

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПУТЕЙ СООБЩЕНИЯ (УрГУПС)**



Курс лекций

В. Н. Малышев

ИНФОРМАТИКА

**Федеральное агентство железнодорожного транспорта
Уральский государственный университет путей сообщения**

Кафедра «Прикладная информатика»

В.Н. Мальшев

ИНФОРМАТИКА

КУРС ЛЕКЦИЙ

для студентов 1 курса технических специальностей
всех форм обучения

2-е издание, переработанное и дополненное

Екатеринбург
2009

УДК 007 ББК 32.81
М 20

В. Н. Малышев

М20 Информатика: Курс лекций.– Изд. 2-е, перераб. и доп.– Екатеринбург: УрГУПС, 2009 – 246 с., ил.

Курс лекций по информатике составлен в соответствии с требованиями Государственного образовательного стандарта к минимуму содержания и уровню подготовки и предназначен для студентов 1 курса технических специальностей всех форм обучения (≈ 200 ч.).

В лекциях изложены краткие сведения об информации, ее свойствах, восприятии, сборе, представлении, измерении, кодировании и обработке любой информации на компьютере. Дано понятие об алгоритмах, описаны их свойства, способы записи и принципы конструирования. Рассмотрены способы создания компьютерных программ, их классификация и назначение. Приведены понятия алгебры логики. Кратко описана структура информационных систем, приведена их классификация, рассмотрены вопросы обеспечения безопасности информации в информационных системах и особенно в компьютерных сетях.

Также изложены физические основы и принципы работы компьютера, принципы хранения информации в компьютере, взаимодействие его устройств. Дано понятие об архитектуре, организации и конфигурации ЭВМ.

Большое внимание уделено локальным вычислительным сетям и Интернету как основе современных информационных технологий.

Курс лекций может быть полезен и студентам других специальностей, изучающим общий курс информатики.

Автор:

В. Н. Малышев, доцент кафедры «Прикладная информатика», УрГУПС

Рецензенты:

В. И. Радченко, зав. кафедрой «Прикладная информатика»,
д-р физ.- мат. наук, проф., УрГУПС,

С. В. Фирстов, главный инженер Екатеринбургского ИВЦ–филиала
Главного вычислительного центра ОАО «РЖД», канд. тех. наук

ISBN 978 – 5 – 94614 – 107 9

©Уральский государственный университет путей сообщения (УрГУПС), 2009

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
Раздел 1. ЭЛЕМЕНТЫ ТЕОРЕТИЧЕСКОЙ ИНФОРМАТИКИ	
<i>Лекция 1.</i> Информация, ее виды и свойства. Информатизация общества. Влияние информатизации на изменение условий труда и характер производства.....	6
<i>Лекция 2.</i> Понятие об информатике и информационных технологиях. Рынок информационных продуктов и услуг. Индустрия информатики.....	15
<i>Лекция 3.</i> Связь между информацией, сообщением и сигналом. Виды сигналов и их особенности	21
<i>Лекция 4.</i> Общие процессы восприятия, сбора, передачи и представления информации человеком	25
<i>Лекция 5.</i> Процессы восприятия, сбора и передачи информации в информационных системах	37
<i>Лекция 6.</i> Системы счисления	41
<i>Лекция 7.</i> Измерение количества информации и ее кодирование	45
<i>Лекция 8.</i> Представление чисел при обработке их в компьютере	55
<i>Лекция 9.</i> Понятие об алгебре логики.....	63
<i>Лекция 10.</i> Понятие об алгоритмах	72
<i>Лекция 11.</i> Понятие о программировании.....	81
<i>Лекция 12.</i> Классификация программ.....	95
<i>Лекция 13.</i> Понятие о компьютерных вирусах и других вредоносных программах	100
<i>Лекция 14.</i> Информационные системы, их назначение, структуры и признаки классификации	113
<i>Лекция 15.</i> Понятие об экспертных системах	124
Раздел 2. КОМПЬЮТЕРНАЯ ТЕХНИКА	
<i>Лекция 16.</i> Классификация компьютеров	134
<i>Лекция 17.</i> Основные принципы работы компьютера	140
<i>Лекция 18.</i> Принципы организации ЭВМ	149
<i>Лекция 19.</i> Принципы хранения информации в компьютере	159
<i>Лекция 20.</i> Физический и логический уровни работы с компьютером	164
Раздел 3. ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ	
<i>Лекция 21.</i> Понятие о телеобработке и вычислительных сетях	167
<i>Лекция 22.</i> Классификация сетей. Способы коммутации и передачи данных	173
<i>Лекция 23.</i> Методы доступа в ЛВС	180
<i>Лекция 24.</i> Обеспечение безопасности информации в ЛВС	189
<i>Лекция 25.</i> Глобальные сети. Сеть Интернет	194
<i>Лекция 26.</i> Службы или сервисы Интернета	203
<i>Лекция 27.</i> Поиск информации в Интернете. Классификация Web-сайтов	210
ЗАКЛЮЧЕНИЕ	224
Приложение. Краткий понятийно-терминологический словарь курса.....	226
Библиографический список	242

