, пакеты прикладных программ, то сейчас ситуация коренным образом изменилась.

Развитие ПО пошло как вглубь (появились новые подходы к построению операционных систем, языков программирования и т.д.), так и вширь (прикладные программы перестали быть прикладными и приобрели самостоятельную ценность).

Соотношение между требующимися программными продуктами и имеющимися на рынке меняется очень быстро. Даже классические программные продукты, такие, как операционные системы, непрерывно развиваются и наделяются интеллектуальными функциями, многие из которых ранее относились только к интеллектуальным возможностям человека.

Кроме того, появились нетрадиционные программы, классифицировать которые по устоявшимся критериям очень трудно, а то и просто невозможно, как, например, программа — *электронный собеседник*.

На сегодняшний день можно сказать, что **более или менее определённо сложились следующие группы программного обеспечения:**

- операционные системы и оболочки;
- системы программирования (трансляторы, библиотеки подпрограмм, отладчики и т.д.);
- инструментальные системы;
- интегрированные пакеты программ;
- динамические электронные таблицы;
- системы машинной графики;
- системы управления базами данных (СУБД);
- прикладное программное обеспечение.

Структура программного обеспечения показана на рис. 6. Разумеется, эту **классификацию нельзя считать исчерпывающей**, но она более или менее наглядно отражает направления совершенствования и развития программного обеспечения.

Прикладная программа — это любая конкретная программа, способствующая решению какой-либо задачи в пределах данной проблемной области.

Например, там, где на компьютер возложена задача контроля за финансовой деятельностью какой-либо фирмы, прикладной будет программа подготовки платежных ведомостей.

Прикладные программы могут носить и общий характер, например, обеспечивать составление и печатание документов и т.п.

В противоположность этому, операционная система или инструментальное ПО не вносят прямого вклада в удовлетворение конечных потребностей пользователя.

Прикладные программы могут использоваться либо автономно, то есть решать поставленную задачу без помощи других программ, либо в составе программных комплексов или пакетов.

Системные программы выполняются вместе с прикладными и служат для управления ресурсами компьютера — центральным процессором, памятью, вводом-выводом.

Это программы общего пользования, которые **предназначены для всех пользователей компьютера**. Системное программное обеспечение разрабатывается так, чтобы компьютер мог эффективно выполнять прикладные программы.

Среди десятков тысяч системных программ особое место занимают **операционные системы**, которые обеспечивают управление **ресурсами компьютера** с целью их эффективного использования. Важными классами системных программ являются также программы вспомогательного назначения — **утилиты** (лат. *utilitas* — польза). Они либо **расширяют и дополняют соответствующие возможности операционной системы**, либо **решают самостоятельные важные задачи**. Кратко опишем некоторые разновидности утилит:

- **программы контроля, тестирования и диагностики**, которые используются для проверки правильности функционирования устройств компьютера и для обнаружения неисправностей в процессе эксплуатации; указывают причину и место неисправности;
- **программы-драйверы**, которые расширяют возможности операционной системы по управлению устройствами ввода-вывода, оперативной памятью и т.д.; с помощью драйверов возможно подключение к компьютеру новых устройств или нестандартное использование имеющихся;
- **программы-упаковщики** (архиваторы), которые позволяют записывать информацию на дисках более плотно, а также объединять копии нескольких файлов в один архивный файл;
- **антивирусные программы**, предназначенные для предотвращения заражения компьютерными вирусами и ликвидации последствий заражения вирусами;
- **программы оптимизации и контроля качества дискового пространства** ;
- **программы восстановления информации, форматирования, защиты данных** ;
- **коммуникационные программы**, организующие обмен информацией между компьютерами;
- **программы для управления памятью**, обеспечивающие более гибкое использование оперативной памяти;
- **программы для записи CD-ROM, CD-R** и многие другие.

Часть утилит входит в состав операционной системы, а другая часть функционирует независимо от нее, т.е. автономно.

Операционная система — это комплекс взаимосвязанных системных программ, назначение которого — организовать взаимодействие пользователя с компьютером и выполнение всех других программ.

Операционная система выполняет роль связующего звена между аппаратурой компьютера, с одной стороны, и выполняемыми программами, а также пользователем, с другой стороны.

Операционная система обычно хранится во внешней памяти компьютера — *на диске*. При включении компьютера она считывается с дисковой памяти и размещается в ОЗУ.

Этот процесс называется *загрузкой операционной системы*.

В функции операционной системы входит:

- осуществление диалога с пользователем;
- ввод-вывод и управление данными;
- планирование и организация процесса обработки программ;
- распределение ресурсов (оперативной памяти и кэша, процессора, внешних устройств);
- запуск программ на выполнение;
- всевозможные вспомогательные операции обслуживания;
- передача информации между различными внутренними устройствами;
- программная поддержка работы периферийных устройств (дисплея, клавиатуры, дисковых накопителей, принтера и др.).

Операционную систему можно назвать программным продолжением устройства управления компьютера. Операционная система скрывает от пользователя сложные ненужные подробности взаимодействия с аппаратурой, образуя прослойку между ними. В результате этого люди освобождаются от очень трудоёмкой работы по организации взаимодействия с аппаратурой компьютера.

В зависимости от количества одновременно обрабатываемых задач и числа пользователей, которых могут обслуживать ОС, различают четыре основных класса операционных систем:

1. **однопользовательские однозадачные**, которые поддерживают одну клавиатуру и могут работать только с одной (в данный момент) задачей;
2. **однопользовательские однозадачные с фоновой печатью**, которые позволяют помимо основной задачи запускать одну дополнительную задачу, ориентированную, как правило, на вывод информации на печать. Это ускоряет работу при выдаче больших объёмов информации на печать;
3. **однопользовательские многозадачные**, которые обеспечивают одному пользователю параллельную обработку нескольких задач. Например, к одному компьютеру можно подключить несколько принтеров, каждый из которых будет работать на "свою" задачу;
4. **многопользовательские многозадачные**, позволяющие на одном компьютере запускать несколько задач нескольким пользователям. Эти ОС очень сложны и требуют значительных машинных ресурсов.

В различных моделях компьютеров используют операционные системы с разной архитектурой и возможностями. Для их работы требуются разные ресурсы. Они предоставляют разную степень сервиса для программирования и работы с готовыми программами.

Операционная система для персонального компьютера, ориентированного на профессиональное применение, должна содержать следующие основные компоненты:

- программы управления вводом/выводом;
- программы, управляющие файловой системой и планирующие задания для компьютера;
- процессор командного языка, который принимает, анализирует и выполняет команды, адресованные операционной системе.

Каждая операционная система имеет свой **командный язык**, который позволяет пользователю выполнять те или иные действия:

- обращаться к каталогу;
- выполнять разметку внешних носителей;
- запускать программы;

- ... другие действия.

Анализ и исполнение команд пользователя, включая загрузку готовых программ из файлов в оперативную память и их запуск, осуществляет **командный процессор** операционной системы.

Для управления внешними устройствами компьютера используются специальные системные программы — **драйверы**. Драйверы стандартных устройств образуют в совокупности **базовую систему ввода-вывода** (BIOS), которая обычно заносится в постоянное ЗУ компьютера.

Состав инструментальных средств программирования

Транслятор (англ. *translator* — переводчик) — это программа-переводчик. Она преобразует программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд.

Трансляторы реализуются в виде компиляторов или интерпретаторов. С точки зрения выполнения работы компилятор и интерпретатор существенно различаются.

Компилятор (англ. *compiler* — составитель, собиратель) читает всю программу *целиком*, делает ее перевод и создает законченный вариант программы на машинном языке, который затем и выполняется.

Интерпретатор (англ. *interpreter* — истолкователь, устный переводчик) переводит и выполняет программу *строка за строкой*.

После того, как программа откомпилирована, ни сама исходная программа, ни компилятор более не нужны. В то же время программа, обрабатываемая интерпретатором, должна заново *переводиться* на машинный язык при каждом очередном запуске программы.

Откомпилированные программы работают быстрее, но *интерпретируемые* проще исправлять и изменять.

Каждый конкретный язык ориентирован либо на компиляцию, либо на интерпретацию — в зависимости от того, для каких целей он создавался. Например, *Паскаль* обычно используется для решения довольно сложных задач, в которых важна скорость работы программ. Поэтому данный язык обычно реализуется с помощью *компилятора*.

С другой стороны, *Бейсик* создавался как язык для начинающих программистов, для которых построчное выполнение программы имеет неоспоримые преимущества.

Иногда для одного языка имеется *и компилятор, и интерпретатор*. В этом случае для разработки и тестирования программы можно воспользоваться интерпретатором, а затем откомпилировать отлаженную программу, чтобы повысить скорость ее выполнения.

Система программирования — это система для разработки новых программ на конкретном языке программирования.

Современные системы программирования обычно предоставляют пользователям **мощные и удобные средства разработки программ**. В них входят:

- компилятор или интерпретатор;
- интегрированная среда разработки;
- средства создания и редактирования текстов программ;
- обширные библиотеки стандартных программ и функций;
- отладочные программы, т.е. программы, помогающие находить и устранять ошибки в программе;
- "дружественная" к пользователю диалоговая среда;
- многооконный режим работы;
- мощные графические библиотеки; утилиты для работы с библиотеками
- встроенный ассемблер;
- встроенная справочная служба;
- другие специфические особенности.

Популярные системы программирования – *Turbo Basic, Quick Basic, Turbo Pascal, Turbo C*.

В последнее время получили распространение системы программирования, ориентированные на создание *Windows-приложений*:

- пакет ***Borland Delphi*** (Дельфи) — блестящий наследник семейства компиляторов Borland Pascal, предоставляющий качественные и очень удобные средства визуальной разработки. Его исключительно быстрый компилятор позволяет эффективно и быстро решать практически любые задачи прикладного программирования.
- пакет ***Microsoft Visual Basic*** — удобный и популярный инструмент для создания Windows-программ с использованием визуальных средств. Содержит инструментарий для создания *диаграмм* и *презентаций*.
- пакет ***Borland C++*** — одно из самых распространённых средств для разработки DOS и Windows приложений.

Инструментальные программные средства — это программы, которые используются в ходе разработки, корректировки или развития других прикладных или системных программ.

По своему назначению они близки системам программирования. К инструментальным программам, например, относятся:

- редакторы;
- средства компоновки программ;
- отладочные программы, т.е. программы, помогающие находить и устранять ошибки в программе;
- вспомогательные программы, реализующие часто используемые системные действия;
- графические пакеты программ и т.п.

Инструментальные программные средства могут оказать помощь на всех стадиях разработки ПО.

Текстовый редактор — это программа, используемая специально для ввода и редактирования текстовых данных.

Этими данными могут быть программа или какой-либо документ или же книга. Редактируемый текст выводится на экран, и пользователь может в диалоговом режиме вносить в него свои изменения.

Текстовые редакторы могут обеспечивать выполнение разнообразных функций, а именно:

- редактирование строк текста;
- возможность использования различных шрифтов символов;
- копирование и перенос части текста с одного места на другое или из одного документа в другой;
- контекстный поиск и замена частей текста;
- задание произвольных межстрочных промежутков;
- автоматический перенос слов на новую строку;
- автоматическая нумерацию страниц;
- обработка и нумерация сносок;
- выравнивание краев абзаца;
- создание таблиц и построение диаграмм;
- проверка правописания слов и подбор синонимов;
- построение оглавлений и предметных указателей;
- распечатка подготовленного текста на принтере в нужном числе экземпляров и т.п.

Возможности текстовых редакторов различны — от программ, предназначенных для подготовки небольших документов простой структуры, до программ для набора, оформления и полной подготовки к типографскому изданию книг и журналов (издательские системы).

Полнофункциональные издательские системы — *Microsoft Publisher*, *Corel Ventura* и *Adobe PageMaker*. Издательские системы незаменимы для компьютерной верстки и графики. Значительно облегчают работу с многостраничными документами, имеют возможности автоматической разбивки текста на страницы, расстановки номеров страниц, создания заголовков и т.д. Создание макетов любых изданий — от рекламных

листочков до многостраничных книг и журналов — становится очень простым, даже для новичков.

Графический редактор — это программа, предназначенная для автоматизации процессов построения на экране дисплея графических изображений. Предоставляет возможности рисования линий, кривых, раскраски областей экрана, создания надписей различными шрифтами и т.д.

Большинство редакторов позволяют обрабатывать изображения, полученные с помощью *сканеров*, а также выводить картинки в таком виде, чтобы они могли быть включены в документ, подготовленный с помощью текстового редактора.

Некоторые редакторы позволяют получать изображения трёхмерных объектов, их сечений, разворотов, каркасных моделей и т.п.

Пользуется известностью ***Corel DRAW!*** — мощный графический редактор с функциями создания публикаций, снабжённый инструментами для редактирования графики и трёхмерного моделирования.

Системы деловой графики дают возможность выводить на экран **различные виды графиков и диаграмм** (рис 6.7):

- гистограммы;
- круговые и секторные диаграммы и т.д.

Системы научной и инженерной графики позволяют в цвете и в заданном масштабе отображать на экране следующие объекты:

- графики двумерных и трехмерных функций, заданных в табличном или аналитическом виде;
- системы изолиний, в том числе, и нанесённые на поверхность объекта;
- сечения, проекции, карты и т.д.

Для построения легко воспринимаемых реалистических изображений трёхмерных объектов системы инженерной графики позволяют удалять линии, не видимые наблюдателю.

Табличный процессор — это комплекс взаимосвязанных программ, предназначенный для обработки электронных таблиц.

Электронная таблица — это компьютерный эквивалент обычной таблицы, состоящей из строк и граф, на пересечении которых располагаются клетки, в которых содержится числовая информация, формулы или текст.

Значение в числовой клетке таблицы может быть либо записано, либо рассчитано по соответствующей формуле; в формуле могут присутствовать обращения к другим клеткам (рис. 6.10).

Каждый раз при изменении значения в клетке таблицы в результате записи в нее нового значения с клавиатуры *пересчитываются* также значения во всех тех клетках, в которых стоят величины, зависящие от данной клетки.

Графам и строкам можно присваивать наименования. Экран монитора трактуется как окно, через которое можно рассматривать таблицу целиком или по частям.

Табличные процессоры представляют собой удобное средство для проведения бухгалтерских и статистических расчетов. В каждом пакете имеются сотни встроенных математических функций и алгоритмов статистической обработки данных. Кроме того, имеются мощные средства для связи таблиц между собой, создания и редактирования электронных баз данных.

Специальные средства позволяют автоматически получать и распечатывать настраиваемые отчеты с использованием десятков различных типов таблиц, графиков, диаграмм, снабжать их комментариями и графическими иллюстрациями.

Табличные процессоры имеют встроенную справочную систему, предоставляющую пользователю информацию по конкретным командам меню и другие справочные

данные. Многомерные таблицы позволяют быстро делать выборки в базе данных по любому критерию.

Самые популярные табличные процессоры — *Microsoft Excel (Эксель)* и *Lotus 1-2-3*.

В *Microsoft Excel* автоматизированы многие рутинные операции, специальные шаблоны помогают создавать отчёты, импортировать данные и многое другое.

Lotus 1-2-3 — профессиональный процессор электронных таблиц. Широкие графические возможности и удобный интерфейс пакета позволяют быстро ориентироваться в нём. С его помощью можно создать любой финансовый документ, отчёт для бухгалтерии, составить бюджет, а затем разместить все эти документы в базах данных.

Система управления базами данных (СУБД) — это система программного обеспечения, позволяющая обрабатывать обращения к базе данных, поступающие от прикладных программ конечных пользователей.

Системы управления базами данных позволяют *объединять* большие объёмы информации и *обрабатывать* их, *сортировать*, *делать выборки* по определённым критериям и т.п.

Современные СУБД дают возможность включать в них не только *текстовую* и *графическую* информацию, но и звуковые фрагменты и даже видеоклипы.

Простота использования СУБД позволяет создавать новые базы данных, не прибегая к программированию, а пользуясь только встроенными функциями.

СУБД обеспечивают *правильность*, *полноту* и *непротиворечивость* данных, а также *удобный доступ* к ним.

Популярные СУБД — *FoxPro*, *Access for Windows*, *Paradox*.

Для менее сложных применений вместо СУБД используются **информационно-поисковые системы** (ИПС), которые выполняют следующие функции:

- *хранение* большого объема информации;
- быстрый *поиск* требуемой информации;
- *добавление, удаление и изменение* хранимой информации;
- *вывод* ее в удобном для человека виде.

Библиотека стандартных подпрограмм — это совокупность подпрограмм, составленных на одном из языков программирования и удовлетворяющих определенным единым требованиям к структуре, организации их входов и выходов, описаниям подпрограмм и т.п.

Стандартные подпрограммы имеют единую форму обращения, что обеспечивает простоту и удобство настройки параметров подпрограммы на решение конкретной задачи.

В качестве примера можно привести библиотеку стандартных подпрограмм по численным математическим методам решения уравнений, вычисления интегралов, нахождения экстремумов и т.п.

Интегрированные пакеты представляют собой набор нескольких программных продуктов, объединенных в единый удобный инструмент. Наиболее развитые из них включают в себя текстовый редактор, органайзер, электронную таблицу, СУБД, средства поддержки электронной почты, программу создания презентационной графики.

Результаты, полученные отдельными подпрограммами, могут быть объединены в окончательный документ, содержащий табличный, графический и текстовый материал.

Интегрированные пакеты, как правило, содержат некоторое ядро, обеспечивающее возможность тесного взаимодействия между составляющими.

Пример: **интегрированный пакет для написания книг, содержащих иллюстрации**. Он содержит:

- текстовый редактор;
- орфографический корректор на 80000 слов (программу обнаружения орфографических ошибок);

- программу слияния текстов;
- программу формирования оглавлений и составления указателей;
- автоматический поиск и замену слов и фраз;
- средства телекоммуникации;
- электронную таблицу;
- систему управления базами данных;
- модули графического оформления;
- графический редактор;
- возможность печати сотнями разных шрифтов и т.д.

Наиболее известные интегрированные пакеты:

Microsoft Office. В этот мощный профессиональный пакет вошли такие необходимые программы, как текстовый редактор **WinWord**, электронная таблица **Excel**, программа создания презентаций **PowerPoint**, СУБД **Access**, средство поддержки электронной почты **Mail**. Мало того, все части этого пакета составляют единое целое, и даже внешне все программы выглядят единообразно, что облегчает как их освоение, так и ежедневное использование.

Microsoft Works — это очень простой и удобный пакет, объединяющий в себе текстовый редактор, электронные таблицы и базы данных, а также телекоммуникационные средства для соединения с другими компьютерами по телефонным линиям. Пакет ориентирован на людей, не имеющих времени осваивать сложные продукты, на начинающих пользователей, а также на домашних пользователей.

ПОДГОТОВКА ПРОГРАММНЫХ СРЕДСТВ ДЛЯ РЕШЕНИЯ ЭКОНОМИЧЕСКИХ ЗАДАЧ

ОБЩАЯ ХАРАКТЕРИСТИКА

ТЕХНОЛОГИИ СОЗДАНИЯ

ПРИКЛАДНЫХ ПРОГРАММНЫХ СРЕДСТВ

Решение задачи на ЭВМ - это процесс получения результатной информации на основе обработки исходной информации с помощью программы, составленной из команд системы управления вычислительной машины. Сама программа представляет собой формали-

зованное описание последовательности действий определенных устройств ЭВМ в зависимости от конкретного характера задачи.

Решение задач с помощью компьютера включает в себя следующие основные этапы, часть из которых осуществляется без участия компьютера.

1. **Постановка задачи:**

- сбор информации о задаче;
- формулировка условия задачи;
- определение конечных целей решения задачи;
- определение формы выдачи результатов;
- описание данных (их типов, диапазонов величин, структуры и

т.п.).

2. **Анализ и исследование задачи, модели:**

- анализ существующих аналогов;
- анализ технических и программных средств;
- разработка математической модели;
- разработка структур данных.

3. **Разработка алгоритма:**

- выбор метода проектирования алгоритма;
- выбор формы записи алгоритма (блок-схемы, псевдокод и др.);
- выбор тестов и метода тестирования;
- проектирование алгоритма.

4. **Программирование:**

- выбор языка программирования;
- уточнение способов организации данных;
- запись алгоритма на выбранном языке программирования.

5. **Тестирование и отладка:**

- синтаксическая отладка;
- отладка семантики и логической структуры;
- тестовые расчеты и анализ результатов тестирования;
- совершенствование программы.

6. **Анализ результатов решения задачи и уточнение** в случае необхо-

димости математической модели с повторным выполнением этапов 2 - 5.

7. Сопровождение программы:

- доработка программы для решения конкретных задач;
- составление документации к решенной задаче, к математической модели, к алгоритму, к программе, к набору тестов, к использованию.

Технология разработки программ решения задачи определяется главным образом двумя факторами:

осуществляется ли разработка программы решения задачи как составного элемента единой системы автоматизированной обработки информации либо как относительно независимой, локальной компоненты общего программного комплекса, обеспечивающего решение на ЭВМ задач управления;

какие программно-инструментальные средства используются для разработки и реализации задач на ЭВМ.

Под программно-инструментальными средствами будем понимать компоненты ПО, позволяющие программировать решение задач управления. К программно-инструментальным средствам в первую очередь относятся алгоритмические языки и соответствующие им трансляторы, затем СУБД с языковыми средствами программирования в их среде, электронные таблицы со средствами их настройки и т.п.

алгоритмическими языками программно-инструментальных средств коренным образом изменило традиционное представление о процессе программирования и программе. В связи с этим в толковом словаре по информатике под термином "программа" наряду с приведенным выше определением понимаются также и "данные, предназначенные для управления конкретными компонентами системы обработки данных в целях реализации определенного алгоритма"¹.

На наш взгляд, лучше употреблять термин "программное средство" (или программное приложение), тем более что эти термины находят широкое употребление среди профессиональных программистов. Таким образом, под программным средством (программным приложением) будем понимать программу или иное формализованное описание, обеспечивающее автоматизацию решения на ЭВМ задачи пользователя как независимо, так и с помощью программно-инструментальных средств.

Исторически сложившаяся технология разработки программ решения задач экономического характера строилась исходя из "позадачного" подхода, при котором слабо учи-

тывались или вообще не учитывались программно-информационные взаимосвязи между отдельными задачами, а в качестве инструментальных средств программирования использовались исключительно алгоритмические языки. Принципиальная схема такого процесса представлена на рис. 7.1.

В зависимости от специфических особенностей конкретной задачи (ее вычислительной и логической сложности, состава и структуры исходной, промежуточной и результатной информации и т.п.), профессионального уровня подготовки специалистов и ряда других факторов некоторые этапы технологического процесса, представленные в общей схеме, могут быть объединены в более крупные этапы или реализовываться в неявном виде.

Первый этап технологического процесса представляет собой постановку задачи. На этом этапе раскрывается организационно-экономическая сущность задачи, т.е. формулируется цель ее решения; определяется взаимосвязь с другими задачами; указывается периодичность ее решения; устанавливаются состав и формы

представления входной, промежуточной и результатной информации; характеризуются формы и методы контроля достоверности информации на ключевых этапах решения задачи; специфицируются формы взаимодействия пользователя с ЭВМ в ходе решения задачи и т.п.

Особое внимание в процессе постановки задачи уделяется детальному описанию входной, выходной (результатной) и промежуточной информации. При этом характеризуются:

форма представления отдельных реквизитов (цифровая, символьная и т.д.);

количество знаков (разрядов), выделяемых для записи реквизитов исходя из их максимальной значности;

вид реквизита по его роли в процессе решения задачи (исходный, расчетный, нормативный, справочный и т.п.);

источник (документ, задача и т.п.) возникновения реквизита.

Кроме того, для цифровой информации указываются: целочисленный или дробный характер реквизита (для последних дополнительно указывается количество десятичных знаков, выделяемых для записи дробной части числа), допустимый диапазон изменения величины реквизита (т.е. его максимальное и минимальное допустимое значение).

Для расчетных реквизитов дается соответствующее описание формул расчета и особо выделяются те реквизиты, которые используются при последующих решениях задачи, так как они подлежат сохранению в памяти ЭВМ.

Особенностью экономических задач является использование в процессе их решения массивов условно-постоянной информации, содержащей многократно используемые справочные, нормативные, расценочные, планово-директивные и другие сведения. Данная информация также детально специфицируется в соответствии с общими требованиями к описанию информации, и, кроме того, указывается периодичность внесения изменений в эти массивы.

Если в процессе решения задачи предполагается интерактивный режим работы пользователя (а это характерно для большинства задач экономического управления), то важной частью постановки задачи является описание перечня и иерархической структуры пользовательских меню.

Завершается постановка задачи описанием контрольного примера, демонстрирующего порядок решения задачи традиционным способом. Основное требование к контрольному примеру - отражение всего многообразия возможных форм существования исходных данных. Контрольный пример сопровождается перечислением различного рода штатных и нештатных ситуаций, которые могут возникнуть при решении задачи, и описанием ответных действий пользователя в каждой конкретной ситуации.

Особенность реализации этого этапа технологического процесса заключается в том, что конечный пользователь разрабатываемой программы, хорошо знающий ее проблемную сторону, обычно хуже представляет специфику и возможности использования ЭВМ для ее решения. В свою очередь, предметная область пользователя (особенно ее отдельные нюансы, способные оказать влияние на решение задачи) зачастую бывает незнакома разработчику программы, хотя он знает возможности и ограничения на применение ЭВМ. Именно эти противоречия являются основной причиной возникновения ошибок при реализации данного этапа технологического процесса разработки программ, которые затем неизбежно отражаются и на последующих этапах.

Отсюда вся важность и ответственность этого этапа, необходимость осуществления корректной и полной постановки задачи, а также однозначность ее понимания как разработчиком программы, так и ее пользователем, в качестве которого обычно выступает постановщик задачи.

Второй этап в технологии разработки программ - экономико-математическое описание задачи и выбор метода ее решения.