Окружающая нас информация абстрактна, объем ее безграничен. Конкретны **знания**, определяющие развитие человечества. Главная цель информатики – превращать информацию в знания.

Введение

Рыночные отношения, изменение структуры управления отраслями народного хозяйства, крупными холдингами и корпорациями выдвинули на передний план решение проблем информатизации общества. Без своевременной, достоверной и полной информации немислима эффективная маркетинговая, финансово-кредитная, инвестиционная, управленческая и другая деятельность. Только на основе владения этой информацией, возможны эффективное управление любой сферой человеческой деятельности, правильное принятие решений.

Для получения своевременной и достоверной информации небывало высокими темпами развиваются интегральные информационно-вычислительные сети связи. Развитие общемировой телекоммуникационной сети Интернет и ее влияние на современный мир не имеют исторических аналогов. Сеть является одновременно не только средой для сотрудничества и общения людей, средством общемирового вещания и распространения информации, но и мощным инструментом для ведения бизнеса, лишенного каких-либо временных и пространственных ограничений. Быстрыми темпами развиваются новые информационные технологии, появилась новая наука – **информатика**, изучающая информационные процессы во всех сферах человеческого общества.

Информатизация всегда приводит к коренной перестройке технологии практически во всех отраслях промышленности, коммерческой и финансово-кредитной деятельности, в управлении и, как следствие, к повышению производительности и улучшению условий труда. Еще в конце 70-х гг. прошлого века директор НИИ Кибернетики АН УССР академик В. М. Глушков писал: «Попытка вставить ЭВМ в традиционную технологию управления эквивалентна попыткам установить реактивный двигатель на телегу или построить мощную ТЭЦ, в котлы которой уголь забрасывался бы лопатами».

Создание информационной поддержки принятия решений, банков данных и баз знаний, поиск методов решения интеллектуальных задач, использование эвристических методов моделирования исследуемых операций, создание экспертных систем, способных принимать решения на уровне экспертов-профессионалов, всемерное использование информационных технологий компьютерных сетей и многое другое остро ставят вопрос о подготовке кадров, имеющих прочные знания в области современных информационных технологий и эффективного использования информационных и телекоммуникационных ресурсов. Современный инженер, экономист, юрист, врач, любой другой специалист обязаны владеть знаниями в области информатики и практическими навыками ра-

боты на компьютере по обработке информации, уметь пользоваться телекоммуникационными вычислительными системами, чтобы быстро найти необходимую информацию, знать основы новых информационных технологий, уметь оценивать точность и полноту информации, влияющей на принятие управленческих решений.

Общие курсы информатики, читаемые для студентов всех специальностей УрГУПС, на отдельных специальностях, до сих пор сводятся большей частью к тому, чтобы научить студента работать на компьютере с приложениями, позволяющими создавать простые текстовые, табличные и другие документы, и основам программирования на том или ином языке высокого уровня. Но сейчас уже новое время и выполнять простые работы в текстовом редакторе или в табличном процессоре учат практически во всех школах, а программирование поставлено на индустриальную основу. Им занимаются многочисленные крупные фирмы программистов, профессионалов высокого класса. Программы стали продуктом купли-продажи. Их разработано на все случаи жизни сотни тысяч. Практически для решения любой задачи можно найти готовую программу. Необходимо знать назначение, возможности и свойства каждой программы или пакета программ, ясно представлять, где и какую программу следует использовать, знать их разновидности, уметь работать в любой операционной среде с любым приложением. *Хороший пользователь в настоящее время ценится не менее чем хороший программист.*

Поэтому автор считает, что студентам большинства технических специальностей лекционный курс по общей информатике должен дать основные теоретические, фундаментальные понятия информатики, а курс лабораторных и самостоятельных работ – умение разрабатывать модели решения самых разнообразных задач и работы на компьютере с любым приложением и с телекоммуникационными сетями, в частности с Интернетом.