Выделение этого этапа обуславливается рядом причин, одна из которых вытекает из свойства неоднозначности естественного языка, на котором осуществляется описание постановки задачи. В связи с этим на втором этапе технологического процесса разработки программ выполняется формализованное описание задачи, т.е. устанавливаются и формулируются логико-математические зависимости между исходными и результатными данными.

Экономико-математическое описание задачи обеспечивает ее однозначное понимание постановщиком (пользователем) и разработчиком программы. В процессе подготовки экономико-математического описания (модели) задачи могут использоваться различные разделы математики. При решении экономических задач наиболее часто используются следующие классы моделей для формализованного описания их постановок:

аналитические (вычислительные);

матричные (балансовые);

графические (частным видом которых являются сетевые).

Выбор класса модели, а иногда и конкретной формы ее представления внутри одного и того же класса позволяет не только облегчить и ускорить процесс решения задачи, но и повысить точность получаемых результатов.

Хотя математическая запись постановки задачи, как правило, отличается высокой точностью отображения ее сущности, лаконичностью записи, а главное однозначностью понимания, далеко не для всех задач она может быть выполнена. Кроме того, математическое описание задачи в большинстве случаев трудно перевести на язык ЭВМ. Для задач, допускающих возможность экономико-математического описания, необходимо выбрать численный метод решения, а для нечисловых задач - принципиальную схему решения в виде однозначно понимаемой последовательности выполнения элементарных математических и логических функций (операций).

При выборе метода решения задачи предпочтение отдается методу, который наиболее полно удовлетворяет следующим требованиям:

- обеспечивает необходимую точность получаемых результатов и не обладает свойством вырождения (т.е. бесконечного закливания на каком-либо участке решения задачи при определенном наборе исходных данных);

позволяет использовать готовые стандартные программы для решения задачи или ее отдельных фрагментов;

ориентирован на минимальный объем исходной информации;
обеспечивает наиболее быстрое получение искомым результатов.

Сложность и ответственность этапа экономико-математического описания задачи и выбора (разработки) соответствующего метода ее решения часто требуют привлечения квалифицированных специалистов в области прикладной математики, обладающих знанием таких дисциплин, как исследование операций, математическая статистика, численный анализ, вычислительная математика и т.п.

Третий этап технологического процесса подготовки решения задач на ЭВМ представляет собой алгоритмизацию ее решения, т.е. разработку оригинального или адаптацию (уточнение и корректировку) уже известного алгоритма.

Алгоритмизация - это сложный творческий процесс. В основу процесса алгоритмизации положено фундаментальное понятие математики и программирования - алгоритм. Название "алгоритм" (правильнее "алгори́фм") происходит от латинизированного воспроизведения арабского имени узбекского математика Аль-Хорезми, жившего в конце VIII - начале IX в., который первым сформулировал правила, позволяющие систематически составлять и решать квадратные уравнения.

Наряду с трактовкой алгоритма в соответствии с принятым стандартом (по ГОСТ 19.004-80 "алгоритм - это точное предписание, определяющее вычислительный процесс, ведущий от варьируемых начальных данных к искомому результату ") термин "алгоритм" может быть представлен более развернутым определением как конечный набор правил, однозначно раскрывающих содержание и последовательность выполнения операций для систематического решения определенного класса задач за конечное число.

Любой алгоритм обладает следующими свойствами: детерминированностью, массовостью, результативностью и дискретностью.

Детерминированность (определенность, однозначность) означает, что набор указаний алгоритма должен быть однозначно и

точно понят любым исполнителем. Это свойство определяет однозначность результата работы алгоритма при одних и тех же исходных данных.

Массовость алгоритма предполагает возможность варьирования исходных данных в определенных пределах. Это свойство определяет пригодность использования ал-

горитма для решения множества задач данного класса. Свойство массовости алгоритма является определяющим фактором, обеспечивающим экономическую эффективность решения задач на ЭВМ, так как для задач, решение которых осуществляется один раз, целесообразность использования ЭВМ, как правило, диктуется внеэкономическими категориями.

Результативность алгоритма означает, что для любых допустимых исходных данных он должен через конечное число шагов (или итераций) завершить работу.

Дискретность алгоритма - это возможность разбиения алгоритмического процесса на отдельные элементарные действия, возможность реализации которых человеком или ЭВМ не вызывает сомнения, а результат их выполнения вполне определен и понятен.

Таким образом, алгоритм дает возможность чисто механически решать любую задачу из некоторого класса однотипных задач.

Сложность и ответственность реализации этапа алгоритмизации объясняются тем, что для решения одной и той же задачи, как правило, существует несколько различных алгоритмов, отличающихся друг от друга уровнем сложности, объемами вычислительных и логических операций, составом необходимой исходной и промежуточной информации, точностью получаемых результатов и другими факторами, которые могут оказать существенное влияние на эффективность выбранного способа решения задачи.

Процесс алгоритмизации решения задачи обычно реализуется по следующей схеме:

выделение автономных этапов процесса решения задачи (как правило, с одним входом и выходом);

формализованное описание содержания работ, выполняемых на каждом выделенном этапе;

проверка правильности реализации выбранного алгоритма на различных примерах решения задачи.

Существует несколько способов описания алгоритмов: словесный, формульно-словесный, графический, средствами языка операторных схем, с помощью таблиц решений и др. Помимо требования обеспечения наглядности выбор конкретного способа диктуется рядом факторов, из которых определяющими являются: степень необходимой детализации представления алгоритма, уровень логической сложности задачи и т.п.

Словесный способ описания алгоритма отражает содержание выполняемых действий средствами естественного языка. К достоинствам этого способа следует отнести его общедоступность, а также возможность описывать алгоритм с любой степенью детализации. Недостатками способа являются достаточно громоздкое описание (и, как следствие, относительно низкая наглядность), отсутствие строгой формализации в силу неоднозначности восприятия естественного языка, вытекающего из свойств синонимии, омонимии, полисемии.

Формульно-словесный способ описания алгоритма основан на записи содержания выполняемых действий с использованием изобразительных возможностей языка математики, дополненного необходимыми пояснениями средствами естественного языка. Данный способ, обладая всеми достоинствами словесного способа, более лаконичен, а значит, и более нагляден, имеет большую формализованность, хотя и не является строго формальным.

Графический способ описания алгоритма представляет собой изображение логико-математической структуры алгоритма, при котором все этапы процесса обработки информации отображаются с помощью установленного набора геометрических фигур (блоков), имеющих строго определенную конфигурацию в соответствии с приписанным им характером выполняемых действий (например, вычислением, вводом-выводом информации, проверкой логических условий и т.п., рис. 7.2).

Изображение схем алгоритмов при этом осуществляется по определенным правилам, ГОСТам и ОСТам, которые повышают их наглядность и однозначность восприятия, что облегчает обнаружение логических ошибок в процессе отладки программ.

Операторный способ записи алгоритма - это изображение последовательности операций процесса обработки данных с помощью заданного набора буквенных символов, обозначающих ту или иную типовую операцию. Последовательность выполнения операций алгоритма определяется расположением операторов в схеме (при чтении слева направо в соответствии с цифровой индексацией). Передача управления от оператора к оператору осуществляется в порядке следования в символической записи алгоритма, в случае отсутствия передачи управления от очередного оператора к последующему оператору записи между ними ставится признак завершения ветви алгоритма - символ точка с запятой. Нарушение естественного порядка выполнения операторов отражается с помощью символов передачи управления (стрелок), которые используются: для указания

перехода от условного оператора при разветвлении алгоритма; в случае отражения безусловного перехода; от последнего оператора, завершающего одну из ветвей алгоритма.

Использование операторного способа представления алгоритма значительно упрощает процесс его записи, так как каждому оператору схемы обычно соответствует определенная совокупность достаточно простых операций обработки информации. Однако из-за малой наглядности и информативности отображения процесса решения задачи использование языка операторных схем не нашло широкого практического использования для разработки и отражения алгоритмов решения задач экономического характера.

Перечисленные способы описания алгоритмов имеют существенный недостаток, они не обеспечивают наглядности представления многовариантных вычислительных процессов, что характерно для алгоритмов решения сложных задач с разветвленной логикой. Особенно остро эти недостатки проявляются, когда возникает потребность внесения изменений и дополнений в логическую схему решения задачи.

Для разработки алгоритмов решения многовариантных расчетов с большим количеством проверок условий, определяющих выбор той или иной ветви процесса обработки информации, целесообразно использовать изобразительные средства в виде таблиц решений.

Таблицы решений, возникшие в обычной деловой практике, зарекомендовали себя как удобное средство, позволяющее четко, быстро и просто описывать достаточно сложные ситуации в задачах управления.

Описывая саму задачу и необходимые для ее решения действия, таблицы решений в наглядной форме определяют, какие условия должны быть выполнены, прежде чем можно будет переходить к тому или иному действию.

Легкость освоения специалистами любой области (профессии), простота модификации, компактность и, главное, более общая по сравнению с блок-схемами форма и более строгая логичность представления исходных условий и получаемой конечной информации, необходимой для программирования задач, - вот основные причины широкого использования таблиц решений.

Кроме того, таблицы решений пригодны для описания параллельных процессов (которые невозможно в удобной форме представить с помощью блок-схем), удобны для описания логики при построении сложных моделей ситуационного управления.

Сравнение изобразительных возможностей таблиц решений и блок-схем можно проиллюстрировать на задаче определения стажа работы на основе дат зачисления и увольнения работника' (табл. 7.1).

Составление (адаптация) программ (кодирование) является завершающим этапом технологического процесса разработки программных средств, предшествующим началу непосредственно машинной реализации алгоритма решения задачи. Процесс кодирования заключается в переводе описания алгоритма на один из доступных для ЭВМ языков программирования. В процессе составления программы для ЭВМ конкретизируются тип и структура используемых данных, а последовательность действий, реализующих алгоритм, отражается посредством конкретного языка программирования.

Тестирование и отладка составляют заключительный этап разработки программы решения задач. Оба эти процесса функционально связаны между собой, хотя их цели несколько отличаются друг от друга. Тестирование представляет собой совокупность действий, предназначенных для демонстрации правильности работы программы в заданных диапазонах изменения внешних условий и режимов эксплуатации программы. Цель тестирования заключается в демонстрации отсутствия (или выявлении) ошибок в разработанных программах на заранее подготовленном наборе контрольных примеров. Процессу тестирования сопутствует понятие "отладка", которое подразумевает совокупность действий, направленных на устранение ошибок в программах, начиная с момента обнаружения фактов ошибочной работы программы и завершая устранением причин их возникновения.

По своему характеру (причине возникновения) ошибки в программах делятся на синтаксические и логические.

Синтаксические ошибки в программе представляют собой некорректную запись отдельных языковых конструкций с точки зрения правил их представления для выбранного языка программирования. Эти ошибки выявляются автоматически при трансляции исходной программы (т.е. в процессе ее перевода с исходного языка программирования во внутренние коды машины) до ее выполнения. После устранения синтаксических ошибок проверяется логика работы программы на исходных данных. При этом возможны следующие основные формы проявления логических ошибок:

в какой-то момент программа не может продолжать работу (возникает программное прерывание, обычно сопровождающееся указанием места в программе, где оно произошло);

программа работает, но не выдает всех запланированных результатов и не выходит на останов (происходит ее "зацикливание");

программа выдает результаты и завершает свою работу, но они полностью или частично не совпадают с контрольными.

После выявления логических ошибок и устранения причин их возникновения в программу вносятся соответствующие исправления, и ее отладка продолжается.

Программа считается отлаженной, если она безошибочно выполняется на достаточно представительном наборе тестовых данных, обеспечивающих проверку всех ее участков (ветвей).

Процесс тестирования и отладки программ носит итерационный характер и считается одним из наиболее трудоемких этапов процесса разработки программ. По оценкам специалистов, он может составлять от 30 до 50% в общей структуре затрат времени на разработку проектов и зависит от объема и логической сложности разрабатываемых программных комплексов.

Для сокращения затрат на проведение тестирования и отладки в настоящее время широко применяются специальные программные средства тестирования (например, генераторы тестовых данных) и приемы отладки (например, метод трассировки программ, позволяющий выявлять, все ли ветви программы были задействованы при решении задачи с заданными наборами исходных данных).

В настоящее время вопросы тестирования программных средств приобретают все более важное значение, так как по мере перехода soft-индустрии в фазу зрелости требования к качеству программной продукции повышаются. Производители программной продукции выделяют значительные средства на тестирование своих продуктов, так как выпуск на рынок тиражируемого продукта[^] содержащего ошибки, способен навсегда подорвать доверие к «го производителю и даже послужить причиной его полного краха. Вместе с тем уровень надежности даже широко распространенных программных продуктов подчас остается недостаточно высоким.

Учитывая возрастающую роль требования обеспечения защиты информации, пользователю (заказчику) ПС необходимо обеспечить проверку приобретаемых (и даже соз-

даваемых в своих собственных структурах) программных средств на наличие различного рода «закладок». К сожалению, история знает немало случаев, когда разработчики включали такие «закладки» с целью удовлетворения своих собственных интересов в ущерб интересам заказчиков.

В этом плане примечателен факт недавнего обвинения компании Microsoft в «шпионаже» за теми, кто устанавливал их программные продукты на своих компьютерах. Так, было выявлено, что программы, входящие в Office 97, создавали специальные идентификационные номера, помечавшие документы в форматах Word и Excel. В результате компания Microsoft была вынуждена создать и распространить по сети Интернет программу-заплатку, препятствующую появлению таких идентификационных номеров, а также программу, стирающую идентификационные номера в Windows 98.

Учитывая вышесказанное, очевидно, что роль независимого качественного тестирования программных средств будет постоянно представлять объект особого внимания потенциальных пользователей информационных систем.

Важность процессов тестирования качества программных средств, с одной стороны, и сложность самой методологии и многообразие применяемых методов тестирования', с другой стороны, обусловили появление специализированных фирм, обладающих дорогостоящими инструментами тестирования и квалифицированным персоналом, предлагающих платные услуги подобного рода.

В настоящее время лидирующее положение на мировом рынке автоматизированных средств контроля качества ПО занимают три компании: Rational Software (=27%), Intersolv (=11%), Mercury

Interactive («11%), тогда как на долю компании Microsoft приходится только 5% мирового рынка соответствующей продукции.

Оценивая возрастание роли независимого тестирования программных средств информационных систем, в нашей стране также стали появляться специализированные центры тестирования программных продуктов. Если до недавнего времени такие работы осуществлялись только в Лаборатории оптимизации серверных приложений (в московском представительстве Intel) и только для платформ этой корпорации, то в 1999 г. компания «АйТи» открыла свой Центр тестирования, который на коммерческой основе оказывает услуги любым компаниям в проведении полномасштабного тестирования информационных систем (как готовящихся к внедрению, так и уже находящихся в экс-

плуатации). В качестве испытательных стендов при этом используются серверы и рабочие станции Hewlett-Packard, Sun, Compaq, работающие под управлением Unix и Windows NT. На их платформах установлены СУБД Oracle, Microsoft SQL Server, Informix и Sybase. При этом клиентские места могут быть реализованы и на компьютерах отечественной сборки.

В качестве основных инструментов тестирования работоспособности и производительности в Центре применяются программные продукты мирового лидера в этой сфере софтверного бизнеса - компании Rational Software Corp. Используемые передовые технологии обеспечивают автоматизированное тестирование приложений архитектуры клиент-сервер как в режиме стабильной, так и стрессовой нагрузки системы (эмулируя произвольное число ее пользователей).

Еще один фундаментальный аспект тестирования состоит в том, что оно должно вписываться в общую инфраструктуру автоматизированных средств контроля качества ПО, используемых на протяжении всего жизненного цикла программного продукта.

После завершения процесса тестирования и отладки программные средства вместе с сопроводительной документацией передаются пользователю для эксплуатации. Основное назначение сопроводительной документации - обеспечить пользователя необходимыми инструктивными материалами по работе с программными средствами. Состав сопроводительной документации обычно оговаривается заказчиком (пользователем) и разработчиком на этапе подготовки технического задания на программное средство. Как правило, это документы, регламентирующие работу пользователя в процессе эксплуатации программы, а также содержащие информацию о программе, необходимую в случае возникновения потребности внесения изменений и дополнений в нее. Сопроводительная документация призвана также облегчить процесс выявления причин возникновения ошибок в работе программы, которые могут быть обнаружены уже в ходе ее эксплуатации пользователем.

Для передачи пользователю разработанных прикладных программных средств создается специальная комиссия, включающая в свой состав представителей разработчиков и заказчиков (пользователей). Комиссия в соответствии с заранее составленным и утвержденным обеими сторонами планом проводит работы по приему-передаче программных средств и сопроводительной документации. По завершении работы комиссии оформляется акт приема-передачи.

В процессе внедрения и эксплуатации прикладных программных средств могут выявляться различные ошибки, не обнаруженные разработчиком при тестировании и отладке программных средств. Поэтому при реализации достаточно сложных и ответственных программных комплексов по согласованию пользователя (заказчика) с разработчиком этап эксплуатации программных средств может быть разбит на два подэтапа: экспериментальная (опытная) и промышленная эксплуатация. Смысл экспериментальной эксплуатации заключается во внедрении разработанных программных средств на объекте заказчика (нередко параллельно с уже существующими методами решения задач) с целью проверки ее работоспособности и удобства работы пользователей при решении реальных задач в течение достаточно длительного периода времени (обычно не менее года). Только после завершения периода экспериментальной эксплуатации и устранения выявленных при этом ошибок и учета замечаний программное средство передается в промышленную эксплуатацию.

Для повышения качества работ, оперативности исправления ошибок, выявляемых в процессе эксплуатации программных средств, а также выполнения различного рода модификаций, в которых может возникнуть необходимость в ходе эксплуатации, разработчик может по договоренности с пользователем осуществлять их сопровождение. Целесообразность привлечения высококвалифицированных специалистов для сопровождения программных средств у пользователя объясняется тем, что затраты на сопровождение программ значительно превосходят первоначальные затраты на их разработку (приобретение).

Следует принимать во внимание, что по своему характеру и последовательности выполняемых действий внесение различного рода изменений в уже функционирующие программные средства представляет в значительной мере повторение рассмотренных выше этапов, начиная с постановки задачи и кончая внесением изменений в сопроводительную документацию.

Описанная схема технологического процесса разработки прикладных программных средств отражает их "жизненный цикл", т.е. временной интервал с момента зарождения программы до момента полного отказа от ее эксплуатации.

ТЕХНОЛОГИЯ СИСТЕМНОГО ПРОЕКТИРОВАНИЯ ПРОГРАММНЫХ СРЕДСТВ

Пока автоматизация решения задач экономического управления носила локальный, частный характер, а количество таких задач было невелико, рассмотренная выше схема технологического процесса могла в большей или меньшей степени удовлетворять разработчиков. Когда возникла потребность создания систем автоматизированной обработки информации, внедрение которых могло обеспечить совершенствование организационно-экономического управления, указанная схема оказалась недостаточно эффективной, так как она не отражала основного принципа разработки - принципа системного подхода, что проявилось особенно ярко в виде массового дублирования данных в информационных массивах.

В качестве альтернативы такому дублированию информации возникла концепция баз данных как единого, централизованного хранилища всей информации, необходимой для решения задач управления. Первоначально в противовес огромному дублированию информации, присущему позадачному подходу, концепция БД подразумевала полное отсутствие такого дублирования. Однако теоретически корректная концепция в реальности оказалась малоэффективной, так как безусловный выигрыш в объемах необходимой памяти оборачивался значительным проигрышем во времени, требуемом на поиск и выборку из БД информации, необходимой для решения той или иной конкретной задачи.

В связи с этим в настоящее время концепция БД подразумевает разумный компромисс между сокращением до минимума необходимого дублирования информации и эффективностью процесса выборки и обновления требуемых данных. Действительное обеспечение такого решения возможно только при условии системного анализа всего комплекса задач, подлежащих автоматизации, уже на этапе описания системы: ее целей и функций, состава и специфики информационных потоков, информационного состава задач и даже отдельных программных модулей. Системный подход, базирующийся на положениях общей теории систем, наиболее эффективен при решении сложных задач анализа и синтеза, требующих одновременного использования ряда научных дисциплин. Общая теория систем выступает в этом плане как общенаучная междисциплинарная методология.

Другим важным фактором, обуславливающим необходимость системного подхода, начиная с этапа формулирования требования и постановки задач, является то, что на этот этап приходится до 70 - 80% всех затрат на разработку прикладного ПО и он имеет

особое значение в обеспечении соответствия результатов разработки потребностям конечных пользователей.

Объективное требование системного подхода к разработке программных средств решения задач при автоматизации систем организационно-экономического управления вызвало необходимость дифференциации специалистов-разработчиков, что проявилось в выделении в их составе: системных аналитиков, системотехников, прикладных и системных программистов.

Системный аналитик, исходя из общих целей, назначения, технических характеристик, состава и описания требований пользователей к прикладным задачам и системе в целом, формулирует общие формальные требования к ПО системы.

Специалист-системотехник преобразует общие формальные требования в детальные спецификации на отдельные программы, участвует в разработке логической структуры базы данных и т.п., т.е. определяет общую информационно-программную структуру проекта.

Прикладной программист преобразует спецификации в логическую структуру программных модулей, а затем и в программный код.

Системный программист обеспечивает сопряжение программных модулей с программной средой, в рамках которой предстоит функционировать прикладным программам (задачам).

В целях сокращения общей длительности разработки системы начало некоторых этапов технологического процесса осуществляется еще до полного завершения работ на предыдущем этапе. Такой частичный параллелизм в работе, кроме того, обуславливается и итерационным характером работ на этих этапах, когда в ходе выполнения отдельных работ /-этапа возникает необходимость уточнения или изменения спецификаций, выполненных на предшествующих этапах, либо пользователь по своей инициативе вносит коррективы в исходные требования, что, естественно, отражается на всей последующей технологической цепочке реализации проекта.

Другой чертой системной разработки проектов прикладных программ является их ориентация на использование интегрированных и распределенных баз данных. В связи с этим в качестве инструментальных средств разработки компонентов ПО наряду с языками программирования стали применяться языковые средства СУБД. В этих условиях

технологическая схема процесса разработки программ решения задач экономического управления претерпела существенное изменение.

Микропроцессорная революция резко изменила приоритеты и актуальность проблем, присущих традиционным технологиям разработки прикладных программ. Быстрорастущая вычислительная мощность, рост других вычислительных возможностей современных ПК в сочетании с возможностью объединения этих ресурсов с помощью вычислительных сетей - все это позволило нивелировать погрешности пользователей - непрофессиональных программистов в плане эффективности создаваемых ими программных средств решения прикладных задач.

Возможность исключения из технологической цепочки программистов-профессионалов (посредников) создала предпосылки для ускорения процесса разработки прикладных программных средств, а главное - для сокращения количества ошибок, присущих традиционным технологическим схемам, когда основные усилия профессиональных программистов затрачивались на то, чтобы адекватно воспринять требования, предъявляемые конечными пользователями к программам, обеспечить своевременное получение достоверных, исчерпывающих данных, необходимых для решения задачи, и т.п.

Но эффект от такого "вытеснения" профессиональных программистов из их сферы деятельности пользователями-непрофессионалами часто снижался или не ощущался вообще в связи с тем, что, не владея основами методологии разработки программных средств, типовыми программистскими приемами и умением использовать "подручные" средства из арсенала той или иной инструментальной среды, последние зачастую попадают в различн"тупиковые" ситуации, которые не составляют каких-либо трудностей для профессионалов в области программирования'.

СОВРЕМЕННЫЕ МЕТОДЫ И СРЕДСТВА РАЗРАБОТКИ ПРИКЛАДНЫХ ПРОГРАММНЫХ СРЕДСТВ

На протяжении всей истории программирования доминирующая роль отводилась проблеме определения методов и способов, облегчающих разработку и последующее сопровождение программ, сокращающих количество ошибок при создании и модификации программ, снижение трудоемкости программирования.

Опыт разработки больших и сложных программных комплексов показал, что рациональный подход к решению этой проблемы опирается на метод, получивший разное название (метод нисходящего проектирования, метод пошаговой детализации, метод иерархического проектирования), но заключающийся в определении спецификаций компонентов системы путем последовательного выделения в ее составе отдельных слагаемых и их постепенной детализации до уровня, обеспечивающего однозначное понимание того, что и как необходимо разрабатывать и реализовывать.

Этот метод является незаменимым при разработке сложных по характеру и больших по объему программ, когда к их разработке необходимо привлекать большое количество программистов, работающих параллельно. Он позволяет концентрировать внимание разработчиков на наиболее ответственных частях программы, а также облегчает возможность постоянного контроля за ее работоспособностью по мере разработки, отладки и объединения отдельных составляющих программ за счет организации непрерывности этого процесса в течение всей разработки.

Для ускорения разработки программного комплекса часто вместо некоторых программ нижнего уровня, находящихся в процессе разработки, могут применяться специальные "программы-заглушки". Программы-заглушки требуются только на ранних стадиях разработки для того, чтобы не сдерживать общий ход создания программного комплекса. Суть программы-заглушки заключается в том, что при обращении к ней в соответствии с заданным набором исходных тестовых данных она не формирует, а выбирает результат "решения" из заранее подготовленного набора. Благодаря этому обеспечивается возможность имитировать работу на ЭВМ реально создаваемой программы, а следовательно, осуществлять проверку работоспособности программ верхнего уровня еще до того, как будут разработаны и отлажены все составляющие ее программы нижнего уровня.

Реализация метода нисходящего проектирования тесно связана с другим понятием программирования - модульным проектированием, так как на практике при декомпозиции сложной программы возникает вопрос о разумном пределе ее дробления на составные части. Вместе с тем понятие модульности нельзя сводить только к представлению сложных программных комплексов в виде набора отдельных функциональных блоков. Модуль - это последовательность логически взаимосвязанных фрагментов,

оформленных как отдельная часть программы. При этом программные модули должны обладать следующими свойствами:

- на модуль можно ссылаться (т.е. обращаться к нему) по имени, в том числе и из других модулей;

- по завершении работы модуль должен возвращать управление тому модулю, который его вызывал;

- модуль должен иметь один вход и выход;

- модуль должен иметь небольшой размер, обеспечивающий его обзорность.

При разработке сложных программ, как правило, в них выделяют головной управляющий модуль и подчиненные ему модули, обеспечивающие реализацию отдельных функций управления, функциональную обработку (т.е. непосредственную реализацию основного назначения программного комплекса), и вспомогательные модули, обеспечивающие сервисное обслуживание пакета (например, сбор и анализ статистики работы программы, обработку различного рода ошибочных ситуаций, обучение и выдачу подсказок и т.п.).

Модульный принцип разработки программ обладает следующими преимуществами:

- большую программу могут разрабатывать одновременно несколько исполнителей, а это позволяет сократить сроки ее разработки;

- появляется возможность создавать (и многократно использовать в дальнейшем) библиотеки наиболее употребляемых программ;

- упрощается процедура загрузки больших программ в оперативную память, когда требуется ее сегментация;

- возникает много естественных контрольных точек для наблюдения за осуществлением хода разработки программ, а в последующем для контроля за ходом исполнения программ;

- обеспечиваются более эффективное тестирование программ, проще осуществляются проектирование и последующая отладка.

Преимущества модульного принципа построения программ особенно наглядно проявляются на этапе сопровождения и модификации программных продуктов, позволяя значительно сократить затраты сил и средств на реализацию этого этапа.

Актуальная для начального периода развития и использования ЭВМ проблема разработки программ, занимающих минимум основной памяти и выполняющихся за кратчайшее время, в последующем в связи с резким падением стоимости аппаратной части ЭВМ, значительным возрастанием их быстродействия и объемов памяти сменилась необходимостью разработки и применения принципиально новых "индустриальных" методов составления программ. Все это нашло свое воплощение в разработке принципа структурного программирования. Одной из целей структурного программирования было стремление облегчить процесс разработки и отладки программных модулей, а главное - процесс их последующего сопровождения и модификации.

В настоящее время структурное программирование - это целая дисциплина, объединяющая несколько взаимосвязанных способов создания ясных, легких для понимания программ. Эффективность применения современных универсальных языков программирования во многом определяется удобством написания с их помощью структурных программ.

Другое направление совершенствования процесса разработки прикладных программ - развитие программно-инструментальных средств программирования экономических задач. Основу таких средств программирования задач организационно-экономического управления составляют системы автоматизации программирования, или системы программирования, которые обеспечивают возможность решения широкого круга задач непосредственно в среде операционной системы ЭВМ.

Вместе с тем следует учитывать специфику задач экономического управления:

доминирование задач с относительно несложными вычислительными алгоритмами и потребностью формирования различного рода накопительных итогов, т.е. задач "прямого счета";

работу с большими массивами (обычно упорядоченной определенным образом) исходной информации;

требование предоставления большинства результатной информации в виде документов табличной формы.

Решение указанных задач может быть осуществлено с использованием программно-инструментальных средств СУБД и электронных таблиц.

Основное достоинство этих инструментальных средств заключается в том, что они предъявляют меньшие требования к их пользователям в области программирования

как науки, обеспечивая в то же время достаточно быстрое эффективное решение большинства задач экономического управления. В связи с этим они пользуются большой популярностью среди непрофессиональных программистов. К наиболее развитым программно-инструментальным средствам относятся системы автоматизации проектирования (САПР) ПО, создание которых было начато в конце 70-х годов.

Однако подобные разработки слабо учитывали требования системного подхода, так как ограничивались автоматизацией лишь части этапов разработки ПО, причем, как правило, узкого класса задач. Вместе с тем появление и быстрое распространение современных ПК среди профессиональных разработчиков ПО, благодаря их широким возможностям интерактивного взаимодействия, открыли новые перспективы в деле автоматизации. Так, за последнее десятилетие в области средств автоматизации программирования сформировалось новое направление под общим названием кейс-технология (Computer Aided Software Engineering-CASE).

Базируясь на классических методах разработки ПО, CASE-технология представляет собой совокупность средств системного анализа, проектирования, разработки и сопровождения сложных программных систем, поддерживаемых комплексом взаимоувязанных инструментальных средств автоматизации всех этапов разработки программ. Благодаря структурным методам на стадиях анализа и проектирования CASE-технология обеспечивает разработчиков широкими возможностями для различного рода моделирования, а централизованное хранение всей необходимой для проектирования информации и контроль за целостностью данных гарантируют согласованность взаимодействия всех специалистов, занятых в разработке ПО.

Высокая "тяжесть" последствий ошибок при разработке ПО, присущих этапу составления спецификаций для автоматизации информационной системы объекта, вызвала поиск путей сокращения их числа на этом этапе до минимума. Естественным решением проблемы была разработка формализованного аппарата для составления описания и последующего анализа информационной модели системы. Впервые такой подход с системных позиций был реализован сотрудниками Мичиганского университета под руководством проф. Д.Тайкроу в рамках проекта ISDOS (Information System Design and Optimization System - проектирование и оптимизация информационных систем).

В основу реализации проекта ISDOS был положен специально разработанный язык PSL (Problem Statment Language - язык постановки задач), который обеспечивал

описание целей, функций и задач систем организационно-экономического управления, и программный анализатор описаний PSA (Problem Statment Analizator- анализатор постановок задач), выполненных средствами PSL.

На языке PSL пользователь специфицирует параметры, определяющие входы и выходы информационной системы и их временные характеристики.

Проект ISDOS был первой западной системой автоматизированного формализованного анализа требований к программному обеспечению. Он состоял из взаимосвязанных модулей, которые обеспечивали:

ввод, контроль и кодирование спецификаций проектируемой системы;

анализ правильности постановки задач и их согласованности;

выявление ошибок и выдачу сообщений пользователям, а так же устранение дублирования в исходной информации;

преобразование постановок задач после проверки и корректировки исходных данных в машинные программы в соответствии с заданными требованиями к системе;

выделение основных элементов информационной системы.

Первая версия ISDOS, разработанная применительно к системам административного управления, впоследствии применялась в области управления правительственными организациями, космическими объектами, торговыми организациями и т.д.

Язык PSL также позволяет системному аналитику описать в формализованном виде требуемые результаты решения задач, необходимые входные данные, взаимосвязи между отдельными процедурами и данными, предоставить информацию о характеристиках отдельных модулей, процедур и данных и т.д.

Подсистема PSA анализирует поставленную и описанную с помощью PSL проблему и генерирует полезные для разработчика интегральные характеристики, такие, как формальные постановки задач, иерархические структуры данных, рекомендации по выбору ключевых слов, обобщенные блок-схемы алгоритмов обработки данных при решении задач и ряд других характеристик.

Поиск методов совершенствования процессов разработки прикладных программных средств обусловил появление методологии, по которой разработка программы начиналась не после завершения процесса выработки окончательных требований к ней, а как только устанавливались требования на первый, «стартовый» (пилотный) вариант прикладной программы, позволяющий начать содержательную работу по ее реализации

на компьютере. Это дало пользователю возможность, получая уже с первых шагов конкретное представление о характере реализации задачи, уточнять ее постановку. Тем самым облегчался процесс экспериментального поиска нужного решения автоматизации задачи.

Благодаря тесному взаимодействию разработчика с заказчиком (пользователем) на самом ответственном этапе создания прикладных программ между ними достигалось быстрое взаимопонимание цели поставленной задачи и возможности ее автоматизации в конкретных условиях. Это повышало скорость разработки программ и послужило основанием для названия такой технологии - RAD (Rapid Application Development), которая получила широкое распространение.

RAD-технология ознаменовала появление принципиально нового, отличного от традиционного, последовательного метода разработки программных средств (при котором очередной этап процесса создания ПС начинается только после завершения предыдущего этапа и не допускает какого-либо возврата к предшествующим этапам). Это дало повод назвать такую итерационную методологию «спиральной» в отличие от последовательной, называемой также «каскадной» или «водопадной».

По сути, спиральный метод представлял собой дополнение каскадного элементами итеративности (обычно до трех раз) и вместе с ним реализовал варианты структурного (или алгоритмического) подхода к разработке ПС.

RAD-технология позволила в какой-то степени сместить центр тяжести разработки ПС на начальные этапы ЖЦ, а кроме того, несмотря на наличие повторений, дала возможность в целом сократить время и затраты на разработку программных приложений. Благодаря этому базирующиеся на этом принципе методы получили название «fast-track» (быстрый путь).

Однако при всех достоинствах RAD-технологии, поддерживающей традиционные методы разработки программных средств, сфера ее эффективного применения ограничена, так как она не может обеспечить создание сложных программных продуктов, а ориентирована на разработку несложного заказного программного обеспечения. Когда в центре внимания разработчиков программного обеспечения оказались сложные информационные системы, не поддающиеся программированию «в лоб», традиционные методы, олицетворяющие структурный принцип разработки систем, оказались малоэффективными.

Стремление избавиться от недостатков структурного подхода привело к развитию новых идей, основанных на объектной декомпозиции систем. Такой подход к разработке программных средств получил название объектно-ориентированного (ООП).

В основе этого подхода лежат понятия объект, класс, инкапсуляция, наследование и полиморфизм.

Каждый объект является представителем некоторого класса однотипных объектов. Класс определяет общие свойства для всех его объектов. К таким свойствам относятся:

состав и структура данных, описывающих атрибуты класса и соответствующих объектов;

совокупность методов - процедур, определяющих взаимодействие объектов этого класса с внешней средой и другими объектами. Объекты и классы обладают характерными свойствами, которые активно используются при объектно-ориентированном подходе и во многом определяют его преимущества:

инкапсуляция - способность изменять реализацию любого класса объектов без опасения, что это вызовет нежелательные побочные последствия в программной среде;

наследование - возможность создавать из имеющихся классов новые классы по принципу «от общего к частному»;

полиморфизм - способность объектов выбирать метод обработки на основе типов данных, принимаемых в сообщении.

Объектно-ориентированная декомпозиция заключается в представлении системы в виде совокупности классов и объектов предметной области. При этом иерархический характер сложной системы отражается в виде иерархии классов, а ее функционирование рассматривается как взаимодействие объектов.

Объектно-ориентированный подход к программированию отличает сочетание четырех важных аспектов:

модульность, позволяющую локализовать область действия процедур по отношению к внутренним переменным, что обеспечивает возможность изменения локальных процедур и внутренних структур без необходимости изменения других программных модулей;

абстракция данных, при которой тип данных определяется на основе некоторого внутреннего представления и множества процедур для доступа и обработки данных абстрактного типа;

динамическая связка программ, позволяющая отказаться от перекомпилирования всей программы при внесении изменений некоторых типов в отдельные модули;

использование механизма «наследования», который позволяет создавать классы объектов и выделять в них подклассы объектов, обладающих теми или иными конкретными свойствами.

Такое качество позволяет создавать компактные программы при спецификации нового объекта, незначительно отличающегося от уже известного, в которых нет необходимости заново описывать общие части, а достаточно только задать различия.

Жизненный цикл объектно-ориентированной разработки программных систем содержит несколько этапов, но в отличие от структурного подхода в нем нет строгой последовательности их выполнения. Весь процесс носит принципиально итеративный характер. Главная особенность жизненного цикла при объектно - ориентированном подходе заключается в том, что он не предполагает жесткой последовательности выполнения отдельных этапов. При разработке может выясниться необходимость проведения дополнительного обследования; программирование и последующее тестирование могут потребовать возврата к проектированию.

Объектно-ориентированный подход обеспечивает быстрое создание прототипов проектируемой системы, постепенное развитие которых приводит к конечному результату. Однако объектно-ориентированный подход также имеет ряд недостатков. Эти недостатки прежде всего лежат в области программирования и заключаются в том, что в условиях динамического связывания объектов, многочисленности методов в классах и их вызовов, а также интенсивного межсегментного обмена, снижение производительности программ может носить критический характер. Частично решение этой проблемы может быть осуществлено за счет использования объектно-ориентированных языков программирования, включающих в себя средства, обеспечивающие более высокую производительность программ по сравнению с традиционными языками, а также за счет использования специальных программистских приемов.

Вместе с тем в условиях возрастающей потребности в прикладных информационных системах и постоянном дефиците квалифицированных программистских кадров указанные недостатки объектно-ориентированного подхода являются сдерживающим фактором его распространения.

В последнее время специалисты в области информационных технологий возлагают большие надежды на быстрое получение качественных программных средств разного

уровня сложности на основе открытых компонентных технологий. Эти технологии предполагают, что приложение должно собираться из готовых, хорошо отлаженных программных компонентов.

Компоненты программного обеспечения - это простейшие элементы, которые можно повторно (неоднократно) использовать при построении программных систем.

В отличие от других повторно используемых программных модулей компоненты можно модифицировать в процессе проектирования на уровне двоичных исполняемых кодов, в то время как библиотеки, подпрограммы и т.п. необходимо изменять на уровне исходных кодов.

В основе реализации компонентной идеологии лежит тот факт, что фундаментальных операций над информацией сравнительно немного - сбор, передача, извлечение, ввод и вывод, вычисления, преобразования, представление (структуризация), размещение, хранение, настройка, администрирование, защита, контроль и диагностика.

Такой ограниченный набор операций облегчает возможность унификации программных компонентов (проектных решений и их элементов) и прежде всего тех, которые являются базовыми, т.е. не затрагивают специфичной бизнес-логики приложений. Естественно, что разработка таких компонентов требует участия программистов высокой квалификации, так как предъявляет повышенные требования к уровню абстракции при их создании. Наиболее опытные специалисты будут конструировать компоненты, а основная масса менее квалифицированных программистов займется простыми задачами сборки приложений на их основе.

При этом выработка стандартов на компоненты гарантирует следующие свойства последних:

компоненты со схожими спецификациями становятся взаимозаменяемыми и допускают независимую модернизацию;

разработчики могут адаптировать как внешний вид представления, так и поведение (действия) компонентов в заранее оговоренных пределах;

компоненты можно объединять друг с другом, формируя более крупные компоненты и законченные приложения.

Таким образом, стандарты компонентов играют ключевую роль в процессе создания приложений, позволяя реализовывать все преимущества, заложенные в техноло-

гии повторно используемых компонентов, - повышенную производительность, единообразие, простоту применения и ускоренный цикл разработки.

Кроме того, стандартами компонентов определяется специальная информация (метаданные):

информация о самом компоненте, которая описывает его общие характеристики, относящиеся к этапам компиляции и выполнения, в том числе указания о том, где можно найти компонент и как его активизировать;

внешние ссылки, указывающие на метаданные, описывающие другие компоненты;

описатели типа, интерфейсы и т.д.

Эти метаданные помогают объединять компоненты и устанавливать взаимоотношения между ними, выражающиеся в том, что с помощью метаданных компоненты могут обнаруживать интерфейсы других компонентов и взаимодействовать с ними на этапе выполнения.

По оценкам специалистов, компонентные технологии создают реальные предпосылки для перенесения центра тяжести с этапа программирования на этап проектирования. Это достигается благодаря тому, что компонентные технологии позволяют в перспективе вырабатывать алгоритмы решения задач непосредственно пользователями-непрограммистами, умеющими сформулировать свои информационные потребности в виде того, что им нужно иметь на выходе системы, без спецификации того, как это необходимо сделать, т.е. применять непроцедурные языки программирования.

Другое направление разработки прикладных программных средств, олицетворяющее собой современный подход к реализации широкого круга задач для принятия управленческих решений, базируется на концепции создания специального хранилища данных (Data Warehouse). Концепция хранилищ данных была выдвинута в начале 90-х годов с целью обеспечения эффективного анализа и поддержки управления. Хранилище данных может быть представлено как общекорпоративная база данных, концептуальное отличие которой от традиционного представления баз данных заключается в следующем:

структура хранилища ориентирована на предметную область, а не на существующие прикладные программы;

актуализация данных в Data Warehouse означает не обновление элементов информации, а добавление новых элементов к уже имеющимся (что расширяет возможности проведения различного рода сравнительного анализа);

данные в хранилище накапливаются с течением времени и четко хронологизированы;

данные аккумулируются и интегрируются из различных источников - операционных систем;

данные хранятся в «атомарном», т.е. наиболее элементарном виде, что позволяет извлекать информацию для анализа в любом необходимом виде;

наряду с информацией, непосредственно отражающей состояние системы управления, в Data Warehouse аккумулируются и метаданные.

Метаданные (данные о данных) облегчают возможность визуального представления содержимого Data Warehouse, позволяют, «перемещаясь» по хранилищу, быстро отбирать необходимые данные для последующей обработки. Основные типы метаданных Data Warehouse отражают:

структуру и содержимое хранилища;

соответствие между исходными и выходными данными;

объемные характеристики данных;

критерии архивирования;

отношения между данными;

информацию по кодированию;

интервал жизни данных;

и т.п.

Концепция Data Warehouse поддерживается RAD-средствами разработки прикладного ПО, благодаря которым даже неспециалист может быстро создавать программные приложения, подбирая необходимые прототипы программ, расширяя их набор путем объединения и настройки более мелких.

Создание программных приложений для Data Warehouse по RAD- технологии представляет итеративный процесс, состоящий из следующих 5 этапов:

отбора необходимых объектов для создания программных приложений;

установки переменных для выполнения просмотра и анализа данных;

различного рода настройки атрибутов в соответствии с требованиями отображения информации и алгоритмов обработки;

тестирования приложения, возвращаясь при необходимости к предыдущим этапам; создания пользовательского интерфейса и пиктограмм.

Концепция Data Warehouse обеспечивает возможность разработки программных приложений для поддержки процессов принятия решений с использованием OLAP-систем. Система OLAP (On-Line Analytical Processing) предоставляет возможность разработки прикладного ПО информационных систем, ориентированных на организацию многомерных баз данных и создание корпоративных сетей, а также обеспечивает поддержку Web-технологий в сетях Internet/Intranet.

Успешное применение инструментальных средств OLAP-систем объясняется быстротой разработки приложений, гибкостью и широкими возможностями в области доступа к данным и их преобразования. В настоящее время на рынке ПО предлагается большое число OLAP-систем, разработчиками которых являются различные фирмы, например Arbor Software, IBM, Informix, Microsoft, Oracle, SAS Institute, Sybase.

СИСТЕМЫ И ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Составной частью общего (системного) программного обеспечения являются системы программирования с соответствующими алгоритмическими языками.

Системы программирования предназначены для совершенствования процесса разработки и отладки программ. Система программирования включает в свой состав: входной язык системы программирования (называемый также исходным языком); транслятор, обеспечивающий перевод (трансляцию) программы с входного языка системы на внутренний (машинный) язык; библиотеку стандартных, наиболее часто используемых подпрограмм (например, сортировки информации, различного рода встроенных функций и т.п.), подключаемых в процессе подготовки программ к выполнению, а также соответствующую документацию.

Языки программирования, или алгоритмические языки, классифицируются: по степени их зависимости от вычислительной машины; по ориентации на сферу применения; по специфике организационной структуры языковых конструкций и т.п. (рис. 7.4).

С учетом зависимости от ЭВМ языки программирования подразделяются на: машинно-зависимые и машинно-независимые.

Структура и средства машинно-зависимых языков отражают (учитывают) специфику функционирования определенного класса ЭВМ. При программировании задач с помощью таких языков требуется знание не только сущности реализуемого алгоритма решения задачи, но и технических особенностей конкретной ЭВМ и специфики способов написания для нее программ.

К машинно-зависимым языкам в первую очередь относятся машинные языки. Машинный язык является внутренним языком ЭВМ и представляет собой систему инструкций и данных, которые не требуют трансляции и могут непосредственно интерпретироваться и исполняться аппаратными средствами ЭВМ. Программирование на этих языках осуществлялось на ЭВМ первого и частично второго поколений.

К машинно-зависимым языкам программирования также относятся машинно-ориентированные языки, основные конструктивные средства которых также позволяют учитывать особенности архитектуры и принципов работы определенной ЭВМ или ряда ЭВМ, т.е. обладают теми же возможностями и требованиями к программистам, что и машинные языки, но в отличие от последних требуют предварительной трансляции на машинный язык программ, составленных с их помощью.

К данному виду языков программирования относятся: автокоды, языки символического кодирования и ассемблеры. В отличие от программирования на машинных языках программирование на машинно-ориентированных языках (ассемблерах) характерно и для современных ПК. Это объясняется тем, что в языке ассемблера допускается использование средств, присущих языкам высокого уровня.

Использование языка ассемблера, как правило, ограничивается областью системного программирования, т.е. программированием микропроцессоров, разработкой операционных систем или их компонентов, разработкой драйверов - программ обмена информацией между центральными и периферийными устройствами, программ увязки взаимодействия отдельных компонентов прикладных программ и т.д.

Тот факт, что языки данного класса учитывают специфику организации и принципов работы конкретных ЭВМ и допускают при программировании указание конкретных режимов работы физических средств ЭВМ, распределение памяти, явное определение внешних устройств и т.п., относит их к языкам "низкого уровня", или языкам уровня 1:1 (т.е. к языкам, для которых одному оператору входного языка программирования соответствует один оператор машинного языка).

Машинно-независимые языки (или языки высокого уровня) не требуют от пользователя полного знания специфики ЭВМ, на которой реализуется программа решения задачи. Инструментальные средства этих языков программирования позволяют записывать программу в виде, допускающем ее реализацию на ЭВМ с различными типами машинных операций, привязка к которым целиком возлагается на соответствующий транслятор.

Решение задачи на этих языках описывается в наглядном, достаточно легко воспринимаемом виде. Для них характерны: возможность написания выражений, символическая идентификация переменных, вызов функций по именам и т.п. Благодаря этому производительность программиста при составлении исходных программ на языках высокого уровня примерно в 10 -15 раз выше, чем на языке ассемблера. Однако получаемые в результате трансляции машинные программы, как правило, в 2 - 5 раз объемнее по сравнению с такой же программой, но написанной на ассемблере, и работают в 2 - 5 раз медленнее.

Быстрый рост производительности ЭВМ, с одной стороны, и хроническая нехватка программистских кадров, с другой стороны, послужили причиной бурного развития и применения высокоуровневых языков программирования.

Обособленное, промежуточное положение между машинно-независимыми и машинно-зависимыми языками занимает язык Си, создание которого явилось результатом попытки объединения достоинств, присущих языкам обоих классов:

в плане максимального использования возможностей конкретной вычислительной архитектуры (что присуще языкам низкого уровня), благодаря чему программы на языке Си компактны и работают эффективно;

в плане максимального использования мощных выразительных возможностей современных языков высокого уровня.

Результат такого компромисса обусловил достаточно сложный синтаксис языка Си.

Язык Си и его модификации в настоящее время используются главным образом для создания системных и прикладных программных продуктов, в которых решающее значение отводится факторам быстродействия и минимизации объемов памяти. На языке Си написано ядро операционной системы Unix, вследствие чего ее легко можно было изменять и модернизировать (а это упрощает процесс ее переноса с одной вычислитель-

ной системы на другую, при этом 95% исходного программного текста операционной системы остается неизменным).

Машинно-независимые языки классифицируются на процедурно-ориентированные и проблемно-ориентированные.

Процедурно-ориентированные (универсальные) языки эффективны для описания алгоритмов решения широкого класса задач. Из языков этого класса наиболее известны: Фортран, Кобол, ПЛ/1, Бейсик, Паскаль, Ада.

Проблемно-ориентированные языки предназначены для описания процессов обработки информации в более узкой, специфической области. Наиболее известными языками этой группы являются: РПГ, Лисп, АПЛ, GPSS.

В последнее время отмечается бурный рост объектно-ориентированных языков программирования, т.е. языков, ориентированных на разработку программных приложений для широкого круга разнообразных по сфере приложения задач, имеющих общность в реализуемых компонентах (например, при взаимодействии с базами данных, работе в условиях функционирования корпоративных сетей организаций или взаимодействии с глобальной сетью Интернет). Объектно-ориентированный подход в программировании позволяет применять одни и те же (типовые) архитектурные и концептуальные решения для быстрого создания эффективных программных приложений.

Основное достоинство алгоритмических языков высокого уровня - возможность описания программ решения задач в форме, максимально удобной для восприятия человеком. Но так как каждое семейство ЭВМ имеет свой собственный, специфический внутренний (машинный) язык и может выполнять лишь те команды, которые записаны на этом языке, то для перевода исходных программ на машинный язык используются специальные программы-трансляторы.

Работа всех трансляторов строится по одному из двух принципов: интерпретация или компиляция.

Интерпретация подразумевает пооператорную трансляцию и последующее выполнение оттранслированного оператора исходной программы. В связи с этим можно отметить два недостатка метода интерпретации: во-первых, интерпретирующая программа должна находиться в памяти ЭВМ в течение всего процесса выполнения исходной программы, т.е. занимать определенный объем памяти; во-вторых, процесс трансляции од-

ного и того же оператора повторяется столько раз, сколько раз должна исполняться эта команда в программе, что резко снижает производительность работы программы.

Несмотря на указанные недостатки, трансляторы-интерпретаторы получили достаточное распространение, так как они поддерживают диалоговый режим, что особенно удобно при разработке и отладке исходных программ. Кроме того, интерпретаторы легче разрабатывать, и они обходятся дешевле, чем компиляторы с того же языка.

В случае многократного решения задачи, когда быстрдействие работы вычислительной системы имеет существенное значение, целесообразно использовать другой принцип - компиляцию.

При компиляции процессы трансляции и выполнения разделены во времени: сначала исходная программа полностью переводится на машинный язык (после чего наличие транслятора в оперативной памяти становится ненужным), а затем оттранслированная программа может многократно исполняться. Следовательно, для одной и той же программы трансляция методом компиляции обеспечивает более высокую производительность вычислительной системы при сокращении требуемой оперативной памяти.

Большая сложность в разработке компилятора по сравнению с интерпретатором с того же самого языка объясняется тем, что компиляция программы включает два действия: анализ, т.е. определение правильности записи исходной программы в соответствии с правилами построения языковых конструкций входного языка, и синтез - генерирование эквивалентной программы в машинных кодах. Трансляция методом компиляции требует неоднократного "просмотра" транслируемой программы, т.е. трансляторы-компиляторы являются многопроходными: при первом проходе они проверяют корректность синтаксиса языковых конструкций отдельных операторов независимо друг от друга, при последующем проходе - корректность синтаксических взаимосвязей между операторами и т.д.

Полученная в результате трансляции методом компиляции программа называется объектным модулем, который представляет собой эквивалентную программу в машинных кодах, но не "привязанную" к конкретным адресам оперативной памяти. Поэтому перед исполнением объектный модуль должен быть обработан специальной программой операционной системы (редактором связей) и преобразован в загрузочный модуль, т.е. программный модуль с относительными адресами.

Загрузочный модуль может иметь простую, оверлейную или динамическую структуру.

Модуль простой структуры состоит из единственного загрузочного модуля, сформированного редактором связей. Этот модуль перед исполнением целиком загружается в оперативную память и включает все необходимые для его работы команды. Модули простой структуры наиболее эффективны с точки зрения производительности, так как в ходе исполнения требуют минимального вмешательства управляющей программы ОС.

Если программа функционально достаточно сложна или велика по размерам, то она реализуется в виде модулей (сегментов) оверлейной структуры (структуры "с перекрытием"). Загрузочный модуль оверлейной структуры состоит из оверлейных сегментов и содержит информацию, используемую оверлейным супервизором для загрузки отдельных сегментов в основную память. При этом разные сегменты такой программы могут повторно использовать одну и ту же область оперативной памяти. Оверлейная организация модулей менее эффективна с точки зрения времени их исполнения, так как требует большего вмешательства управляющей программы ОС.

При выполнении модулей простой и оверлейной структуры управляющая программа ОС не осуществляет вызов других загрузочных модулей, и в этом они уступают динамической организации модулей. При выполнении загрузочных модулей динамической структуры могут появляться запросы на другие загрузочные модули, и управляющая программа ОС может начать загрузку этих модулей в оперативную память еще до завершения исполнения предыдущего модуля. Благодаря этому вызываемые программные модули могут исполняться как последовательно, так и параллельно, что повышает быстродействие программы.

Наряду с рассмотренными выше трансляторами-интерпретаторами и трансляторами-компиляторами на практике используются также трансляторы интерпретаторы-компиляторы, которые объединяют в себе достоинства обоих принципов трансляции: на этапе разработки и отладки программ транслятор работает в режиме интерпретатора, а после завершения процесса отладки исходная программа повторно транслируется в объектный модуль (т.е. уже методом компиляции). Это позволяет значительно упростить и ускорить процесс составления и отладки программ, а за счет последующего получения объектного модуля обеспечить более эффективное исполнение программы.

Учитывая, что эффективность программ, получаемых с помощью высокоуровневых языков программирования, может значительно уступать аналогичным программам, составленным на языках низкого уровня, в ряде случаев используются оптимизирующие трансляторы (оптимайзеры), которые после завершения компиляции осуществляют оптимизацию объектного модуля.

Например, хороший компилятор способен распознать небрежность программиста и исключить из цикла «статические» вычисления. Под «статическими» вычислениями понимается выполнение в цикле определенной операции, результат которой не зависит от итераций цикла. Распознав такую конструкцию, компилятор выводит ее за рамки цикла. Этот тип оптимизации называется перемещением выражений.

Другим более сложным видом оптимизации является устранение неиспользуемого кода. Компилятор находит операторы, которые не выполняются ни при каких условиях, и не включает их в исполняемый код.

В общем случае в зависимости от выигрыша в производительности и временных затрат все виды оптимизации можно разделить на несколько уровней. Первый и второй уровни оптимизации, как правило, повышают быстродействие на 10 - 15% при минимальных затратах. Третий уровень оптимизации позволяет увеличить производительность еще на 5%, однако это обойдется значительно дороже.

Таким образом, выбор типа алгоритмического языка (см. рис. 7.3) зависит от многих факторов: назначения, удобства написания исходных программ, эффективности получаемых объектных программ и т.п. Для ПК ведущее место в настоящее время занимают языки высокого уровня, например различные версии языков: Бейсик, Паскаль, Фортран, Кобол, АПЛ, Ада, Си и т.д.

Среди них доминирующая роль отводится процедурно-ориентированным языкам, называемым также универсальными (хотя некоторые из них, например Фортран, Кобол, Бейсик и т.п., и разрабатывались с ориентацией на конкретную сферу применения).

Абсолютное большинство языков программирования в настоящее время составляют процедурные языки, с помощью которых программист определяет последовательность реализации событий в объектной программе путем последовательной записи предложений в исходной программе. Иными словами, программирование на этих языках подразумевает необходимость описания не только того, что необходимо получить в результате решения задачи, но и того, как это необходимо осуществить.

Одним из главных направлений совершенствования языков программирования для задач экономического управления было стремление к разработке таких языков, в которых до минимума сводилась (а в идеале вообще отсутствовала) проблема как. Естественно, что такие языки не могут быть процедурными.

Первую попытку создания такого языка предпринял Ломбарда, предложивший и реализовавший в 1963 г. для частного класса задач, связанных с обработкой файлов в сфере управления, язык General business-oriented language based on decision expression. В этой же области свои языки предложили фирмы ICL (язык NICOL) и IBM (язык РПГ), в которых многое связанное с процедурой работы программы носит скорее неявный, чем явный характер. Учитывая, что значительная часть логики программы реализуется автоматически, время создания программ с помощью таких языков и число отладочных проверок значительно сократились.

Таким образом, наряду с универсальными процедурно-ориентированными языками стали создаваться проблемно-ориентированные языки программирования, предназначенные для описания процессов обработки информации в какой-либо узкой (специфической) области, в которых решение задачи в большей степени сосредоточивалось на проблеме, что необходимо получить в результате, а проблема, как это необходимо сделать, в большей или меньшей степени снималась с программиста. Среди этих языков наиболее известными являются: РПГ - язык для генерации отчетов, Лисп - язык для обработки списков, GPSS - язык для моделирования, АПЛ - язык для статистической обработки массивов.

Актуальности решения проблемы разработки языков программирования, базирующихся на принципах что без как, способствовал международный конгресс IFIP (International Federation of Information Processing- Международная федерация по обработке информации), прошедший в Стокгольме в 1976 г. под лозунгом "В 1980г. программирование без программистов". В рамках реализации этой идеи были созданы непроцедурные языки, приближающиеся по своему синтаксису к естественному языку и ориентированные на пользователей - специалистов управления, не являющихся программистами.

Наиболее известными из языков этого типа являются:

Smalltalking - малый разговорный;

QBE (Query By Example-программирование на примере);

Форт, который находит применение при решении сложных задач имитационного моделирования, в системах искусственного интеллекта в графических системах и т.п.

Основной особенностью языка Форт является его открытость, которая позволяет на основе имеющихся определений строить новые функции. При этом программист может вводить новые операции, типы данных или определения. Возможность поддержки средствами Форт многозадачного режима работы придают ему свойства операционной системы.

Особое место среди языков программирования занимают функциональные языки, в частности Пролог (PROLOG -PROgram-ming in LOGic - логическое программирование), предложенный А.Калмероз в 1978 г., являющийся языком логического программирования, относящимся к языкам пятого поколения. Главное назначение языка - разработка интеллектуальных программ и систем. Пролог - это язык программирования, созданный специально для работы с базами знаний, основанными на фактах и правилах (одного из элементов систем искусственного интеллекта). В языке реализован механизм возврата для выполнения обратной цепочки рассуждений, при котором предполагается, что некоторые выводы или заключения истинны, а затем эти предположения проверяются в базе знаний, содержащей факты и правила логического вывода. Если предположение не подтверждается, выполняется возврат и выдвигается новое предположение. Языковые средства СУБД предназначены в первую очередь для разработки прикладных программ решения задач экономического управления, информация для которых хранится и поддерживается с помощью баз данных. Синтаксис языка программирования в среде СУБД мало чем отличается от синтаксиса высокоуровневых языков программирования, в связи с чем указанные программно-инструментальные средства ориентированы в основном на профессиональных программистов, хотя наличие развитых средств подсказки и помощи (в виде примеров, демонстрирующих использование отдельных языковых конструкций) значительно облегчает работу достаточно широкого круга пользователей.

Sequel (Structured English QUERy Language) и его усовершенствованный вариант SQL - языки манипулирования данными, основанные на исчислении отношений. Используются в реляционных СУБД в качестве языка запросов к базам данных и языка программирования задач обработки данных.

С развитием компьютерных сетей, увеличением вычислительной мощности компьютеров и их ресурсов возникла потребность в интерпретирующем языке, позволяющем получать многоплатформенную вычислительную среду путем преобразования с его помощью программ, написанных на других языках программирования (при незначительном снижении их производительности). Наиболее близко к реализации подобного языка подошла технология языка Java (а точнее ее часть - байт-код). Именно его разработка и использование составляют принципиальное отличие, которое выделяет язык Java среди других языков программирования высокого уровня. Объектно-ориентированный язык Java (разработанный на базе языка Си++) предназначен для создания надежных, переносимых, распределенных сетевых программных приложений, работающих в различных многооконных системах в условиях архитектуры клиент-сервер, а также для администраторов сети, использующих Java-приложения для улучшения интерактивных качеств Web-серверов.

Другим объектно-ориентированным языком программирования является язык Delphi. Созданный на базе языка Паскаль специалистами фирмы Borland язык Delphi, обладая мощностью и гибкостью языков Си и Си++, превосходит их по удобству и простоте интерфейса при разработке приложений, обеспечивающих взаимодействие с базами данных и поддержку различного рода работ в рамках корпоративных сетей и сети Интернет.

В последние годы резко расширилась практика программирования в среде электронных таблиц. В основе реализации программирования задач с помощью электронных таблиц (ЭТ) лежит идея компьютеризации работы пользователей с "пустографкой" как специфической формой представления документа, с которым обычно приходится иметь дело специалистам управления. Разработчики электронных таблиц (впоследствии названных табличными процессорами) воплотили эту идею путем предоставления пользователям возможности записи в любую клетку электронной таблицы цифровой, символьной (текстовой) информации либо формулы, обеспечивающей получение искомого результата по различным исходным данным.

Пользователь ЭТ получил удобное инструментальное средство, позволяющее: вводить исходные данные, необходимые для решения задач; указывать формулы получения результатных данных; оформлять решение задачи в виде привычных для него табличных документов. При этом способ описания расчетных формул почти ничем не от-

личается от принятых правил их представления в математике (за исключением требования линейной структуры записи). Порядок реализации арифметических действий в формулах совпадает с принятым приоритетом выполнения операций в математике, а для изменения такого порядка используется система круглых скобок.

Единственным наследием, доставшимся пользователям ЭТ от программирования на высокоуровневых языках, осталось правило построения логических выражений, реализуемое по стандартной для языков программирования схеме построения условных Операторов: IF - THEN - ELSE (если условие соблюдается, то выполняется действие-1, иначе осуществляется действие-2).

Быстрому и широкому распространению ЭТ как инструментальному средству решения экономических задач, помимо простоты и удобства подготовки решения с их помощью задач, способствовали также:

наличие большого числа встроенных функций (математических, статистических, финансовых и т.п.);

возможность "проигрывания" различных вариантов решения задач и выбора лучшего из них (за счет быстрого автоматического пересчета конечных результатов при любом изменении исходных данных);

поддержка ЭТ средствами графической интерпретации, наглядное представление результатной информации.

Кроме того, работа с современными электронными таблицами может быть автоматизирована благодаря использованию командных языков, макросов и т.п.

Межкомпьютерная связь

Назовём задачи, которые трудно или невозможно решить без организации информационной связи между различными компьютерами:

- **перенос информации на большие расстояния** (сотни, тысячи километров);
- **совместное использование несколькими компьютерами дорогостоящих аппаратных, программных или информационных ресурсов** — мощного процессора, ёмкого накопителя, высокопроизводительного лазерного принтера, баз данных, программного обеспечения и т.д.;

- **совместная работа над большим проектом**, когда исполнили должны всегда иметь последние (актуальные) копии общих данных во избежание путаницы, и т.д.

Есть **три основных способа организации межкомпьютерной связи**:

- объединение двух рядом расположенных компьютеров через их коммуникационные порты посредством специального **кабеля**;
- передача данных от одного компьютера к другому посредством **модема** с помощью проводных или спутниковых линий связи;
- объединение компьютеров в **компьютерную сеть**.

Часто при организации связи между двумя компьютерами за **одним компьютером** закрепляется роль поставщика ресурсов (программ, данных и т.д.), а за другим — роль пользователя этих ресурсов. В этом случае первый компьютер называется **сервером**, а второй — **клиентом** или рабочей станцией. Работать можно только на компьютере-клиенте под управлением специального программного обеспечения.

Сервер (англ. *serve* — обслуживать) — это высокопроизводительный компьютер с большим объёмом внешней памяти, который обеспечивает *обслуживание* других компьютеров путем управления распределением дорогостоящих ресурсов совместного пользования (программ, данных и периферийного оборудования).

Клиент (иначе, рабочая станция) — любой компьютер, имеющий доступ к услугам сервера.

Например, сервером может быть мощный компьютер, на котором размещается центральная база данных, а клиентом — обычный компьютер, программы которого по мере необходимости запрашивают данные с сервера. В некоторых случаях компьютер может быть **одновременно и клиентом, и сервером**. Это значит, что он может предоставлять свои ресурсы и хранимые данные другим компьютерам и одновременно использовать их ресурсы и данные.

Клиентом также называют *прикладную программу*, которая от имени пользователя *получает услуги сервера*. Соответственно, программное обеспечение, которое позволяет компьютеру *предоставлять услуги* другому компьютеру, называют *сервером* — так же, как и сам компьютер.

Для преодоления **несовместимости интерфейсов** отдельных компьютеров вырабатывают специальные стандарты, называемые протоколами коммуникации.

Протокол коммуникации — это согласованный набор конкретных правил обмена информацией между разными устройствами передачи данных. Имеются протоколы для скорости передачи, форматов данных, контроля ошибок и др.

Для работы с сетью необходимо наличие специального **сетевого программного обеспечения**, которое обеспечивает передачу данных в соответствии с заданным протоколом.

Протоколы коммуникации предписывают разбить весь объём передаваемых данных на *пакеты* — отдельные блоки фиксированного размера. *Пакеты нумеруются*, чтобы их затем можно было собрать в правильной последовательности. К данным, содержащимся в пакете, добавляется дополнительная информация примерно такого формата:

Адрес получателя	Адрес отправителя	Длина	Данные	Поле контрольной суммы
---------------------	----------------------	-------	--------	------------------------------

Контрольная сумма данных пакета содержит информацию, необходимую для контроля ошибок. Первый раз она вычисляется передающим компьютером. После того, как пакет будет передан, контрольная сумма повторно вычисляется принимающим компьютером. Если значения не совпадают, это означает, что данные пакета были *повреждены при передаче*. Такой пакет отбрасывается, и автоматически направляется запрос *повторно передать пакет*.

При установлении связи устройства обмениваются сигналами для согласования коммуникационных каналов и протоколов. Этот процесс называется *подтверждением установления связи* (англ. *HandShake* — рукопожатие).

Компьютерная сеть

Компьютерная сеть (англ. *Computer NetWork*, от *net* — сеть, и *work* — работа) — это система обмена информацией между компьютерами. Представляет собой совокупность трех компонент:

- сети передачи данных (включающей в себя каналы передачи данных и средства коммутации);
- компьютеров, взаимосвязанных сетью передачи данных;
- сетевого программного обеспечения.

Пользователи компьютерной сети получают возможность совместно использовать её программные, технические, информационные и организационные ресурсы.

Компьютерная сеть представляет собой совокупность *узлов* (компьютеров, рабочих станций и др.) и соединяющих их *ветвей*.

Ветвь сети — это путь, соединяющий два смежных узла.

Узлы сети бывают трёх типов:

- ***оконечный*** узел — расположен в конце только одной ветви;
- ***промежуточный*** узел — расположен на концах более чем одной ветви;
- ***смежный*** узел — такие узлы соединены по крайней мере одним путём, не содержащим никаких других узлов.

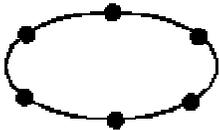
Компьютеры могут объединяться в сеть разными способами.

Способ соединения компьютеров в сеть называется её *топологией*.

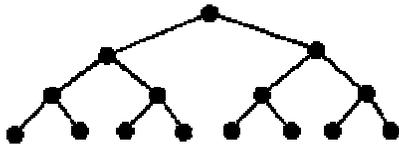
Наиболее распространенные виды топологий сетей:



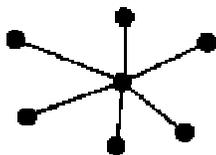
Линейная сеть. Содержит только два оконечных узла, любое число промежуточных узлов и имеет только один путь между любыми двумя узлами.



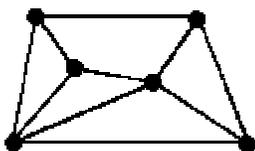
Кольцевая сеть. Сеть, в которой к каждому узлу присоединены две и только две ветви.



Древовидная сеть. Сеть, которая содержит более двух оконечных узлов и по крайней мере два промежуточных узла, и в которой между двумя узлами имеется только один путь.



Звездообразная сеть. Сеть, в которой имеется только один промежуточный узел.



Ячеистая сеть. Сеть, которая содержит по крайней мере два узла, имеющих два или более пути между ними.

Полносвязанная сеть. Сеть, в которой имеется ветвь между любыми двумя узлами. Важнейшая характеристика компьютерной сети — её архитектура.

Архитектура сети — это реализованная структура сети передачи данных, определяющая её *топологию, состав устройств и правила их взаимодействия в сети*. В рамках архитектуры сети рассматриваются вопросы кодирования информации, её адресации и передачи, управления потоком сообщений, контроля ошибок и анализа работы сети в аварийных ситуациях и при ухудшении характеристик.

Наиболее распространённые архитектуры:

- **Ethernet** (англ. *ether* — эфир) — широковещательная сеть. Это значит, что все станции сети могут принимать все сообщения. Топология — линейная или звездообразная. Скорость передачи данных 10 или 100 Мбит/сек.
- **Arcnet** (*Attached Resource Computer Network* — компьютерная сеть соединённых ресурсов) — широковещательная сеть. Физическая топология — дерево. Скорость передачи данных 2,5 Мбит/сек.
- **Token Ring** (эстафетная кольцевая сеть, сеть с передачей маркера) — кольцевая сеть, в которой принцип передачи данных основан на том, что каждый узел кольца ожидает прибытия некоторой короткой уникальной последовательности битов — *маркера* — из смежного предыдущего узла. Поступление маркера указывает на то, что можно передавать сообщение из данного узла дальше по ходу потока. Скорость передачи данных 4 или 16 Мбит/сек.
- **FDDI** (*Fiber Distributed Data Interface*) — сетевая архитектура высокоскоростной передачи данных по оптоволоконным линиям. Скорость передачи — 100 Мбит/сек. Топология — двойное кольцо или смешанная (с включением звездообразных или древовидных подсетей). Максимальное количество станций в сети — 1000. Очень высокая стоимость оборудования.
- **ATM** (*Asynchronous Transfer Mode*) — перспективная, пока ещё очень дорогая архитектура, обеспечивает передачу цифровых данных, видеoinформации и голоса по одним и тем же линиям. Скорость передачи до 2,5 Гбит/сек. Линии связи оптические.

Эталонная модель взаимодействия открытых систем

Международная организация по стандартизации (ISO), основываясь на опыте многомашинных систем, который был накоплен в разных странах, выдвинула концепцию архитектуры открытых систем - эталонную модель, используемую при разработке международных стандартов.

На основе этой модели вычислительная сеть предстает как распределенная вычислительная среда, включающая в себя большое число разнообразных аппаратных и программных средств. По вертикали данная среда представляется рядом логических уровней, на каждый из которых возложена одна из задач сети. По горизонтали информационно-вычислительная среда делится на локальные части (открытые системы), отвечающие требованиям и стандартам структуры открытых систем.

Часть открытой системы, выполняющая некоторую функцию и входящая в состав того или иного уровня, называется объектом.

Правила, по которым осуществляется взаимодействие объектов одного и того же уровня, называются протоколом (методика связи).

Протоколы определяют порядок обмена информацией между сетевыми объектами. Они позволяют взаимодействующим рабочим станциям посылать друг другу вызовы, интерпретировать данные, обрабатывать ошибочные ситуации и выполнять множество других различных функций. Суть протоколов заключается в регламентированных обменах точно специфицированными командами и ответами на них (например, назначение физического уровня связи - передача блоков данных между двумя устройствами, подключенными к одной физической среде).

Каждый уровень подразделяется на две части:

- спецификацию услуг;
- спецификацию протокола.

Спецификация услуг определяет, что делает уровень, а спецификация протокола - как он это делает.

Причем, каждый конкретный уровень может иметь более одного протокола.

Большое число уровней, используемых в модели, обеспечивает декомпозицию информационно-вычислительного процесса на простые составляющие. В свою очередь, увеличение числа уровней вызывает необходимость включения дополнительных связей

в соответствии с дополнительными протоколами и интерфейсами. Интерфейсы (макрокоманды, программы) зависят от возможностей используемой ОС.

Международная организация по стандартизации предложила семиуровневую модель, которой соответствует и программная структура (рис.5.).



Рис. 4. 1. Уровни управления и протоколы ЛВС

Рассмотрим функции, выполняемые каждым уровнем программного обеспечения:

1. Физический - осуществляет как соединения с физическим каналом, так и расторжение, управление каналом, а также определяется скорость передачи данных и топология сети.

2. Канальный - осуществляет обрамление передаваемых массивов информации вспомогательными символами и контроль передаваемых данных. В ЛВС передаваемая информация разбивается на несколько пакетов или кадров. Каждый пакет содержит адреса источника и места назначения, а также средства обнаружения ошибок.

3. Сетевой - определяет маршрут передачи информации между сетями (ПЭВМ), обеспечивает обработку ошибок, а так же управление потоками данных.

Основная задача сетевого уровня - маршрутизация данных (передача данных между сетями). Специальные устройства - *Маршрутизаторы (Router)* определяют для ка-

кой сети предназначено то или другое сообщение, и направляет эту посылку в заданную сеть. Для определения абонента внутри сети используется *Адрес Узла (Node Address)*. Для определения пути передачи данных между сетями на маршрутизаторах строятся *Таблицы Маршрутов (Routing Tables)*, содержащие последовательность передачи данных через маршрутизаторы. Каждый маршрут содержит адрес конечной сети, адрес следующего маршрутизатора и стоимость передачи данных по этому маршруту. При оценке стоимости могут учитываться количество промежуточных маршрутизаторов, время, необходимое на передачу данных, просто денежная стоимость передачи данных по линии связи. Для построения таблиц маршрутов наиболее часто используют либо *Метод Векторов* либо *Статический Метод*. При выборе оптимального маршрута применяют динамические или статические методы. На сетевом уровне возможно применение одной из двух процедур передачи пакетов:

- *датаграмм* - т.е., когда часть сообщения или пакет независимо доставляется адресату по различным маршрутам, определяемым сложившейся динамикой в сети. При этом каждый пакет включает в себя полный заголовок с адресом получателя. Процедуры управления передачей таких пакетов по сети называются датаграммной службой;

- *виртуальных соединений* - когда установление маршрута передачи всего сообщения от отправителя до получателя осуществляется с помощью специального служебного пакета - запроса на соединение. В таком случае для этого пакета выбирается маршрут и, при положительном ответе получателя на соединение закрепляется для всего последующего трафика (потока сообщений в сети передачи данных) и получает номер соответствующего виртуального канала (соединения) для дальнейшего использования его другими пакетами того же сообщения. Пакеты, которые передаются по одному виртуальному каналу, не являются независимыми и поэтому включают сокращенный заголовок, включающий порядковый номер пакета, принадлежащему одному сообщению.

Недостатки: значительная по сравнению с датаграммой сложность в реализации, увеличение накладных расходов, вызванных установлением и разъединением сообщений.

ВЫВОД. Датаграммный режим предпочтительнее использовать для сетей сложной конфигурации, где значительное число ЭВМ в сети, иерархическая структура сети, надежность, достоверность передачи данных по каналам связи, длина пакета более 512 байт.

4. Транспортный - связывает нижние уровни (физический, канальный, сетевой) с верхними уровнями, которые реализуются программными средствами. Этот уровень как бы разделяет средства формирования данных в сети от средств их передачи. Здесь осуществляется разделение информации по определенной длине и уточняется адрес назначения. Транспортный уровень позволяет мультиплексировать передаваемые сообщения или соединения. Мультиплексирование сообщений позволяет передавать сообщения одновременно по нескольким линиям связи, а мультиплексирование соединений - передает в одной посылке несколько сообщений для различных соединений.

5. Сеансовый - на данном уровне осуществляется управление сеансами связи между двумя взаимодействующими пользователями (определяет начало и окончание сеанса связи: нормальное или аварийное; определяет время, длительность и режим сеанса связи; определяет точки синхронизации для промежуточного контроля и восстановления при передаче данных; восстанавливает соединение после ошибок во время сеанса связи без потери данных.

6. Представительский - управляет представлением данных в необходимой для программы пользователя форме, генерацию и интерпретацию взаимодействия процессов, кодирование/декодирование данных, в том числе компрессию и декомпрессию данных. На рабочих станциях могут использоваться различные операционные системы : DOS, UNIX, OS/2. Каждая из них имеет свою файловую систему, свои форматы хранения и обработки данных. Задачей данного уровня является преобразование данных при передаче информации в формат, который используется в информационной системе. При приеме данных данный уровень представления данных выполняет обратное преобразование. Таким образом появляется возможность организовать обмен данными между станциями, на которых используются различные операционные системы.

Форматы представления данных могут различаться по следующим признакам:

- порядок следования битов и размерность символа в битах;
- порядок следования байтов;
- представление и кодировка символов;
- структура и синтаксис файлов.

Компрессия или упаковка данных сокращает время передачи данных. Кодирование передаваемой информации обеспечивает защиту ее от перехвата.

7. Прикладной - в его ведении находятся прикладные сетевые программы, обслуживающие файлы, а также выполняет вычислительные, информационно-поисковые работы, логические преобразования информации, передачу почтовых сообщений и т.п. Главная задача этого уровня - обеспечить удобный интерфейс для пользователя.

На разных уровнях обмен происходит различными единицами информации: биты, кадры, пакеты, сеансовые сообщения, пользовательские сообщения.

Средства организации экономических информационных систем

В отличие от централизованной обработки данных, связанной с концентрацией основных вычислительных мощностей в ВЦ, есть возможность отказаться от этой в значительной мере "искусственной тенденции" и проводить обработку информации в местах ее непосредственного возникновения и использования. Это позволяет ликвидировать промежуточные звенья при общении человека с ЭВМ. В результате все технологические процедуры, начиная от ввода информации и кончая получением выходных данных, могут выполняться работниками управления непосредственно на своих рабочих местах.

Системы обработки данных (СОД) на базе концепции АРМ получили широкое развитие.

АРМ - автоматизированное рабочее место системы управления, оборудованное средствами, обеспечивающими участие человека в реализации автоматизированных функций АСУ.

АРМ присущи следующие признаки:

- доступная пользователю совокупность технических, программных, информационных и др. средств;
- размещение ВТ непосредственно (или вблизи) на рабочем месте пользователя;
- возможность создания и совершенствования проектов автоматизированной обработки данных в конкретной сфере деятельности;
- осуществление обработки данных самим пользователем;

- диалоговый режим взаимодействия пользователя с ЭВМ как в процессе решения задач управления, так и в процессе их проектирования.

Таким образом АРМ в системе управления представляет собой проблемно-ориентированный комплекс технических, программных, лингвистических (языковых) и др. средств, установленный непосредственно на рабочем месте пользователя и предназначенный для автоматизации операций взаимодействия пользователя с ЭВМ в процессе проектирования и реализации задач.

Множество известных АРМ может быть *классифицировано* на основе следующих обобщенных признаков:

- функциональная сфера использования (научная деятельность, проектирование, производственно-технологические процессы, организационное управление);
- тип используемой ЭВМ (микро-, мини-, макро ЭВМ);
- режим эксплуатации (индивидуальный, групповой, сетевой);
- квалификация пользователей (профессиональные и непрофессиональные).

Внутри каждой из выделенных групп АРМ может быть проведена более детальная классификация.

Например, АРМы организационного управления могут быть разделены на АРМ руководителей организаций и подразделений, плановых работников, работников материально-технического снабжения, бухгалтеров и др. Условно все эти АРМ-ы можно называть *АРМ - экономиста*.

Концептуальное отличие АРМ на базе ПЭВМ состоит в том, что АРМ открытая архитектура ПЭВМ функционально, физически и эргономически настраивается на конкретного пользователя (персональный АРМ) или группу пользователей (групповой АРМ).

Деловые АРМ сближают пользователя с возможностями современной информатики и ВТ и создают условия для работы без посредника - профессионального программиста. При этом обеспечивается как автономная работа, так и возможность связи с другими пользователями в пределах организационных структур (с учетом особенностей этих структур).

Параметрический ряд деловых АРМ позволяет создать единую техническую, организационную и методологическую базу компьютеризации управления. Первона-

начально информационная технология локализуется в пределах персонального или группового АРМ, а в последующем (при объединении АРМ средствами коммуникации) создаются АРМ сектора, отдела, учреждения и формируется коллективная технология. Тем самым достигается гибкость всей структуры и возможность наращивания информационной мощности.

Можно выделить три класса типовых АРМ:

- АРМ руководителя;
- АРМ специалиста;
- АРМ технического и вспомогательного персонала.

Состав функциональных задач и видов работ (административно-организационный, профессионально-творческий, технический...) требует применения различных инструментальных средств при создании АРМ.

Например, административно-организационная работа - контроль исполнения, анализ текущего состояния дел и планирования работы...; профессионально-творческая - разработка документов, анализ информации, ...; техническая работа - получение, передача, хранение, печать документов, сводок, контроль за движением документов.

Для автоматизации каждой категории работ в настоящее время ПЭВМ оснащены различными ППП.

Программное обеспечение любого АРМ (рис.3.1) подразделяется на бщее функциональное:

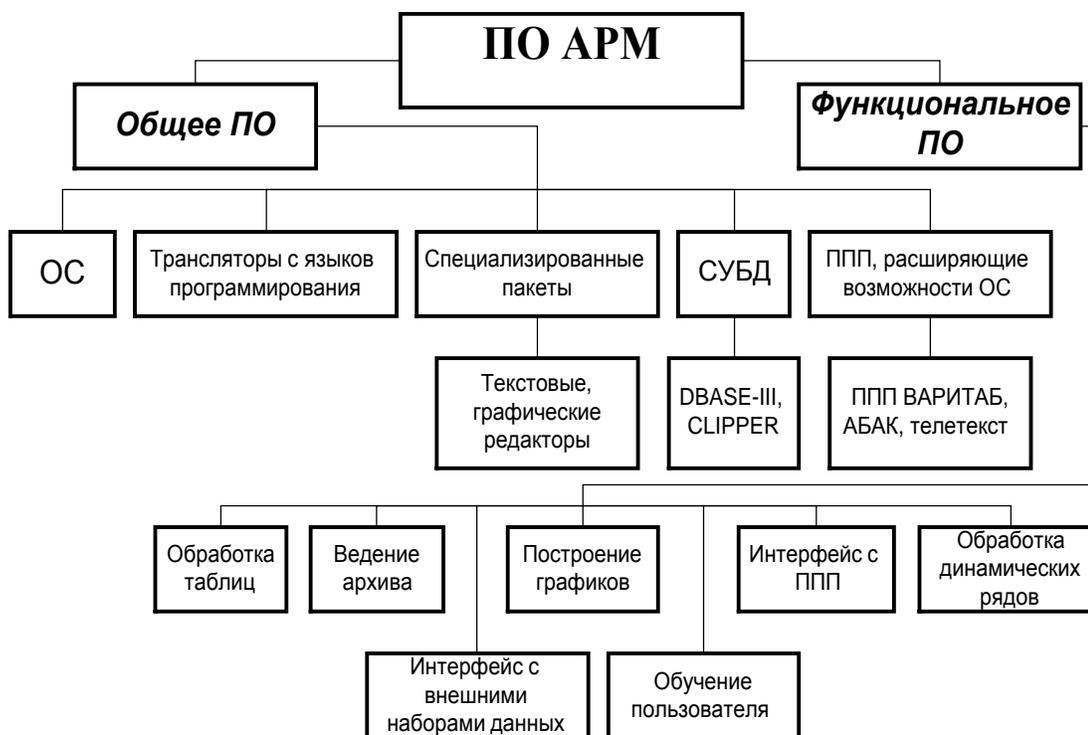


Рис.3.1. Схема программного обеспечения АРМ

При проектировании ПО АРМ необходимо соблюдать принцип ориентации разрабатываемых программных средств на конкретного пользователя, что должно обеспечить реализацию функций, соответствующих профессиональной ориентации АРМ. В целом, разрабатываемое ПО АРМ должно обладать свойствами гибкости, адаптивности, модифицируемости и настраиваемости на конкретное применение.

АРМ должен быть укомплектован необходимыми программно-инструментальными средствами:

- операционные системы ЭВМ;
- трансляторы (интерпретаторы) с различных алгоязыков и языков пользователей;
- средства проектирования и обработки данных (экранные редакторы текстовой, графической информации, СУБД, табличные процессоры, генераторы выходных форм);
- собственно пользовательские программы (обрабатывающие, обучающие, СУБД знаний и др.).

Следует отметить, что АРМ включает в себя следующие основные элементы: ЭВМ; программно-инструментальные средства, БД и Бзнаний пользователя. Комплек-

тация АРМ техническими и программными средствами, а также перечисленными выше элементами зависит от назначения и состава решаемых задач. Решение экономических задач на основе АРМ связано с поиском требуемой информации в информационной базе, последующей ее обработкой по расчетным алгоритмам и выдачей результатов на экран или печать. Эффективная эксплуатация АРМ требует использования языков общения пользователя с ЭВМ. Наиболее развитые средства общения пользователя с ЭВМ реализуются *лингвистическими* процессорами, способными осуществить различные виды анализа входного сообщения (синтаксический, морфологический, семантический), и ориентированными на работу с конкретной предметной областью. В АРМ-е часто общение основывается на макетировании изображений экрана в виде образцов-прототипов документов. Для этого используются разнообразные технические приемы обеспечения диалога пользователя и ЭВМ: управление положением курсора на экране с применением светового пера, мерцание и подсветка полей экрана, программирование функциональных клавиш.

Диалог реализуется на основе предварительно разработанного сценария, который представляется семантическими сетями, таблицами диалога, фреймами (структуры данных нового типа, на основе которых строятся интеллектуальные БД) и др. средствами, используемыми для задания моделей предметной области.

Описанные функциональные возможности АРМ реализуются совокупностью программных компонентов.

Каждый из программных компонентов выполняет широкий набор действий и в большинстве случаев может использоваться независимо от других. Центральным компонентом, без которого невозможна работа других средств является ОС. Она обеспечивает: создание и актуализацию каталога файлов различных типов, просмотр каталогов и распечатку файлов, переименование и редактирование файлов, защиту файлов, распределение внешней памяти и др.

Специальные версии ОС (например, CP/NET), предоставляют средства, с помощью которых несколько ЭВМ, оснащенных ОС CP/M, можно объединить в локальную сеть для совместного использования ресурсов каждой системы. К таким ресурсам относятся диски, устройства печати, различные программы и БД.

В состав ПО АРМ-экономиста входят средства управления информационной базой, обеспечивающие:

- создание и актуализацию информационной базы;
 - поиски требуемой информации по регламентируемым и нерегламентируемым запросам;
- организацию форматного ввода-вывода информации;
- вычислительную обработку и др.

Для представления данных в информационной базе часто используется реляционная модель. Наряду с реляционными СУБД применяются табличные процессоры. В этом случае входные и выходные данные и НСИ представляются в форме таблиц, алгоритмизация сводится к построению модели расчета показателей выходных документов (ППП Excel). К этой же группе относятся интегрированные СУИБ (Works), которые реализуют функции табличных процессоров, СУБД, редакторов текстов, генераторов выходных документов (ППП SIMPHJNY, LOTUS).

В АРМ-экономиста необходимо обеспечить выдачу информации в графической форме для всестороннего анализа экономических показателей. Графики могут иметь двумерное (плоское) и трехмерное (объемное) представление. С помощью графических средств осуществляется планирование (разметка) площади экрана, изображение на экране графических элементов в виде линии, точки, отрезка, прямоугольника, эллипса, штриховки графических элементов с использованием требуемых цветов, подбора шрифта и т.п.

Организация экранного диалога - одно из основных требований к технологии АРМ:

1. технология ЭО текстов реализует четыре функции:

- функции ввода - набора текста с заданием параметров для его верстки, просмотра;
- обработки (смысловая сортировка текста, вычисления в табл.);
- воспроизведение текста;
- форматирование текста и получение документа.

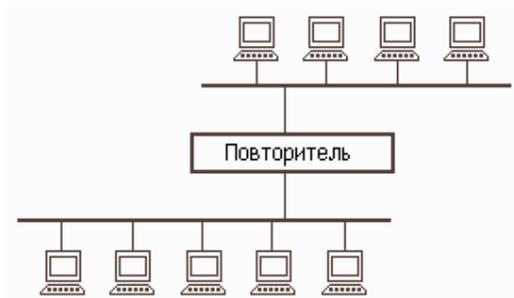
2. технология ЭО форм (электронные таблицы, шаблоны).

3. система ЭО деловой графики (в виде графиков и диаграмм - столбиковых, линейных, круговых, секторных диаграмм, гистограмм и др.). Например, интегрированный ППП FRAMEWORK (РАДУГА) - работа с рамками, окнами, иерархическими фреймами.

Соединение между собой устройств сети

Для этого используется специальное оборудование:

- **Сетевые кабели** (*коаксиальные*, состоящие из двух изолированных между собой концентрических проводников, из которых внешний имеет вид трубки; *оптоволоконные*; кабели на *витых парах*, образованные двумя переплетёнными друг с другом проводами, и др.).
- **Коннекторы** (*соединители*) для подключения кабелей к компьютеру; **разъёмы** для соединения отрезков кабеля.
- **Сетевые интерфейсные адаптеры для приёма и передачи данных.** В соответствии с определённым протоколом управляют доступом к среде передачи данных. Размещаются в системных блоках компьютеров, подключенных к сети. К разъёмам адаптеров подключается сетевой кабель.
- **Трансиверы** повышают уровень качества передачи данных по кабелю, отвечают за приём сигналов из сети и обнаружение конфликтов.
- **Хабы** (*концентраторы*) и **коммутирующие хабы** (*коммутаторы*) расширяют топологические, функциональные и скоростные возможности компьютерных сетей. Хаб с набором разнотипных портов позволяет *объединять сегменты сетей с различными кабельными системами*. К порту хаба можно подключать как отдельный узел сети, так и другой хаб или сегмент кабеля.

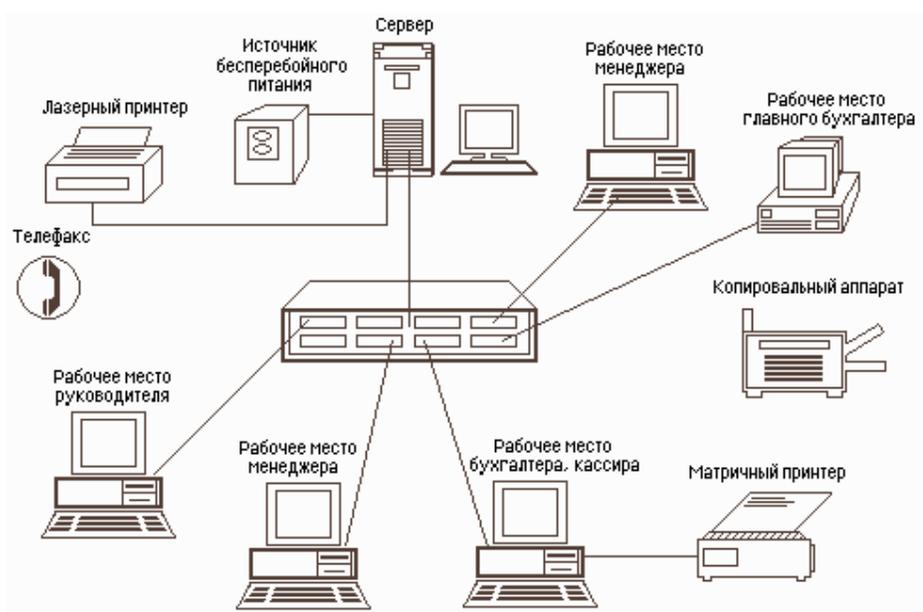


- **Повторители** (*репитеры*) усиливают сигналы, передаваемые по кабелю при его большой длине.

Классификация компьютерных сетей по степени географического распространения

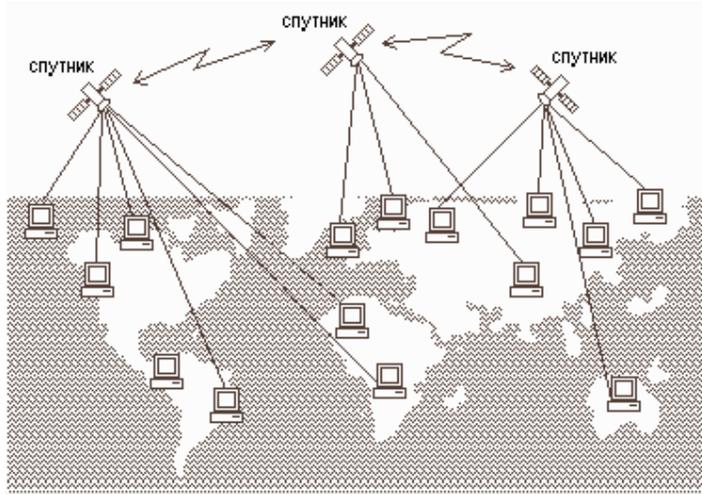
По степени географического распространения сети делятся на локальные, городские, корпоративные, глобальные и др.

Локальная сеть (ЛВС или LAN — *Local Area NetWork*) — сеть, связывающая ряд компьютеров в зоне, ограниченной пределами одной комнаты, здания или предприятия.



Небольшая офисная локальная сеть

Глобальная сеть (ГВС или WAN — *World Area NetWork*) — сеть, соединяющая компьютеры, удалённые географически на большие расстояния друг от друга. Отличается от локальной сети более протяженными коммуникациями (спутниковыми, кабельными и др.). Глобальная сеть объединяет локальные сети.



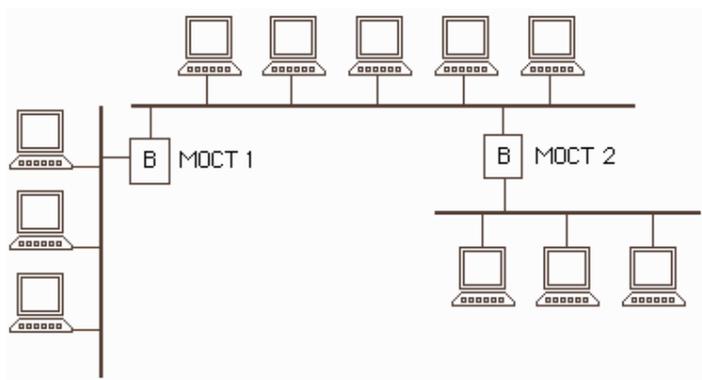
Глобальная сеть

Городская сеть (MAN — *Metropolitan Area NetWork*) — сеть, которая обслуживает информационные потребности большого города.

Соединение между собой локальных сетей

Для соединения локальных сетей используются следующие устройства, которые различаются между собой по назначению и возможностям:

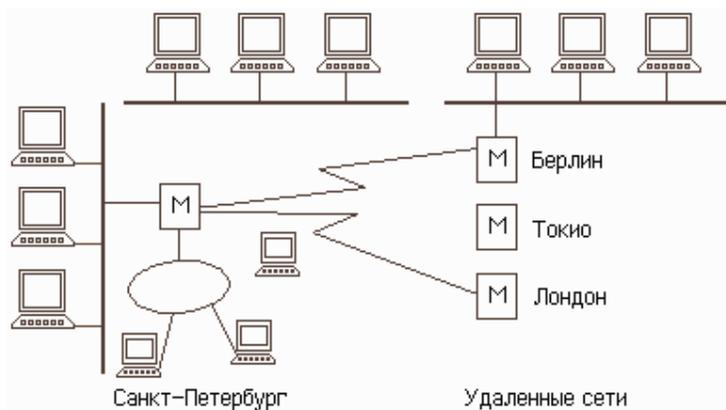
- **Мост** (англ. *Bridge*) — связывает две локальные сети. **Передаёт данные между сетями в пакетном виде, не производя в них никаких изменений.** Ниже на рисунке показаны три локальные сети, соединённые двумя мостами.



Соединение локальных сетей посредством мостов

Здесь мосты создали **расширенную сеть**, которая обеспечивает своим пользователям **доступ к прежде недоступным ресурсам.** Кроме этого, мосты могут **фильтро-**

вать пакеты, охраняя всю сеть от локальных потоков данных и пропуская наружу только те данные, которые предназначены для других сегментов сети.



- **Маршрутизатор** (англ. *Router*) объединяет сети с общим протоколом более эффективно, чем мост. Он позволяет, например, расщеплять большие сообщения на более мелкие куски, обеспечивая тем самым взаимодействие локальных сетей с разным размером пакета.

Маршрутизатор может пересылать пакеты на конкретный адрес (мосты только отфильтровывают ненужные пакеты), выбирать лучший путь для прохождения пакета и многое другое. Чем сложнее и больше сеть, тем больше выгода от использования маршрутизаторов.

- **Мостовой маршрутизатор** (англ. *Brouter*) — это гибрид моста и маршрутизатора, который сначала пытается выполнить маршрутизацию, где это только возможно, а затем, в случае неудачи, переходит в режим моста.

- **Шлюз** (англ. *GateWay*), в отличие от моста, применяется в случаях, когда соединяемые сети имеют **различные сетевые протоколы**. Поступившее в шлюз сообщение от одной сети преобразуется в другое сообщение, соответствующее требованиям следующей сети. Таким образом, шлюзы не просто соединяют сети, а позволяют им работать как единая сеть. С помощью шлюзов также локальные сети подсоединяются к **мэйнфреймам** — универсальным мощным компьютерам.

Беспроводные сети

Беспроводные сети используются там, где прокладка кабелей затруднена, нецелесообразна или просто невозможна. Например, в исторических зданиях, промышленных помещениях с металлическим или железобетонным полом, в офисах, полученных в краткосрочную аренду, на складах, выставках, конференциях и т.п.

В этих случаях сеть реализуется при помощи *сетевых радио-адаптеров*, снабжённых *всенаправленными антеннами* и использующих в качестве среды передачи информации *радиоволны*. Такая сеть реализуется топологией “**Все-Со-Всеми**” и работоспособна при дальности 50–200 м.

Для связи между беспроводной и кабельной частями сети используется специальное устройство, называемое *точкой входа* (или *радиомостом*). Можно использовать и обычный компьютер, в котором установлены два сетевых адаптера — *беспроводной* и *кабельный*.

Другой важной областью применения беспроводных сетей является **организация связи между удалёнными сегментами локальных сетей при отсутствии инфраструктуры передачи данных** (кабельных сетей общего доступа, высококачественных телефонных линий и др.), что типично для нашей страны. В этом случае для наведения беспроводных мостов между двумя удалёнными сегментами используются *радиомосты с антенной направленного типа*.

Если в сеть нужно объединить **несколько сегментов**, то используется топология типа “звезда”. При этом в *центральной* узле устанавливается *всенаправленная антенна*, а *удалённых* узлах — *направленные*. Сети звездообразной топологии могут образовывать сети разнообразной конфигурации.

Сетевая магистраль с беспроводным доступом позволяет отказаться от использования медленных модемов.

Сеть Интернет

Интернет (англ. *Internet* — *между сетей*) — гигантская всемирная компьютерная сеть, объединяющая десятки тысяч сетей всего мира. Её *назначение* — обеспечить лю-

бому желающему постоянный доступ к любой информации. Интернет *предлагает* практически неограниченные информационные ресурсы, полезные сведения, учёбу, развлечения, возможность общения с компетентными людьми, услуги удалённого доступа, передачи файлов, электронной почты и многое другое. Интернет *обеспечивает* принципиально новый способ общения людей, не имеющий аналогов в мире.

Благодаря сети стал доступен (бесплатно или за умеренную плату) огромный объём информации. Так, пользователь в любой стране может связаться с людьми, разделяющими его интересы, или получить ценные сведения в электронных библиотеках, даже если они находятся на другом конце света.

Нужная информация окажется в его компьютере *за считанные секунды*, пройдя путь по длинной цепочке промежуточных компьютеров, по кабелям и по радио, через горы и моря, по дну океана и через спутник.

Интернет финансируется правительствами, научными и образовательными учреждениями, коммерческими структурами и миллионами частных лиц во всех частях света, но *никто конкретно не является её владельцем*. Управляет сетью “Совет по архитектуре Интернет”, формируемый из приглашённых добровольцев.

Сеть была создана в 1984 году, и сейчас ею пользуются примерно сорок миллионов человек. Интернет всё время изменяется, поскольку имеет много квалифицированных пользователей, которые пишут программы для себя, а затем распространяют их среди желающих. Постоянно появляются новые серверы, а существующие обновляют свой “репертуар”. Стремительно растут информационные потоки.

Как можно связаться с Интернет? Самый распространенный и недорогой способ — посредством модема и телефонной линии. При этом используются три типа подключения, отличающиеся друг от друга по объёму услуг и цене:

1. ***почтовое*** — позволяет только обмениваться электронной почтой с любым пользователем Интернет, самое дешёвое;
2. ***сеансное*** в режиме *on-line* (“на прямом проводе”) — работа в диалоговом режиме — все возможности сети на время сеанса;

3. **прямое** (личное), самое дорогостоящее — все возможности в любое время.

При работе в сеансном режиме доступ к Интернет обычно покупается у **провайдеров** (англ. *provide* — предоставлять, обеспечивать) — фирм, предоставляющих доступ к некоторой части Интернет и поставляющих её пользователям разнообразные услуги.

Отдельные участки Интернет представляют собой сети различной архитектуры, которые связываются между собой с помощью **маршрутизаторов**. Передаваемые данные разбиваются на небольшие порции, называемые **пакетами**. Каждый пакет перемещается по сети **независимо** от других пакетов.

Сети в Интернет **неограниченно коммутируются** (т.е. связываются) друг с другом, потому что все компьютеры, участвующие в передаче данных, используют **единый протокол коммуникации ТСП/ИР** (читается “*ти-си-ни/ай-ни*”).

На самом деле протокол ТСП/ИР — это два разных протокола, определяющих различные аспекты передачи данных в сети:

- **протокол ТСП** (*Transmission Control Protocol*) — протокол управления передачей данных, использующий автоматическую повторную передачу пакетов, содержащих ошибки; этот протокол отвечает за разбиение передаваемой информации на пакеты и правильное восстановление информации из пакетов получателем;
- **протокол ИР** (*Internet Protocol*) — протокол межсетевого взаимодействия, отвечающий за адресацию и позволяющий пакету на пути к конечному пункту назначения проходить по нескольким сетям.

Схема передачи информации по протоколу ТСП/ИР такова: протокол ТСП разбивает информацию на пакеты и нумерует все пакеты; далее с помощью протокола ИР все пакеты передаются получателю, где с помощью протокола ТСП проверяется, все ли пакеты получены; после получения всех пакетов протокол ТСП располагает их в нужном порядке и собирает в единое целое.

Каждый компьютер, подключенный к сети Интернет имеет два равноценных уникальных адреса: цифровой IP-адрес и символический доменный адрес. Присваивание адресов происходит по следующей схеме: международная организация Сетевой информационный центр выдает группы адресов владельцам локальных сетей, а последние распределяют конкретные адреса по своему усмотрению.

IP-адрес компьютера имеет длину 4 байта. Обычно первый и второй байты определяют **адрес сети**, третий байт определяет **адрес подсети**, а четвертый — **адрес компьютера в подсети**. Для удобства IP-адрес записывают в виде четырех чисел со значениями от 0 до 255, разделенных точками, например: 145.37.5.150. Адрес сети — 145.37; адрес подсети — 5; адрес компьютера в подсети — 150.

Доменный адрес (англ. *domain* — область), в отличие от цифрового, является символическим и легче запоминается человеком. Пример доменного адреса: *barsuk.les.nora.ru*. Здесь домен *barsuk* — имя реального компьютера, обладающего IP-адресом, домен *les* — имя группы, присвоившей имя этому компьютеру, домен *nora* — имя более крупной группы, присвоившей имя домену *les*, и т.д. В процессе передачи данных доменный адрес преобразуются в IP-адрес.

Основные сервисы системы Интернет.

World Wide Web (WWW, “Всемирная паутина”) — основной инструмент Интернет, её главный информационный сервис.

World Wide Web (WWW, “Всемирная паутина”) — гипертекстовая, а точнее, гипермедийная информационная система поиска ресурсов Интернет и доступа к ним.

Гипертекст — информационная структура, позволяющая устанавливать смысловые связи между элементами текста на экране компьютера таким образом, чтобы можно было легко осуществлять переходы от одного элемента к другому. На практике в гипертексте некоторые слова выделяют путем подчёркивания или **окрашивания в другой цвет**. Выделение слова говорит о наличии связи этого слова с некоторым документом, в котором тема, связанная с выделенным словом, рассматривается более подробно.

Гипермедиа — это то, что получится, если в определении гипертекста заменить слово “*текст*” на “*любые виды информации*”: звук, графику, видео. Такие гипермедийные ссылки возможны, поскольку наряду с текстовой информацией можно связывать и любую другую двоичную информацию, например, закодированный звук или графику. Так, если программа отображает карту мира и если пользователь выбирает на этой карте с помощью мыши какой-либо континент, программа может тут же дать о нём графическую, звуковую и текстовую информацию.

Система WWW построена на специальном протоколе передачи данных, который называется **протоколом передачи гипертекста HTTP** (читается “*эйч-ти-ти-пи*”, *HyperText Transfer Protocol*).

Всё содержимое системы WWW состоит из **WWW-страниц**, называемых **сайтами** (англ. *site* — участок).

WWW-страницы (сайты) — гипермедийные документы системы *World Wide Web*. Создаются с помощью языка разметки гипертекста **HTML** (*Hypertext markup language*).

Язык HTML позволяет добавлять к текстовым документам специальные командные фрагменты — **тэги** (англ. *tag* — “этикетка, ярлык”) таким образом, что становится возможным связывать с этими документами другие тексты, графику, звук и видео, задавать заголовки различных уровней, разделять текст на абзацы, строить таблицы и т.д. Например, заголовок документа может иметь такой вид: **<TITLE> Клуб любителей персиков </TITLE>**

Одну WWW-страницу на самом деле обычно составляет **набор гипермедийных документов, расположенных на одном сервере, переплетённых взаимными ссылками и связанных по смыслу** (например, содержащих информацию об одном учебном заведении или об одном музее). Каждый документ страницы, в свою очередь, может содержать несколько экранных страниц текста и иллюстраций. Каждая WWW-страница имеет свой “**титульный лист**” (англ. “*homepage*”) — гипермедийный документ, содержащий ссылки на главные составные части страницы. Адреса “**титульных листов**” распространяются в Интернет в качестве адресов страниц.

Личные страницы — такие WWW-страницы, которые принадлежат не фирмам и не организациям, а отдельным людям. Содержание и оформление такой страницы зависит только от её автора.

При работе с системой WWW пользователи имеют дело с **программами-клиентами системы, называемыми браузерами**.

Браузеры (англ. *browse* — листать, просматривать) — программы, с помощью которых пользователь организует диалог с системой WWW: просматривает WWW страницы, взаимодействует с WWW-серверами и другими ресурсами в Интернет.

Существуют сотни программ-браузеров. Самые популярные браузеры: *Netscape Navigator* и *Microsoft Explorer*.

Браузеры WWW умеют взаимодействовать с любыми типами серверов, используя при этом их собственные протоколы. Информацию, полученную от любого сервера, браузер WWW выводит на экран в стандартной, удобной для восприятия форме. При этом переключения с одного протокола на другой для пользователя часто остаются незамеченными.

Программа удалённого доступа Telnet. Позволяет входить в другую вычислительную систему, работающую в Интернет, с помощью протокола **TELNET**. Эта программа состоит из двух компонент: *программы-клиента*, которая выполняется на *компьютере-клиенте*, и *программы-сервера*, которая выполняется на *компьютере-сервере*.

Функции **программы-клиента**:

- установление соединения с сервером;
- приём от абонента входных данных, преобразование их к стандартному формату и отсылка серверу;
- приём от сервера результатов запроса в стандартном формате и преформатирование их в вид, удобный клиенту.

Функции **программы-сервера**:

- ожидание запроса в стандартной форме;
- обслуживание этого запроса;
- отсылка результатов программе-клиенту.

Telnet — простое и поэтому универсальное средство связи в Интернет.

Программа пересылки файлов Ftp. Перемещает копии файлов с одного узла Интернет на другой в соответствии с протоколом **FTP** (*File Transfer Protocol* — “протокол передачи файлов”). При этом не имеет значения, где эти узлы расположены и как соединены между собой.

Компьютеры, на которых есть файлы для общего пользования, называются FTP-серверами. В Интернет имеется более 10 Терабайт бесплатных файлов и программ.

Электронная почта (*Electronic mail*, англ. *mail* — почта, сокр. *E-mail*, читается “и-мэйл”). Служит для **передачи текстовых сообщений в пределах Интернет, а также между другими сетями электронной почты**. К тексту письма современные почтовые программы позволяют прикреплять звуковые и графические файлы, а также двоичные файлы — программы.

При использовании электронной почты каждому абоненту присваивается уникальный почтовый адрес, формат которого имеет вид: *<имя пользователя> @ <имя почтового сервера>*. Например: *earth@space.com*, где *earth* — имя пользователя, *space.com* — имя компьютера, @ — разделительный символ “эт коммерческое”.

Сообщения, поступающие по *E-Mail*, хранятся в специальном “почтовом” компьютере в выделенной для получателя области дисковой памяти (его “почтовом ящике”), откуда он может их выгрузить и прочитать с помощью специальной программы-клиента.

Для отсылки сообщения нужно знать **электронный адрес абонента**. При качественной связи электронное письмо доходит в любую точку мира в течение нескольких минут.

Система телеконференций Usenet (от *Users Network*). Эта система организует **коллективные обсуждения** по различным направлениям, называемые **телеконференциями**. В каждой телеконференции проводится ряд **дискуссий** по конкретным темам.

Сегодня Usenet имеет более десяти тысяч дискуссионных групп (*NewsGroups*) или телеконференций, каждая из которых посвящена определённой теме и является средством обмена мнениями. Телеконференции разбиты на несколько групп:

- **news** — вопросы, касающиеся системы телеконференций;
- **comp** — компьютеры и программное обеспечение;
- **rec** — развлечения, хобби и искусства;
- **sci** — научно-исследовательская деятельность и приложения;
- **soc** — социальные вопросы;
- **talk** — дебаты по различным спорным вопросам;
- **misc** — всё остальное.

Внутри этих категорий существует **иерархия**. Так, например, *rec.music.beatles* — это дискуссия о творчестве Битлз, входящая в подгруппу “музыка” группы дискуссий по искусству.

Существует большой выбор **программ чтения телеконференций**, которые формируют материал дискуссий в упорядоченном виде и предоставляют в распоряжение корреспондентов.

Аналог телеконференций в других сетях — “электронная доска объявлений” (*Bulletin Board System, BBS*).

Системы информационного поиска сети Интернет

В Интернет представлена информация на любые темы, которые только можно себе представить. Но найти в ней нужную информацию не так-то легко из-за того, что сеть по своей природе не имеет чёткой структуры. Поэтому для **ориентировки в Интернет и быстрого получения свежей справочной информации разработаны системы поиска информации**.

Все системы поиска информации Интернет располагаются на специально выделенных компьютерах с мощными каналами связи. Ежеминутно они бесплатно обслуживают огромное количество клиентов.

Поисковые системы можно разбить на два типа:

- **предметные каталоги**, формируемые людьми-редакторами;
- **автоматические индексы**, формируемые специальными компьютерными программами, без участия людей.

Системы, основанные на предметных каталогах. Используют базы данных, формируемые специалистами-редакторами, которые отбирают информацию, устанавливают связи для баз данных, организуют и снабжают данные в разных поисковых категориях перекрёстными ссылками. Кампании, владеющие предметными каталогами, *непрерывно исследуют, описывают и каталогизируют содержимое WWW-серверов и других сетевых ресурсов, разбросанных по всему миру.* В результате этой работы клиенты Интернет имеют постоянно обновляющиеся иерархические (древовидные) каталоги, на верхнем уровне которых собраны самые общие категории, такие как “бизнес”, “наука”, “искусство” и т.п., а элементы самого нижнего уровня представляют собой ссылки на отдельные WWW-страницы и серверы вместе с кратким описанием их содержимого.

Пример. Если нужно выяснить, какая в мире имеется информация о *пище динозавров*, достаточно спуститься по иерархии:

Науки ==> Млекопитающие ==> Палеонтология ==> Динозавры ==> Пища.

Каталоги, составленные людьми, *более осмыслены*, чем автоматические индексы. Их очень мало, так как *их создание и поддержка требуют огромных затрат.* Для примера рассмотрим самый популярный предметный каталог **Yahoo!**, который обладает одной из крупнейших баз данных. Имеет информационные базы для детей и подростков. Поддерживает два основных метода работы с каталогом — *поиск по ключевым словам* и *поиск по иерархическому дереву разделов.* Не принимает запросов на естественном языке.

Автоматические индексы. Переоценить их трудно. Поиск по ключевым словам в одной базе данных, занимающий в худшем случае несколько секунд, принесёт те же результаты, что и обшаривание всех WWW-страниц во всей сети Интернет.

Автоматический индекс состоит из трёх частей:

- *программы-робота;*
- *базы данных,* собираемой этим роботом;
- *интерфейса* для поиска в этой базе, с которым и работает пользователь.

Все эти компоненты функционируют *без вмешательства человека.*

К автоматическим индексам следует прибегать только тогда, когда ключевые слова точно известны, например, фамилия человека или несколько специфических терминов из соответствующей области. Индексы получают информацию из каждого отдельного узла, регистрируют и индексируют её и добавляют к своим базам данных.



Среди известных индексов выделяется: *AltaVista* — одна из самых мощных полностью автоматических поисковых систем. Обладает полнотекстовой базой данных. Выдаёт наибольшее количество ссылок. Проиндексировано 30 млн. страниц с 300 тысяч серверов и 4 млн. статей из телеконференций Usenet. За один день *AltaVista* обслуживает около 20 млн. запросов.

В Интернет один и тот же узел сети может одновременно работать по нескольким протоколам. Поэтому крупные узлы сети сейчас обладают полным набором серверов, и к ним можно обращаться почти по любому из существующих протоколов.

**Режимы компьютерной обработки данных в
экономической деятельности.**

В настоящее время широко используются пакетный и диалоговый режимы обработки данных, причем последний не является альтернативой первого, а может рассматриваться скорее как его развитие. Выбор того или иного режима вытекает из особенностей каждого из них и особенностей решаемой задачи. Характеризуя пакетный режим обработки данных, необходимо отметить следующие его характерные черты. Ввод потока заданий осуществляется с локальных устройств ввода. Выполнение режима включает три фазы обработки : подготовку, выполнение и завершение процесса. При этом первая фаза требует определения последовательности действий и ввода исходных данных. Вторая фаза предполагает логическое преобразование исходных файлов, создания и упорядочения рабочих файлов, обработку информации и формирование выходных данных, осуществляя контроль результатов решения. На завершающей фазе выполняется печать. Эти особенности необходимо рассмотреть в связи со спецификой функциональной задачи. Применение пакетного режима позволяет уменьшить вмешательство оператора в процесс решения задачи, требует только предварительного ввода данных, исключает возможность вмешательства пользователя и, таким образом, изменения последовательности выполняемых действий. Однако, за счет этого появляется более полная загрузка оборудования, которое начинает работать по жесткому графику. В некоторых случаях для решения задачи выполняется и параллельная обработка данных. Пакетный режим более тесно связан с бумажной технологией.

Диалоговый режим, напротив, предполагает активное вмешательство пользователя в процесс работы комплекса и ориентацию на безбумажную технологию. В ходе его выполнения отсутствует заранее установленная последовательность операций обработки данных и дополнительного их ввода. В процессе решения задачи удобство диалогового режима в полной мере проявляется в процессе общения с базой данных. Среди них можно отметить такие как : - возможность перебора различных комбинаций поисковых признаков в запросе; - обеспечение более быстрого поиска данных; - улучшение характеристик выходных данных за счет оперативной коррекции запроса с терминала; - возможность расширения, сужения или изменения направлений поиска сразу после получения результатов; - множественность точек доступа; - быстрый доступ к относительно редко используемой информации; - оперативный анализ получаемых сведений. Приближение пользователя к процессу обработки данных повлекло за собой много проблем и одна из них - это проблема диалога конечного пользователя и ЭВМ. В настоящее время эта проблема решается в двух альтернативных направлениях : создание меню-ориентированных систем и систем, основанных на использовании языков, близких к естественному. Поэтому при обосновании выбора диалогового режима необходимо остановиться и на этом вопросе. Меню-ориентированные системы применяются тогда, когда число переборов вариантов расчетов относительно невелико. Обычно в меню с пятиуровневой иерархией уже

наступает комбинаторный взрыв. При необходимости повышения гибкости диалога более удобен язык близкий к естественному, однако, реализация его всегда сложна.

Защита информации

Надежная защита информации в разрабатываемых и функционирующих системах обработки данных может быть эффективной, если она будет надежной на всех объектах и во всех элементах системы, которые могут быть подвергнуты угрозам. В связи с этим для создания средств защиты важно определить природу угроз, формы и пути их возможного проявления и осуществления, перечень объектов и элементов, которые, с одной стороны, могут быть подвергнуты (косвенно или непосредственно) угрозам с целью нарушения защищенности информации, а с другой - могут быть достаточно четко локализованы для организации эффективной защиты информации.

При проведении исследовательских работ в этом направлении необходимо четко разграничить два класса рисков : один - для автономных ПК и автономных компьютерных систем; другой - для систем, имеющих выход в большие сети, включая Интернет.

В специальной литературе под *объектом защиты* понимается такой структурный компонент системы, в котором находится или может находиться подлежащая защите информация, а под *элементом защиты* - совокупность данных, которая может содержать подлежащие защите сведения.

Практика показывает, что информация в процессе ввода, хранения, обработки, вывода и передачи подвергается различным случайным воздействиям, в результате которых на аппаратном уровне происходят физические изменения в сигнальных формах представления информации. Если в каком-то или в каких-то разрядах цифрового кода, несущего информацию, произошло инвертирование двоичного знака (с 1 на 0 или наоборот) и оно не обнаружено специальными аппаратными средствами функционального контроля, то при дальнейшей обработке информации либо будет получен неверный результат, либо сообщение направится по ложному адресу, либо произойдут другие нежелательные события (разрушение, модификация, утечка информации и др.).

На программном уровне в результате случайных воздействий может произойти изменение алгоритма обработки информации на непредусмотренный и, как следствие

этого, - прекращение или модификация процесса, в результате которого опять же возможны разрушение или утечка информации (при перепутывании, например, адресата).

Причинами случайных воздействий при функционировании компьютерных систем могут быть:

- отказы и сбои аппаратуры в случае ее некачественного исполнения и физического старения;
- помехи в каналах и на линиях связи от воздействия внешней среды;
- аварийные ситуации (пожар, наводнение, выход из строя электропитания и др.);
- схемные и системотехнические ошибки и просчеты разработчиков и производителей ПК;
- алгоритмические и программные ошибки;
- ошибки человека при работе с ПК.

Злоумышленные или преднамеренные угрозы - результат активного воздействия человека на объекты и процессы по самым различным причинам (материальный интерес, желание навредить, развлечение с самоутверждением своих способностей и др.).

В качестве *объектов защиты* информации в системах обработки данных можно выделить следующие:

- терминалы пользователей (персональные компьютеры, рабочие станции сети);
 - терминал администратора сети или групповой абонентский узел;
 - узел связи;
 - средства отображения информации;
 - средства документирования информации;
- машинный зал (компьютерный или дисплейный) и хранилище носителей информации;
- внешние каналы связи и сетевое оборудование;
 - накопители и носители информации.

В соответствии с приведенным выше определением в качестве *элементов защиты* выступают блоки (порции, массивы, потоки и др.) информации в объектах защиты, в частности:

- данные и программы в основной памяти компьютера;
- данные и программы на внешнем машинном носителе (гибком и жестком дисках);
- данные, отображаемые на экране монитора;
- данные, выводимые на принтер при автономном и сетевом использовании ПК;
 - пакеты данных, передаваемые по каналам связи;
- данные, размножаемые (тиражируемые) с помощью копировально-множительного оборудования;
- отходы обработки информации в виде бумажных и магнитных носителей;
- журналы назначения паролей и приоритетов зарегистрированным пользователям;
- служебные инструкции по работе с комплексами задач;
- архивы данных и программного обеспечения и др.

Доступ к объектам и элементам защиты информации теоретически и практически возможен для двух категорий лиц: законных пользователей и нарушителей.

При отсутствии на рабочем месте законного пользователя или при его халатном отношении к своим должностным обязанностям, при недостаточной защите информации квалифицированный нарушитель может осуществить путем ввода соответствующих запросов (команд) несанкционированный доступ к информации. При достаточно свободном доступе в помещение, где размещены средства ВТ, можно визуально наблюдать информацию на средствах отображения и документирования, а также похитить носители с информацией (дискеты, ленты, листинги и др.) либо снять с них копию. При бесконтрольной загрузке в компьютер программы нарушитель может модифицировать данные и алгоритмы, ввести вредоносную программу типа «троянский конь», с помощью которой впоследствии он может реализовывать нужные для себя функции.

Особо опасна ситуация, когда нарушителем является пользователь компьютерной системы, имеющий согласно своим функциональным обязанностям законный доступ к одной части информации, но обращающийся к другой за пределами своих полномочий.

Несанкционированный доступ к информации может происходить во время технического обслуживания (профилактики или ремонта) компьютеров за счет прочтения

информации на машинных и других носителях, несмотря на ее удаление (стирание) пользователем обычными методами. Другой способ - прочтение информации с носителя во время его транспортировки без охраны внутри объекта или региона.

Современные средства вычислительной техники базируются на широком применении интегральных схем. При работе таких схем происходят высокочастотные изменения уровней напряжения и токов, а это, в свою очередь, приводит к возникновению в цепях питания, в эфире, в близко расположенной аппаратуре и т.п. различных электромагнитных полей и наводок, которые с помощью специальных средств (условно назовем их «шпионскими») можно трансформировать в обрабатываемую информацию. Причем с уменьшением расстояния между приемником нарушителя и аппаратными средствами вероятность такого рода съема и расшифровки информации увеличивается.

Несанкционированное ознакомление с информацией возможно путем непосредственного подключения нарушителем «шпионских» средств к каналам связи и сетевым аппаратным средствам.

Несанкционированное ознакомление с информацией подразделяется на *пассивное* и *активное*. В первом случае не происходит нарушения информационных ресурсов, и нарушитель лишь получает возможность раскрывать содержание сообщений, используя это в дальнейшем в своих корыстных целях. Во втором случае нарушитель может выборочно изменить, уничтожить, переупорядочить и перенаправить сообщения, задержать и создать поддельные сообщения и др.

Для обеспечения безопасности информации в личных компьютерах и особенно в офисных системах и компьютерных сетях проводятся различные мероприятия, объединяемые понятием «система защиты информации».

Система защиты информации - это совокупность организационных (административных) и технологических мер, программно-технических средств, правовых и морально-этических норм,

направленных на противодействие угрозам нарушителей с целью сведения до минимума возможного ущерба пользователям и владельцам системы.

На практике при построении системы защиты информации сложились два подхода: фрагментарный и комплексный. В первом случае мероприятия по защите направляются на противодействие вполне определенным угрозам при строго определенных условиях, например, обязательная проверка носителей антивирусными программами,

применение криптографических систем шифрования и т.д. При комплексном подходе различные меры противодействия угрозам объединяются, формируя так называемую *архитектуру безопасности систем*.

Исследования практики функционирования систем обработки данных и компьютерных сетей показали, что существует достаточно много возможных направлений утечки информации и путей несанкционированного доступа к ней в системах и сетях:

- перехват электронных излучений;
 - принудительно электромагнитное облучение (подсветка) линий связи;
 - применение «подслушивающих» устройств;
 - дистанционное фотографирование;
 - перехват акустических волновых излучений;
 - хищение носителей информации и производственных отходов систем обработки данных;
 - считывание информации из массивов других пользователей;
 - чтение остаточной информации в аппаратных средствах;
 - копирование носителей информации и файлов с преодолением мер защиты;
 - модификация программного обеспечения путем исключения или добавления новых функций;
 - использование недостатков операционных систем и прикладных программных средств;
 - незаконное подключение к аппаратуре и линиям связи, в том числе в качестве активного ретранслятора;
 - злоумышленный вывод из строя механизмов защиты;
 - маскировка под зарегистрированного пользователя и присвоение себе его полномочий;
 - введение новых пользователей;
- внедрение компьютерных вирусов.

Кроме того, система защиты не должна допускать, чтобы:

- злоумышленник мог снять с себя ответственность за формирование ложной или разрушающей информации;

- были отказы от фактов получения информации, которая фактически была получена, но в другое время;

- подтверждались сообщения о посылке кому-то информации, которая на самом деле не посылалась;

- в передаваемой информации содержалась другая (вредоносная) информация;

- в число пользователей попадали без регистрации новые лица, чтобы не удалялись и не модифицировались действующие лица;

- отдельные пользователи незаконно расширяли свои полномочия по доступу к информации и процессам ее обработки;

- создавались помехи обмену сообщениями между пользователями с целью нарушения и искажения передаваемой информации.

Учитывая важность, масштабность и сложность решения проблемы сохранности и безопасности информации, рекомендуется разрабатывать архитектуру безопасности в несколько этапов:

- анализ возможных угроз;
- разработка системы защиты;
- реализация системы защиты;
- сопровождение системы защиты.

Следует, однако, заметить, что на конкретном объекте и в конкретной системе обработки данных из всего многообразия угроз и возможных воздействий следует в первую очередь выбрать наиболее вероятные, а также те, которые могут нанести наиболее серьезный ущерб.

Этап разработки системы защиты информации предусматривает использование различных комплексов мер и мероприятий организационно-административного, технического, программно-аппаратного, технологического, правового, морально-этического характера и др.

Организационно-административные средства защиты сводятся к регламентации доступа к информационным и вычислительным ресурсам, функциональным процессам систем обработки данных, к регламентации деятельности персонала и др. Их цель - в наибольшей степени затруднить или исключить возможность реализации угроз безопасности. Наиболее типичные организационно-административные средства:

- создание контрольно-пропускного режима на территории, где располагаются средства обработки информации;
- изготовление и выдача специальных пропусков;

- мероприятия по подбору персонала, связанного с обработкой данных;
- допуск к обработке и передаче конфиденциальной информации только проверенных должностных лиц;
- хранение магнитных и иных носителей информации, представляющих определенную тайну, а также регистрационных журналов в сейфах, не доступных для посторонних лиц;
- организация защиты от установки прослушивающей аппаратуры в помещениях, связанных с обработкой информации;
- организация учета использования и уничтожения документов (носителей) с конфиденциальной информацией;
- разработка должностных инструкций и правил по работе с компьютерными средствами и информационными массивами;
- разграничение доступа к информационным и вычислительным ресурсам должностных лиц в соответствии с их функциональными обязанностями.

Технические средства защиты призваны создать некоторую физически замкнутую среду вокруг объекта и элементов защиты. В этом случае используются такие мероприятия:

- установка средств физической преграды защитного контура помещений, где ведется обработка информации (кодовые замки; охранная сигнализация - звуковая, световая, визуальная без записи и с записью на видеопленку);
 - ограничение электромагнитного излучения путем экранирования помещений, где происходит обработка информации, листами из металла или специальной пластмассы;
 - осуществление электропитания оборудования, обрабатывающего ценную информацию, от автономного источника питания или от общей электросети через специальные сетевые фильтры;
- применение, во избежание несанкционированного дистанционного съема информации, жидкокристаллических или плазменных дисплеев, струйных или лазерных принтеров соответственно с низким электромагнитным и акустическим излучением;
- использование автономных средств защиты аппаратуры в виде кожухов, крышек, дверец, шторок с установкой средств контроля вскрытия аппаратуры.

Программные средства и методы защиты активнее и шире других применяются для защиты информации в персональных компьютерах и компьютерных сетях, реализуя такие функции защиты, как разграничение и контроль доступа к ресурсам; регистрация и анализ протекающих процессов, событий, пользователей; предотвращение возможных разрушительных воздействий на ресурсы; криптографическая защита информации; идентификация и аутентификация пользователей и процессов и др.

В настоящее время наибольший удельный вес в этой группе мер в системах обработки экономической информации составляют специальные пакеты программ или отдельные программы, включаемые в состав программного обеспечения с целью реализации задач по защите информации.

Технологические средства защиты информации - это комплекс мероприятий, органично встраиваемых в технологические процессы преобразования данных. Среди них:

- создание архивных копий носителей;
- ручное или автоматическое сохранение обрабатываемых файлов во внешней памяти компьютера;
- регистрация пользователей компьютерных средств в журналах;
- автоматическая регистрация доступа пользователей к тем или иным ресурсам;
- разработка специальных инструкций по выполнению всех технологических процедур и др.

К *правовым и морально-этическим мерам и средствам защиты* относятся действующие в стране законы, нормативные акты, регламентирующие правила обращения с информацией и ответственность за их нарушение; нормы поведения, соблюдение которых способствует защите информации.

Примером действующих законодательных актов в Российской Федерации, которыми регламентированы цивилизованные юридические и моральные отношения в сфере информационного рынка, являются законы РФ «Об информации, информатизации и защите информации» от 20.02.1995 г. № 24-ФЗ; «О правовой охране программ для ЭВМ и баз данных» № 5351 -4 от 9.07.1993 г. в редакции Федерального закона от 19.07.95 № 110-ФЗ и др.; примером предписаний морально-этического характера - «Кодекс профессионального поведения членов Ассоциации пользователей ЭВМ США».

Закон РФ «Об информации, информатизации и защите информации» от 20.02.1995 г. создает условия для включения России в международный информацион-

ный обмен, предотвращает бесхозяйственное отношение к информационным ресурсам и информатизации, обеспечивает информационную безопасность и права юридических и физических лиц на информацию. Заложив юридические основы гарантий прав граждан на информацию, закон направлен на обеспечение защиты собственности в сфере информационных систем и технологий, формирование рынка информационных ресурсов, услуг, систем, технологий, средств их обеспечения. Все 25 статей Закона РФ «Об информации, информатизации и защите информации» сгруппированы в главы: общие положения; информационные ресурсы; пользование информационными ресурсами; информатизация; информационные системы, технологии и средства их обеспечения; защита информации и прав субъектов в области информационных процессов и информатизации.

Следует заметить, что в действующем ныне Уголовном кодексе РФ имеется глава «Преступления в сфере компьютерной информации». В ней содержатся три статьи: «Неправомерный доступ к компьютерной информации» (ст. 272), «Создание, использование и распространение вредоносных программ для ЭВМ» (ст. 273) и «Нарушение прав эксплуатации ЭВМ, систем ЭВМ или их сетей» (ст. 274).

В зависимости от серьезности последствий компьютерного злоупотребления к лицам, его совершившим, могут применяться различные меры наказания, вплоть до лишения свободы сроком до 5 лет.

СРЕДСТВА ОПОЗНАНИЯ И РАЗГРАНИЧЕНИЯ ДОСТУПА К ИНФОРМАЦИИ

В компьютерных системах сосредотачивается информация, право на пользование которой принадлежит определенным лицам или группам лиц, действующим в порядке личной инициативы или в соответствии с должностными обязанностями. Чтобы обеспечить безопасность информационных ресурсов, устранить

возможность несанкционированного доступа, усилить контроль санкционированного доступа к конфиденциальной либо к подлежащей засекречиванию информации, внедряются различные системы опознавания, установления подлинности объекта (субъекта) и разграничения доступа. В основу построения таких систем закладывается принцип допуска и выполнения только таких обращений к информации, в которых присутствуют соответствующие признаки разрешенных полномочий.

Ключевыми понятиями в этой системе являются "идентификация" и "аутентификация". *Идентификация* - это присвоение какому-либо объекту или субъекту уникального имени или образа. *Аутентификация* - это установление подлинности, т.е. проверка, является ли объект (субъект) действительно тем, за кого он себя выдает.

Конечная цель процедур идентификации и аутентификации объекта (субъекта) - допуск его к информации ограниченного пользования в случае положительной проверки либо отказ в допуске в случае отрицательного исхода проверки.

Объектами идентификации и аутентификации могут быть: люди (пользователи, операторы и др.); технические средства (мониторы, рабочие станции, абонентские пункты); документы (ручные, распечатки и др.); магнитные носители информации; информация на экране монитора, табло и др.

Установление подлинности объекта может производиться аппаратным устройством, программой, человеком и т.д. При этом для защиты информации в компьютерных системах должна обеспечиваться конфиденциальность образов и имен образов.

При обмене информацией между человеком и компьютером (либо только между компьютерами в сети) должна быть предусмотрена взаимопроверка подлинности взаимодействующих объектов. Необходимо, чтобы каждый из объектов (субъектов) хранил в своей памяти, не доступной для посторонних, список имен (образов) объектов (субъектов), с которыми производится обмен информацией.

Один из наиболее распространенных методов аутентификации - присвоение лицу или другому имени пароля и хранение его значения в вычислительной системе. *Пароль* - это совокупность символов, определяющая объект (субъект). При выборе пароля возникают вопросы о его размере, стойкости к несанкционированному подбору, способам его применения. Естественно, чем больше длина пароля, тем большую безопасность будет обеспечивать система, ибо потребуются большие усилия для его отгадывания. При этом выбор длины пароля в значительной степени определяется развитием технических средств, их элементной базой и быстродействием. К примеру, если пользователь в качестве своего пароля использует четырехразрядное десятичное число, то компьютерная программа, выполнив перебор чисел от 0000 до 9999 (9999 комбинаций) по так называемому «лобовому методу», сумеет распознать код.

Четырехзначный пароль, в котором используются цифровые символы и 26 букв латинского алфавита (то есть всего 36 возможных знаков), требует более трудоемкого

процесса распознавания, ибо он допускает 36^4 , или 1 679 616, уникальных комбинаций; при пятизначной длине пароля число комбинаций возрастает до $36^5 = 60\,466\,176$. Увеличивая длину пароля и число используемых символов, можно увеличить число возможных комбинаций, повышая время на лобовой взлом пароля.

Сейчас широко применяются многосимвольные пароли с разрядностью более 10 знаков.

Наиболее высокий уровень безопасности достигается в случае деления пароля на две части: одну 3-6-значную, легко запоминаемую человеком, и вторую, содержащую количество знаков, определяемое требованиями к защите и возможностями технической реализации системы. Эта часть помещается на специальный физический носитель - карточку, устанавливаемую пользователем в специальное считывающее устройство.

В случае применения пароля как средства аутентификации необходимо заменять его на новый не реже одного раза в год, чтобы снизить вероятность его перехвата путем прямого хищения носителя, снятия его копии и даже физического принуждения человека. Пароль вводится пользователем в начале взаимодействия с компьютерной системой, иногда и в конце сеанса (в особо ответственных случаях пароль нормального выхода может отличаться от входного). Для правомочности пользователя может предусматриваться ввод пароля через определенные промежутки времени.

Пароль как средство обеспечения безопасности может использоваться для идентификации и установления подлинности терминала, с которого входит в систему пользователь, а также для

обратного установления подлинности компьютера по отношению к пользователю.

Учитывая важность пароля как средства повышения безопасности информации от несанкционированного использования, следует соблюдать некоторые меры предосторожности, в том числе:

- не хранить пароли в вычислительной системе в незашифрованном виде;
- не печатать и не отображать пароли в явном виде на терминале пользователя;
- не использовать в качестве пароля свое имя или имена родственников, а также личную информацию (дата рождения, номер домашнего или служебного телефона, название улицы и др.);

- не использовать реальные слова из энциклопедии или толкового словаря;
- выбирать длинные пароли;
- использовать смесь символов верхнего и нижнего регистров клавиатуры;
- использовать комбинации из двух простых слов, соединенных специальными символами (например, +, = и др.);
- придумывать новые слова (абсурдные или даже бредового содержания);
- чаще менять пароль.

Для идентификации пользователей могут применяться сложные в плане технической реализации системы, обеспечивающие установление подлинности пользователя на основе анализа его индивидуальных параметров: отпечатков пальцев, рисунка линий руки, радужной оболочки глаз, тембра голоса и др. Но пока эти приемы носят скорее рекламный, чем практический характер.

Более широкое распространение нашли физические методы идентификации с использованием носителей кодов паролей. Такими носителями являются пропуска в контрольно-пропускных системах; пластиковые карты с именем владельца, его кодом, подписью; пластиковые карточки с магнитной полосой, содержащей около 100 байт информации, которая считывается специальным считывающим устройством (используются как кредитные карточки, карточки для банкоматов и др.); пластиковые карты, содержащие встроенную микросхему (smart-card); карты оптической памяти и др.

Одно из интенсивно разрабатываемых направлений по обеспечению безопасности информации - идентификация и установление подлинности документов на основе *электронной цифровой подписи* - ныне простирается от проведения финансовых и банковских операций до контроля за выполнением различных договоров. Естественно, при передаче документов по каналам связи применяется факсимильная аппаратура, но в этом случае к получателю приходит не подлинник, а лишь копия документа с копией подписи, которая в процессе передачи может быть подвергнута повторному копированию для использования ложного документа.

Электронная цифровая подпись представляет собой способ шифрования с помощью криптографического преобразования и является паролем, зависящим от отправителя, получателя и содержания передаваемого сообщения. Для предупреждения повторного использования подпись должна меняться от сообщения к сообщению.

КРИПТОГРАФИЧЕСКИЙ МЕТОД ЗАЩИТЫ ИНФОРМАЦИИ

Криптографическое преобразование - один из наиболее эффективных методов, резко повышающих безопасность:

- передачи данных в компьютерных сетях;
- данных, хранящихся в удаленных устройствах памяти;
- информации при обмене между удаленными объектами.

Криптография известна с древнейших времен, однако она всегда оставалась привилегией правительственных и военных учреждений. Изменение ситуации связывается с публикацией в 1949 г. книги К. Шеннона по теории информации и кибернетике, когда к криптографическим методам преобразования информации обратились многие ученые, банковские и коммерческие системы.

Защита информации методом криптографического преобразования заключается в приведении ее к неявному виду путем преобразования составных частей информации (букв, цифр, слогов, слов) с помощью специальных алгоритмов либо аппаратных средств и кодов ключей. *Ключ* - это изменяемая часть криптографической системы, хранящаяся в тайне и определяющая, какое шифрующее преобразование из возможных выполняется в данном случае.

Для преобразования (шифрования) используется некоторый алгоритм или устройство, реализующее заданный алгоритм, которые могут быть известны широкому кругу лиц. Само же управление процессом шифрования осуществляется с помощью периодически меняющегося кода ключа, обеспечивающего каждый раз оригинальное представление информации при использовании одного и того же алгоритма или устройства. Знание ключа позволяет относительно быстро, просто и надежно расшифровать текст. Однако без знания ключа эта процедура может оказаться практически невыполнимой даже при использовании компьютера.

К методам криптографического преобразования применимы следующие требования:

- метод должен быть достаточно устойчивым к попыткам раскрытия исходного текста на основе зашифрованного;
- обмен ключа не должен быть труден для запоминания;
- затраты на защитные преобразования должны быть приемлемы при заданном уровне сохранности информации;
- ошибки в шифровании не должны приводить к явной потере информации;

- длина зашифрованного текста не должна превышать длину исходного текста.

Существует несколько методов защитных преобразований, которые можно подразделить на четыре основные группы: *перестановки*, *замены (подстановки)*, *аддитивные* и *комбинированные методы*.

Для методов *перестановки* и *замены (подстановки)* характерна короткая длина ключа, а надежность защиты определяется сложностью алгоритмов преобразования, и, наоборот, для аддитивных методов характерны простые алгоритмы и длинные ключи.

Названные четыре метода криптографического преобразования относятся к методам симметричного шифрования, т.е. один и тот же ключ используется и для шифрования, и для дешифрования. Однако в последние годы учеными разработан метод несимметричного шифрования, при котором для шифрования применяется один ключ, называемый открытым, а для дешифрования другой - закрытый.

Основными методами криптографического преобразования считаются методы *перестановки* и *замены*. Суть первого метода заключается в разбиении исходного текста на блоки, а затем в записи этих блоков и чтении зашифрованного текста по разным путям геометрической фигуры, например, запись исходного текста - по строкам матрицы, а чтение - по ее столбцам.

Шифрование методом замены заключается в том, что символы исходного текста (блока), записанные в одном алфавите, заменяются символами другого алфавита в соответствии с принятым ключом преобразования.

Комбинация этих методов породила так называемый *производный шифр*, обладающий сильными, криптографическими возможностями. Этот *комбинированный метод* принят в США в качестве стандарта для шифрования данных, а также в отечественном ГОСТе 28147-89. Алгоритм метода реализуется как аппаратно, так и программно, но базовый алгоритм рассчитан на реализацию с помощью электронных устройств специального назначения, что обеспечивает высокую производительность и упрощенную организацию обработки информации. Налаженное в ряде стран Запада промышленное производство аппаратуры для криптографического шифрования позволяет резко повысить безопасность коммерческой информации при ее хранении и электронном обмене в компьютерных системах.

КОМПЬЮТЕРНЫЕ ВИРУСЫ И АНТИВИРУСНЫЕ ПРОГРАММНЫЕ СРЕДСТВА

Массовое использование ПК в сетевом режиме, включая выход в глобальную сеть Интернет, породило проблему заражения их компьютерными вирусами.

Компьютерным вирусом принято называть специально написанную, обычно небольшую по размерам программу, способную самопроизвольно присоединяться к другим программам (т.е. заражать их), создавать свои копии (не обязательно полностью совпадающие с оригиналом) и внедрять их в файлы, системные области компьютера и в другие объединенные с ним компьютеры с целью нарушения нормальной работы программ, порчи файлов и каталогов, создания различных помех при работе на компьютере. Способ функционирования большинства вирусов - это такое изменение системных файлов компьютера, чтобы вирус начинал свою деятельность при каждой загрузке. Некоторые вирусы инфицируют файлы загрузки системы, другие специализируются на EXE-, COM- и других программных файлах. Всякий раз, когда пользователь копирует файлы на гибкий диск или посылает инфицированные файлы по модему, переданная копия вируса пытается установить себя на новый диск.

Обычно вирус разрабатывается так, чтобы он появился, когда происходит некоторое событие вызова, например, пятница 13-е, другая дата, определенное число перезагрузок зараженного или какого-то конкретного приложения, процент заполнения жесткого диска и др.

После того как вирус выполнит нужные ему действия, он передает управление той программе, в которой он находится, и ее работа некоторое время не отличается от работы незараженной. Все действия вируса могут выполняться достаточно быстро и без выдачи каких-либо сообщений, поэтому пользователь часто и не замечает, что компьютер работает со «странностями». К признакам появления вируса можно отнести:

- замедление работы компьютера;
- невозможность загрузки операционной системы;
- частые «зависания» и сбои в работе компьютера;
- прекращение работы или неправильную работу ранее успешно функционировавших программ;
- увеличение количества файлов на диске;
- изменение размеров файлов;
- периодическое появление на экране монитора неуместных сообщений;
- уменьшение объема свободной оперативной памяти;

- заметное возрастание времени доступа к жесткому диску;
- изменение даты и времени создания файлов;
- разрушение файловой структуры (исчезновение файлов, искажение каталогов и др.);
- загорание сигнальной лампочки дисковода, когда к нему нет обращения, и др.

Надо заметить, что названные симптомы необязательно вызываются компьютерными вирусами, они могут быть следствием других причин, поэтому компьютер следует периодически диагностировать.

Во многих странах действуют законодательные меры по борьбе с компьютерными преступлениями и злоумышленными действиями, разрабатываются антивирусные программные средства, однако количество новых программных вирусов возрастает. Лица, которые используют свои знания и опыт для несанкционированного доступа к информационным и вычислительным ресурсам, к получению конфиденциальной и секретной информации, к совершению вредоносных действий, называют *хакерами*.

Деятельность хакеров зачастую бывает социально опасной. В июне 1987 г. спецслужбами ФРГ был арестован некто М. Шпеер, «взломавший» компьютерную систему военной базы в штате Алабама, хранившую сведения о боеготовности ракет большой дальности, информационную сеть ЦРУ, банк данных Пентагона и других государственных учреждений. В ноябре 1999 г. студент Корнеллского университета (США) запустил компьютерную *программу-червь* в Интернет. В течение нескольких часов программа заразила около 6000 компьютеров, в том числе военной сети Министерства обороны США, работавших под управлением операционной системы Unix.

В мае 2000 г. вирус под названием "I love you" (Я тебя люблю), распространенный по электронной почте, поразил сотни тысяч персональных компьютеров в США и странах Европы. Это, по оценкам специалистов, принесло в первые же дни эпидемии до 1 млрд долл. убытков фирмам и неприятности простым пользователям. Зараженными оказались даже компьютерные системы британского парламента и американского конгресса.

По более детализированной схеме классификации компьютерных злоумышленников делят на хакеров (hacker), кракеров (cracker) и фрикеров (phracer).

Действия хакеров, или компьютерных хулиганов, могут наносить существенный вред владельцам компьютеров и владельцам (создателям) информационных ресурсов,

так как приводят к простоям компьютеров, необходимости восстановления испорченных данных либо к дискредитации юридических или физических лиц, например, путем искажения информации на электронных досках объявлений или на WEB-серверах в Интернете. Мотивы действий компьютерных злоумышленников самые различные: стремление к финансовым приобретениям; желание навредить и отомстить руководству организации, из которой по тем или иным причинам уволился сотрудник; психологические черты человека (зависть,

тщеславие, желание проявить свое техническое превосходство над другими, просто хулиганство и пр.).

Основными путями заражения компьютеров вирусами являются съемные диски (дискеты и CD-ROM) и компьютерные сети. Заражение жесткого диска компьютера может произойти при загрузке компьютера с дискеты, содержащей вирус. Для усиления безопасности необходимо обращать внимание на то, как и откуда получена программа (из сомнительного источника, имеется ли наличие сертификата, эксплуатировалась ли раньше и т.д.). Однако главная причина заражения компьютеров вирусами - отсутствие в операционных системах эффективных средств защиты информации от несанкционированного доступа.

По данным специальной литературы, к концу 1998 г. в мировой практике было зарегистрировано более 20 тыс. компьютерных вирусов, и каждую неделю появляется около десяти новых вирусов. Одна из схем классификации компьютерных вирусов представлена на рис. 11.2.

В зависимости от среды обитания вирусы классифицируются на загрузочные, файловые, системные, сетевые, файлово-загрузочные.

Загрузочные вирусы внедряются в загрузочный сектор диска или в сектор, содержащий программу загрузки системного диска.

Файловые вирусы внедряются в основном в исполняемые файлы с расширением .COM и .EXE.

Системные вирусы проникают в системные модули и драйверы периферийных устройств, таблицы размещения файлов и таблицы разделов.

Сетевые вирусы обитают в компьютерных сетях; *файлово-загрузочные* (многофункциональные) поражают загрузочные секторы дисков и файлы прикладных программ.

По способу заражения среды обитания вирусы подразделяются на резидентные и на нерезидентные.

Резидентные вирусы при заражении компьютера оставляют в оперативной памяти свою резидентную часть, которая затем перехватывает обращение операционной системы к другим объектам заражения, внедряется в них и выполняет свои разрушительные действия вплоть до выключения или перезагрузки компьютера. *Нерезидентные вирусы* не заражают оперативную память ПК и являются активными ограниченное время.

Алгоритмическая особенность построения вирусов оказывает влияние на их проявление и функционирование. Так, реплика-

торные программы благодаря своему быстрому воспроизводству приводят к переполнению основной памяти, при этом уничтожение программ-репликаторов усложняется, если воспроизводимые программы не являются точными копиями оригинала. В компьютерных сетях распространены *программы-черви*. Они вычисляют адреса сетевых компьютеров и рассылают по этим адресам свои копии, поддерживая между собой связь. В случае прекращения существования «червя» на каком-либо ПК оставшиеся отыскивают свободный компьютер и внедряют в него такую же программу.

«*Троянский конь*» - это программа, которая, маскируясь под полезную программу, выполняет дополнительные функции, о чем пользователь и не догадывается (например, собирает информа-

цию об именах и паролях, записывая их в специальный файл, доступный лишь создателю данного вируса), либо разрушает файловую систему.

Логическая бомба - это программа, которая встраивается в большой программный комплекс. Она безвредна до наступления определенного события, после которого реализуется ее логический механизм. Например, такая вирусная программа начинает работать после некоторого числа прикладной программы, комплекса, при наличии или отсутствии определенного файла или записи файла и т.д.

Программы-мутанты, самовоспроизводясь, воссоздают копии, которые явно отличаются от оригинала.

Вирусы-невидимки, или стелс-вирусы, перехватывают обращения операционной системы к пораженным файлам и секторам дисков и подставляют вместо себя незараженные объекты. Такие вирусы при обращении к файлам используют достаточно ори-

гинальные алгоритмы, позволяющие «обманывать» резидентные антивирусные мониторы.

Макровирусы используют возможности макроязыков, встроенных в офисные программы обработки данных (текстовые редакторы, электронные таблицы и т.д.).

По степени воздействия на ресурсы компьютерных систем и сетей, или по деструктивным возможностям, выделяются безвредные, неопасные, опасные и разрушительные вирусы.

Безвредные вирусы не оказывают разрушительного влияния на работу ПК, но могут переполнять оперативную память в результате своего размножения.

Неопасные вирусы не разрушают файлы, но уменьшают свободную дисковую память, выводят на экран графические эффекты, создают звуковые эффекты и т.д. *Опасные вирусы* нередко приводят к различным серьезным нарушениям в работе компьютера; *разрушительные* - к стиранию информации, полному или частичному нарушению работы прикладных программ. Необходимо иметь в виду, что любой файл, способный к загрузке и выполнению кода программы, является потенциальным местом, куда может внедриться вирус.

Массовое распространение компьютерных вирусов вызвало разработку антивирусных программ, позволяющих обнаруживать и уничтожать вирусы, «лечить» зараженные ресурсы.

В основе работы большинства антивирусных программ лежит принцип поиска сигнатуры вирусов. *Вирусная сигнатура* - это некоторая уникальная характеристика вирусной программы, которая выдает присутствие вируса в компьютерной системе. Обычно в антивирусные программы входит периодически обновляемая база данных сигнатур вирусов. Антивирусная программа просматривает компьютерную систему, проводя сравнение и отыскивая соответствие с сигнатурами в базе данных. Когда программа находит соответствие, она пытается вычистить обнаруженный вирус.

По методу работы антивирусные программы подразделяются на фильтры, ревизоры доктора, детекторы, вакцины и другие.

Программы-фильтры, или «сторожа», постоянно находятся в оперативной памяти, являясь резидентными, и перехватывают все запросы к операционной системе на выполнение подозрительных действий, т.е. операций, используемых вирусами для своего размножения и порчи информационных и программных ресурсов в компьютере, в том

числе для переформатирования жесткого диска. Такими действиями могут быть попытки изменения атрибутов файлов, коррекции исполняемых СОМ- или ЕХЕ-файлов, записи в загрузочные секторы диска и др.

При каждом запросе на такое действие на экран компьютера выдается сообщение о том, какое действие затребовано и какая программа желает его выполнять. Пользователь в ответ на это должен либо разрешить выполнение действия, либо запретить его. Подобная часто повторяющаяся «назойливость», раздражающая пользователя, и то, что объем оперативной памяти уменьшается из-за необходимости постоянного нахождения в ней «сторожа», являются главными недостатками этих программ. К тому же программы-фильтры не «лечат» файлы или диски, для этого необходимо использовать другие антивирусные программы. Примером программ-сторожей являются *AUP*, *Norton Ant (Virus for Windows 95)*, *McAfee Virus Scan 95*, *Thunder Byte Professional for Windows 95*.

Надежным средством защиты от вирусов считаются *программы-ревизоры*. Они запоминают исходное состояние программ, каталогов и системных областей диска, когда компьютер еще не был заражен вирусом, а затем периодически сравнивают текущее состояние с исходным. При выявлении несоответствий (по длине файла, дате модификации, коду циклического контроля файла и др.) сообщение об этом выдается пользователю. Примером программ-ревизоров являются программа *Adinf* фирмы «Диалог-Наука» и дополнение к ней в виде *Adinf Cure Module*.

Программы-доктора не только обнаруживают, но и «лечат» зараженные программы или диски, «выкусывая» из зараженных программ тело вируса. Программы этого типа делятся на *фаги* и *полифаги*. Последние служат для обнаружения и уничтожения большого количества разнообразных вирусов. Наибольшее распространение в России имеют такие полифаги, как *MS AntiVirus*, *Aids test* и *Doctor Web*, которые непрерывно обновляются для борьбы с появляющимися новыми вирусами.

Программы-детекторы позволяют обнаруживать файлы, зараженные одним или несколькими известными разработчикам программ вирусами.

Программы-вакцины, или иммунизаторы, относятся к резидентным программам. Они модифицируют программы и диски таким образом, что это не отражается на работе программ, но вирус, от которого производится вакцинация, считает их уже зараженными и не внедряется в них.

К настоящему времени зарубежными и отечественными фирмами и специалистами разработано большое количество антивирусных программ. Многие из них, получившие широкое признание, постоянно пополняются новыми средствами для борьбы с вирусами и сопровождаются разработчиками.

У российских пользователей персональных компьютеров популярностью пользуется антивирусный комплекс АО «Диалог-Наука», в который входят программа-ревизор диска *Adinf* и лечащий блок *Adinf Cure Module*. Одна из последних версий этой программы-полифага *A ids test* обнаруживает более 1700 вирусов. *Aidstest* для своего нормального функционирования требует, чтобы в оперативной памяти не было других резидентных антивирусных программ, блокирующих запись в программные файлы, поэтому их следует предварительно выгрузить.

При запуске программы *Aidstest* проверяет оперативную память на наличие известных вирусов и обезвреживает их. При этом парализуются функции вируса, связанные с размножением, а другие побочные эффекты могут оставаться, в связи с чем после окончания обезвреживания вируса программа выдает запрос о перезагрузке. Перезагрузку рекомендуется осуществить кнопкой **RESET**, так как при перезагрузке клавишами **[CTRL]+[Alt]+[Del]** некоторые вирусы могут сохраняться. Кроме того, компьютер и антивирусную программу лучше запустить с защищенной от записи дискеты, чтобы при запуске с зараженного диска вирус не смог записаться в память резидентом и препятствовать лечению.

Aidstest тестирует программное тело на наличие известных вирусов, а также по искажению в своем коде судит о заражении неизвестным вирусом, при этом возможны случаи «ложной тревоги», например при сжатии антивируса программой-упаковщиком.

Программа не имеет графического интерфейса, режимы ее работы задаются с помощью ключей. Указав путь, можно проверить не весь диск, а отдельный подкаталог. Оптимальный режим для ежедневной работы задается ключами */g* (проверка всех файлов) и */s* (медленная проверка). Увеличение времени при таких опциях практически не ощутимо.

Ключ *//следует* использовать тогда, когда *Aidstest*, а также другие программы-антивирусы указывают на наличие вируса в каком-либо файле. При обнаружении вируса в ценном для пользователя файле этот файл следует переписать на дискету и попытаться вылечить с помощью ключа */f*. Если попытка не увенчается успехом, надо удалить все зараженные копии файла и проверить диск снова. Если в файле содержится важная

информация, которую нежелательно удалять, можно сархивировать файл или найти другую антивирусную программу, способную лечить этот тип вируса.

Для создания в файле протокола работы программы *Aidstest* служит ключ */p*. Протокол нужен, если пользователь не успевает просмотреть имена зараженных файлов. Для поддержки антивирусного программно-аппаратного комплекса *Sheriff* предназначен ключ */г*.

Программа-полифаг *Doctor Web* необходима прежде всего для борьбы с полиморфными вирусами, которые появились сравнительно недавно. Так же как и *Aidstest*, *Doctor Web* обновляется не реже раза в месяц, а в промежутках между версиями выходят 1-3 дополнения вирусной базы *Doctor Web*.

Использование программы *Doctor Web* для проверки дисков и удаления обнаруженных вирусов в целом подобно *Aidstest*, в

связи с чем эту программу можно запускать сразу после (или до) запуска *Aids test*. При этом практически не происходит «дублирования», так как *Aidstest* и *Doctor Web* работают на разных наборах вирусов.

В режиме эвристического анализа программа *Doctor Web* способна эффективно определять файлы, зараженные новыми, неизвестными вирусами. Применяя одновременно *Aidstest* и *Doctor Web* для контроля дискет и получаемых по сети файлов, можно почти наверняка избежать заражения.

С программой *Doctor Web* можно работать как в режиме полноэкранного интерфейса с использованием меню и диалоговых окон, так и в режиме командной строки. В командной строке задаются диск, путь и необходимые ключи. Среди ключей: */o/* - диагностика всех файлов на заданном устройстве; */p* - удаление вирусов с подтверждением пользователя; */</>* - удаление файлов, корректное «лечение» которых невозможно; *ICU* - «лечение» дисков и файлов; */zp* - запись протокола работы в файл. При работе в режиме полноэкранного интерфейса после запуска антивирусной программы пользователь использует необходимые установки через пункты основного меню: *Тест Настройки Дополнения/*.

Переход на использование операционной системы Windows 95/ NT породил проблемы с защитой от вирусов, создаваемых специально для этой среды. Кроме того, появилась новая разновидность инфекции - макровирусы, «вживляемые» в документы, подготавливаемые текстовым процессором Word и электронными таблицами Excel. АО «Диалог-Наука» предложен программно-аппаратный комплекс *Sheriff*, предназначенный для антивирусного

мониторинга и защиты, но он предполагает установку в ПК дополнительной платы. Известными антивирусными программами являются *AntiViral Toolkit Pro (AVP32)*, *Norton AntiVirus for Windows*, *McAfee VirusScan*, *Sophos SWEEP for Windows*, *Thunder BYTE AntiVirus Utilities* и др. Эти программы работают в виде программ-сканеров и проводят антивирусный контроль оперативной памяти, папок и дисков, содержат алгоритмы для распознавания новых типов вирусов, позволяют в процессе проверки лечить файлы и диски.

Программа *AntiViral Toolkit Pro (AVP32)* является 32-разрядным приложением, работающим в Windows NT, имеет удобный пользовательский интерфейс, систему помощи, гибкую систему настроек, выбираемых пользователем, распознает более 7 тыс. различных вирусов. Для работы этой программы компьютер должен иметь не менее 4 Мбайт оперативной памяти и не менее 2 Мбайт свободного места на жестком диске. *AntiViral Toolkit Pro* распознает (детектирует) и удаляет полиморфные вирусы, вирусы-мутанты и вирусы-невидимки, макровирусы, заражающие документ Word и таблицы Excel, объекты *Access* - «тройные кони».

Важная особенность этой программы состоит в возможности контроля всех файловых операций в системе в фоновом режиме и обнаружении вирусов до момента реального заражения системы, а также в возможности детектирования вирусов внутри архивов формата ZIP, ARJ, ZHA, RAR.

Интерфейс программы *AllMicro AntiVirus for Windows 98* довольно прост и не требует от пользователя дополнительных знаний о продукте - просто нужно нажать кнопку Пуск (Scan), после чего начинается проверка или сканирование оперативной памяти, загрузочных и системных секторов жесткого диска, а затем всех файлов, включая архивные и упакованные. Но простота работы не означает плохое качество: в базе данных программы содержится более 8000 сигнатур, и она может проверять не только традиционные исполняемые файлы, но и архивы, и новые типы файлов Windows 98/NT. При этом пользователь имеет возможность задать строку условий для поиска не знакомых программе вирусов.

Программа *Vscan 98* каждый раз в процессе начальной загрузки проверяет память компьютера, загрузочные секторы системного диска и все файлы в корневом каталоге. Две другие программы пакета (*McAfee Vshield* и *Vscan*) созданы как приложения Windows 98. Первая обеспечивает после загрузки Windows 95 слежение за вновь подключенными дисками, контроль исполняемых программ и копируемых файлов, вторая программа служит

для дополнительной проверки памяти, дисков и файлов. Пакет *McAfee VirusScan* умеет находить макровирусы в файлах *MS Word*.

Учитывая развитие локальных компьютерных сетей, электронной почты и сети Интернет и внедрение сетевой ОС *Windows NT*,

разработчиками антивирусных программ разработаны и поставляются на рынок такие программы, как *Mail Checker* - для проверки входящей и исходящей электронной почты, *AntiViral Toolkit Pro* для *Novell NetWare (A VPN)* - для обнаружения, лечения, удаления и перемещения в специальный каталог пораженных вирусом файлов при работе с сетевой ОС *Novell NetWare* версий 3.x и 4.x.

AVPN работает как антивирусный сканер и фильтр, постоянно контролируя хранящиеся на сервере файлы. В режиме фильтра сканируются на наличие известных файловых вирусов файлы, приходящие на сервер и исходящие с сервера (в том числе запускаемые и считываемые), в режиме сканера происходит немедленное или автоматическое сканирование томов сервера.

AVPN имеет возможность удалять, перемещать и «лечить» зараженные объекты; проверять упакованные и архивные файлы; детектировать неизвестные вирусы с помощью эвристического механизма; проверять в режиме сканера удаленные серверы; отключать зараженную станцию от сети и т.д.

Кроме того, *AVPN* легко настраивается для сканирования файлов различных типов; имеет удобную схему пополнения антивирусной базы; посылает сообщения о заражении сервера вирусом по сети, электронной почте и на пейджер; осуществляет автоматическое ведение файла-отчета о проделываемых операциях и управление программой с рабочей станции.

ЗАЩИТА ПРОГРАММНЫХ ПРОДУКТОВ

Программные продукты - важные объекты защиты по ряду причин. Во-первых, они являются продуктом интеллектуального труда специалистов высокой квалификации, зачастую групп в несколько десятков или даже сотен человек. Во-вторых, процесс проектирования этих продуктов связан с потреблением значительных материальных и трудовых ресурсов, основан на использовании дорогостоящего компьютерного оборудования и наукоемких технологий. В-третьих, восстановление нарушенного про-

граммного обеспечения требует значительных трудозатрат, а простой вычислительного оборудования чреват негативными результатами для организаций или физических лиц.

Защита программных продуктов преследует следующие цели:

- ограничить несанкционированный доступ отдельных категорий пользователей к работе с ними;
- исключить преднамеренную порчу программ с целью нарушения нормального хода обработки данных;
- исключить преднамеренную модификацию программы с целью порчи репутации производителя программной продукции;
- исключить несанкционированное тиражирование (копирование) программ;
- исключить несанкционированное изучение содержания, структуры и механизма работы программы.

Программные продукты должны защищаться от несанкционированных воздействий различных объектов: человека, технических средств, специализированных программ, окружающей среды и др.

Человек может воздействовать на программный продукт путем хищения или физического уничтожения документации на программу или самого машинного носителя, нарушения работоспособности программных средств и др.

Технические средства (аппаратура) путем подключения к компьютеру или к передающей среде могут произвести считывание, расшифровку программ, а также их физическое разрушение.

С помощью специализированных программ могут быть проведены вирусное заражение программного продукта, его несанкционированное копирование, незаконное изучение его содержания и др.

Наконец, окружающая среда в силу аномальных явлений (повышенное электромагнитное излучение, пожар, наводнение и др.) может привести к физическому разрушению программного продукта.

Самым простым и доступным способом защиты программных продуктов является ограничение доступа к ним путем:

- парольной защиты программ при их запуске;
- использования ключевой дискеты;

- использования специального технического устройства (электронного ключа), подключаемого к порту ввода-вывода компьютера.

Во избежание несанкционированного копирования программ специальные программные средства защиты должны:

- идентифицировать среду, из которой запускается программа;
- вести учет количества выполненных санкционированных инсталляций или копирования;
- противодействовать (вплоть до саморазрушения) изучению алгоритмов и программ работы системы.

Эффективными защитными мерами для программных продуктов являются: нестандартное форматирование запускающей дискеты; закрепление месторасположения программы на жестком диске; привязка к электронному ключу, вставляемому в порт ввода-вывода; привязка к номеру BIOS и др.

Защита программных продуктов должна обязательно осуществляться и правовыми методами, в числе которых лицензионные соглашения и договоры, патентная защита, авторские права, технологическая и производственная секретность и др.

ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ ДАННЫХ НА АВТОНОМНОМ КОМПЬЮТЕРЕ

Независимо от характера решаемой задачи на ПК ценность данных во многом связывается с тем, как дорого оценивается время пользователя, сколько данных сохранено и как долго придется вновь создавать потерянные данные.

Наиболее типичными случаями, создающими угрозу данным, являются случайное стирание данных, отказ программного обеспечения и аппаратные сбои. Кроме того, потеря данных на автономном ПК может происходить из-за стихийных бедствий и несанкционированного доступа к ПК других людей. Самая первая рекомендация пользователю состоит в резервировании данных. Для длительной сохранности данные хранятся на магнитных дисках, но, к сожалению, не вечно. Для МД существует такой параметр, как среднее время между отказами, хотя этот параметр может выражаться в годах, резервное копирование нередко нелишне. При работе на ПК данные иногда не читают-

ся из-за выхода из строя платы управления жестким диском. Заменяв плату контроллера и перезагрузив ПК, можно вновь выполнять прерванную работу.

Создание резервных копий обеспечивает сохранность данных при стихийных бедствиях, а также нередких сбоях напряжения в сети. График резервного копирования должен быть основан на оценке вероятности возможных повреждений и сбоев, еще не приводящих к серьезным последствиям. Как правило, современные коммерческие программы для работы с текстом предусматривают автоматическое сохранение документа в резервном файле каждые 5-10 мин. Практика рекомендует делать резервные копии файлов данных после установки новых приложений, ибо сами программы можно всегда переустановить с инсталляционных (установочных) гибких дисков или с компакт-дисков (CD-ROM). При этом, правда, возникают вопросы о выборе среды и средства резервного копирования. Тенденции развития ПК - увеличение емкости жестких дисков до нескольких Гбайтов - делают нереальной создание полной копии диска на дискетах. Для подтверждения этого достаточно привести такой расчет: для копирования 1 Гбайта данных потребуется около 700 дискет емкостью записи по 1,44 Мбайта каждая. В силу этого наиболее доступный вариант для копирования больших объемов данных - использование устройства резервного копирования на магнитную ленту, т.е. стриммера. При выборе стриммера следует руководствоваться, как минимум, двумя правилами:

- приобретать стриммер большей емкости (желательно больше или сопоставимого с емкостью жесткого диска) для уменьшения числа перестановок ленты в нем;
- приобретение внешнего или внутреннего стриммера.

Преимущество внешнего стриммера - возможность использования одного устройства для копирования с нескольких ПК; преимущества внутреннего - удобство в работе, например, автоматическое копирование в отсутствие пользователя на рабочем месте.

Использование метода резервного копирования как способа обеспечения безопасности данных требует выбора программного продукта, процедуры (полное, частичное или выборочное копирование) и частоты резервного копирования (частичное в середине и полное в конце дня; полное в конце рабочего дня; раз в неделю и др.). В зависимости от важности информации (или качества магнитной ленты) иногда производится дубль-резервное копирование, не следует также пренебрегать тестированием резервных копий. Наилучшим методом для хранения резервных копий на различных носителях

является несгораемый шкаф; кроме того (опять же из особенностей задач), лучше хранить оригиналы и копии в разных помещениях.

В автономно работающем компьютере либо работающем в малой сети, где пользователи используют общие ресурсы файлового сервера, тоже нередко следует защищать данные даже от людей, с которыми пользователь хорошо знаком, учитывая психологическую черту людей интересоваться чужими вещами (делами, информацией). К таким простым методам обеспечения безопасности относятся атрибуты файлов и каталоги типа «скрытый» и/или «только для чтения»; сохранение важных данных на гибких магнитных дисках; размещение данных в защищенные паролем архивные файлы; включение в защитную программу регулярной проверки на компьютерные вирусы.

Одним из свойств компьютерного вируса признается его возможность оставаться незамеченным достаточно долго, чтобы распространиться на другие компьютеры. По этой причине перед началом активных действий многие вирусы имеют длительный «инкубационный» период. Поэтому пользователь должен использовать имеющийся у него шанс обнаружить и уничтожить вирус с помощью антивирусной программы. Учитывая, что множество новых вирусов создается практически каждый месяц, рекомендуется с целью повышения безопасности данных антивирусное программное обеспечение обновлять хотя бы не реже одного раза в месяц.

Имеются три основных способа использования антивирусных программ. В первом случае поиск вируса производится при начальной загрузке, когда команда запуска антивирусной программы включается в AUTOEXEC.bat. Способ достаточно эффективен, но при этом увеличивается время начальной загрузки компьютера, и для слишком нетерпеливых пользователей он оказывается угнетающим. Преимущество метода в том, что поиск вирусов в программах происходит автоматически.

Второй метод состоит в том, что пользователь запускает вирусную программу вручную, когда считает необходимым. Третий метод поиска вирусной инфекции заключается в визуальном просмотре каждого загружаемого файла.

Для предотвращения записи вируса на загрузочные дискеты или дискеты с какими-либо прикладными программами пользователь может применить такой примитивный, но эффективный прием, как передвижение пластмассового квадратика в углу 3,5-дюймовой дискеты.

Достаточно прагматичным способом обеспечения безопасности информации на автономном ПК является парольная защита (ряд общих положений был рассмотрен в разд. 11.3 данного учебника).

Из предыдущего материала известно, что в составе оперативной памяти имеются микросхемы, хранящие специальные программы начальной загрузки и защиты паролями разных операций компьютера. Эти специальные программы записаны непосредственно в кристалле, выполняются перед запуском ОС и не зависят от нее. После включения ПК и запуска программы установки CMOS пользователю предлагается дважды ввести какую-то информацию, которая становится паролем. В дальнейшем защита с использованием пароля на уровне BIOS блокирует компьютер целиком, если не введен правильный пароль¹.

Если нежелательно использовать пароль при начальной загрузке, то некоторые модели клавиатуры можно заблокировать с помощью физических ключей, поставляемых в комплекте с компьютером. Правда, при этом работоспособной остается мышь.

Возможность защиты отдельных файлов предусматривается при работе пользователя с офисными пакетами (текстовыми процессорами, электронными таблицами, СУБД и др.) и прежде всего при выполнении команды сохранения файлов (Сохранить как ...). Если при этом нажать на кнопку Options (Параметры), то в открывшемся диалоговом окне можно задать пароль, ограничивающий возможности работы с этим документом. Для восстановления первоначальной формы защищенных таким образом данных необходимо ввести тот же самый пароль. Правда, не стоит забывать, что пароль (а тем более несколько паролей) достаточно просто можно забыть либо, записав его на бумажном носителе, элементарно потерять. В этом случае пользователь автономного ПК может столкнуться с еще большими неприятностями, чем при работе без парольной защиты.

¹ В специальной литературе предлагается довольно простой способ нейтрализации пароля начальной загрузки - удалить провода, соединяющие батарейку питания и кристалл CMOS. Итак, можно сделать некоторые выводы.

Способы защиты ПК, работающих автономно или в составе небольшой сети, дома или в офисе, достаточно разнообразные. Выработывая стратегию защиты информации на ПК, надо найти рациональный компромисс между ценностью защищаемых данных, затратами на обеспечение защиты и неудобствами, налагаемыми системой защиты на работу с данными.

. БЕЗОПАСНОСТЬ ДАННЫХ В ИНТЕРАКТИВНОЙ СРЕДЕ

Интерактивные среды более уязвимы с позиций безопасности данных, а многочисленные и разнообразные средства несанкционированного доступа к ним делают защиту сложнее. Примерами интерактивных сред, где происходит взаимодействие большого числа различных элементов, являются любые системы с коммуникационными возможностями, например электронная почта, компьютерные сети, Интернет. Причем глобализация информационных связей будет продолжаться, а вместе с этим процессом возникают, с одной стороны, новые формы и методы противоправных действий, а с другой - способы обеспечения достаточной безопасности данных.

Употребленный термин *электронная почта* достаточно широк, и под ним понимается любой вид связи с использованием компьютеров и модемов. Отправляя кому-либо сообщения через локальные вычислительные сети, коммерческие интерактивные службы, корпоративную сеть, через сеть Интернет, собственник информации посылает электронную почту. При этом следует помнить, что человек с правами супервизора имеет доступ абсолютно ко всему в сети или по крайней мере к трафику, проходящему через его систему.

Электронная почта, направленная через сеть телефонной компании или через Интернет, прежде чем достичь адреса, проходит через многочисленные «почтовые отделения». Любое сообщение, предназначенное для отправки по электронной почте, обрабатывается почтовой программой. Некоторые из них производят шифровку сообщения, большинство же пересылают сообщения в виде обычного текста, используя протоколы POP или SMTP . Электронную почту можно сравнить с пересылкой по обычной почте почтовых карточек (а не конвертов), а это значит, что электронное сообщение может быть продублировано (скопировано) перед отправкой, прочтено системным администратором сетевого узла, а, кроме того, сама технология нередко требует, чтобы делались копии всей проходящей через систему электронной почты на случай аварийного отказа электронной почтовой системы.

Наиболее незащищенными местами в электронной почте являются пункт исходящей почты отправителя и почтовый ящик получателя. Почти каждый программный пакет электронной почты позволяет архивировать входящие и исходящие сообщения

по любому другому адресу, чем могут воспользоваться злоумышленники. Следует также помнить, что почтовые сообщения не поступают непосредственно на компьютер конечного пользователя - вначале они записываются на сервер провайдера получателя и хранятся там, пока конечный пользователь не даст команду «доставить почту». Большинство почтовых программ имеет опцию «удалить прочитанную почту», освобождая почтовый ящик пользователя; однако это не означает, что сообщение удаляется с сервера полностью. Технологией обеспечения работоспособности и восстановления электронной почты может быть предусмотрен вариант, когда весь входящий трафик сообщений дублируется на отдельный жесткий диск или на другой сервер. Тогда информация хранится долго, и хакер, получивший статус супервизора на такой системе, может достаточно свободно читать сообщения.

Для повышения безопасности в состав некоторых почтовых программ включаются функции шифрования, усиливающие конфиденциальность пересылки сообщения: не зная ключевого слова, использовавшегося для шифрования, достаточно сложно и трудоемко раскодировать сообщение. Специалистами признается, что достаточно надежными и эффективными системами шифрования являются PEM (повышенная конфиденциальность) и PGP (достаточно надежная секретность). Одна из мер предосторожности при работе с электронной почтой - использование утилит, реализованных в браузерах. В частности, в Netscape имеется утилита Unsczamble BOT-13, которая просто заменяет символы в сообщении, сдвигая их на 13 позиций латинского алфавита (к примеру, А становится М; В заменяется на N и т.д.). Для чтения закодированного таким способом сообщения получателем достаточно воспользоваться этой же утилитой, восстановив сообщение.

В любом электронном сообщении всегда присутствуют адреса отправителя и получателя. Для сокрытия адреса отправителя в системе электронной почты используются два приема: фальшивая почта и анонимная почта. Фальшивая почта - это возможность отправки электронных сообщений через Интернет, используя измененный адрес корреспондента. В случае анонимной почты отправитель, используя анонимный транзитный почтовый узел (ремейлер), скрывает свое имя от всех, кроме получателя. Анонимный ремейлер - это почтовый сервер, разработанный для приема посылаемого ему сообщения, удаления всей идентифицирующей отправителя информации в заголовке и замены ее на анонимную информацию. Затем сообщение передается дальше. В некоторых слу-

чаях (чтобы остаться неизвестным) собственник данных посылает их через несколько таких серверов.

Электронная почта, обеспечивая пересылку сообщения, способна принести в то же время немалый вред получателю сообщений. Для устранения нежелательных последствий нужно использовать и другие приемы безопасности, в частности:

- никогда не следует сразу запускать программы, полученные по электронной почте, особенно вложения. Следует сохранить файл на диске, проверить его антивирусной программой и только потом запускать. Если есть подозрение, что в программе имеется вирус, письмо надо удалить;
- не доверять адресам «солидных» отправителей. Поскольку адрес отправителя можно подделать, пользователь может получить «антинужную» информацию или вирус;
- не сообщать свой пароль и личные данные, если отправитель предлагает адресату даже нечто очень заманчивое;
- открывая полученные файлы MS Office (в Word, Excel и др.), по возможности не использовать макросы;
- стараться пользоваться проверенными, а также более новыми версиями почтовых программ.

Важной проблемой для пользователей Интернета стала проблема безопасности данных в самой сети. Напомним, что Интернет развивалась из сети ARPAnet - сети коммуникаций пакетов, автоматически направляющей сообщения между компьютерами в сети и способной надежно функционировать в ситуациях, когда отдельные ее части разрушены. При этом особая сложность в организации безопасности связана с Всемирной информационной паутиной WWW.

Подключение пользователя к ресурсам сети Интернет осуществляется через провайдера. Как правило, почти все провайдеры приобретают свое оборудование и программное обеспечение у небольшой группы производителей, поэтому предлагаемые ими функции и услуги практически ничем не отличаются, за исключением цены, надежности и качества обслуживания. В связи с однотипностью предоставляемых услуг в сети Интернет возможно применение некоторых универсальных приемов нарушения безопасности компьютера пользователя.

Для защиты информации от хулиганствующих элементов, неквалифицированных пользователей и преступников в системе Интернет используется система полно-

мочий, или иначе управление доступом. Каждый файл данных (или другие ресурсы компьютера) снабжается набором атрибутов, которые сообщают, что, к примеру, данный файл может просмотреть кто угодно, но изменять его разрешается лишь владельцу. Другая ситуация состоит в том, что никто, кроме владельца, не может просмотреть файл, хотя и видны имена этих информационных ресурсов.

Обычно пользователь хочет каким-то образом защитить свою информацию, но надо помнить, что системные администраторы и «умные вредители» могут преодолеть системы защиты. В этом случае на помощь приходят разнообразные методы шифрования информации с использованием ключей, разработанных пользователем.

Одной из проблем работы в сети Интернет является ограничение доступа некоторых категорий пользователей к информационным ресурсам (например, детей и школьников к сексуальным материалам). Обеспечить это можно с помощью специальных программных продуктов - брандмауэров (например, Cyber Patrol, Net Nanny, Surf-Watch и др.), которые базируются на принципе фильтрации по ключевым словам, фиксированным спискам мест служб WWW, где содержится нежелательный для детей материал. Такие словари создаются родителями (для дома) и учителями (для школ).

Аналогичного вида программы, ведущие запись сеансов Интернета и отказывающие в доступе к определенным местам сети, например *WebTzack*, могут устанавливаться в офисных и других учреждениях, чтобы их работники не тратили время в личных интересах. Интернет - это система, где многочисленные пользователи имеют свои Web-серверы, содержащие рекламную или справочную информацию на Web-страницах. Конкуренты, взломав систему защиты, могут изменить или испортить содержание Web-страничек. Лучший способ избежать неприятностей в таких ситуациях - регулярно просматривать Web-странички и в случае обнаружения порчи информации восстанавливать заранее заготовленными копиями файлов.

Следует иметь в виду, что обеспечивать безопасность информации на серверах должны провайдеры, систематически просматривая протоколы событий и обновляя программное обеспечение, если в нем обнаруживаются проблемы в защите.