Раздел 1. ЭЛЕМЕНТЫ ТЕОРЕТИЧЕСКОЙ ИНФОРМАТИКИ

Лекция 1

Информация ее виды и свойства. Информатизация общества. Влияние информатизации на изменение условий труда, характер производства и развитие общества.

Во второй половине XX в. человечество вступило в новую фазу своего развития. Закончился этап индустриального развития, и начался переход от индустриального общества к информационному. Процесс, обеспечивающий этот переход, получил название информатизации. Информатизация охватывает все сферы деятельности человека, и прежде всего элементы рыночной инфраструктуры экономики:

- промышленные и сельскохозяйственные предприятия;
- складское и транспортное хозяйство;
- сеть оптовой и розничной торговли;
- товарные, фондовые, валютные биржи, биржи труда;
- сети банков и кредитно-финансовых учреждений и т. д.

Информатизация – это процесс создания, развития и всеобщего применения информационных средств и технологий, обеспечивающих достижение и поддержание уровня информированности всех членов общества, необходимого и достаточного для кардинального улучшения качества труда и условий жизни людей.

Именно информатизация призвана обеспечить решение проблем экономической реформы и на этой базе существенно повысить жизненный уровень людей, усовершенствовать структуру и ускорение процессов, определяющих научно-технический прогресс, усовершенствовать общественные, производственные, правовые отношения. По мнению специалистов-экспертов, любая страна, насколько бы индустриально развитой она ни была, перейдет в разряд стран третьего мира, если опоздает с информатизацией. Первыми на этот путь встали в середине XX в. США, Япония и страны Западной Европы. Характерным примером влияния информатизации на развитие страны, которому в мире нет аналогов, является Япония. Имея численность населения примерно такую же, как Россия, но в 47 раз меньшую площадь и соответственно меньше природных ресурсов, Япония по развитию вышла на одно из первых мест в мире.

Информация становится важнейшим *стратегическим ресурсом* и занимает ключевое место в жизни человеческого общества в производственной и социальной сферах. Например, в информационном процессе, каким является управленческая деятельность, информация выступает как один из возможных ресурсов наряду с энергетическими, материальными, трудовыми, финансовыми и другими ресурсами. Прежде, чем создать новый материальный или интеллекту-

альный продукт, необходимо провести большую работу по сбору, обработке и анализу информации о существующих аналогах, методах и способах его создания, рыночном спросе, конкурентоспособности и т. д. В настоящее время формирование научного понятия «информация» дало людям новые знания об окружающем нас мире, показало новую сторону материального единства мира. Это позволило подойти с единой точки зрения ко многим, ранее казавшимися совершенно различными процессам. Например, что общего в передаче сообщений по техническим каналам связи, развитии биологических систем, работе компьютеров, функционировании нервной системы человека, автоматическом управлении различными процессами? Все это, помимо энергетических и материальных затрат, связано с передачей, хранением и обработкой информации. Информация во всех этих процессах играет роль, аналогичную понятию энергии в физике, дающему возможность с общей точки зрения описать самые различные физические явления. Поэтому понятие информации – *одно из самых фундаментальных* в современной жизни. Наряду с понятиями «вещество» и «энергия» оно составляет *основу* современной научной картины мира, является одной из категорий мироздания.

Источниками информации являются, прежде всего, природные объекты: галактики, планеты, звезды, леса, луга, растения, животные, люди, окружающие нас предметы и т. д. Вместе с тем по мере развития науки и техники источниками информации становятся научные эксперименты, машины, аппараты, технологические процессы. Источники информации в то же время являются и *приемниками* (потребителями) ее. Различные источники информации порождают разные способы и методы ее представления в виде звуковых, электрических, световых и других сигналов, в виде речи человека или письменного сообщения, в виде графических знаков дорожного движения, рисунков, чертежей, видеороликов, различных таблиц и т. д.

Информация, полученная в ходе научно-исследовательских опытов и экспериментов, или получаемая от работников, собирающих данные непосредственно на производстве о производственных и хозяйственных операциях, о материально-энергетических, трудовых и финансовых затратах, выпуске продукции, фактах приобретения и продажи товаров и т. д., впервые сформулированная и документированная, или информация, характеризующая технологический процесс и собираемая техническими устройствами от датчиков и измерительных приборов с целью выработки управляющих сигналов, является *первичной*. Собранная первичная информация накапливается, структурируется, обрабатывается, анализируется и передается в пункты потребления информации для использования. Это уже будет *вторичная информация, или информационный продукт*.

В зависимости от сферы применения информация подразделяется на политическую, техническую, экономическую, криминалистическую, биологическую, генетическую и т. д. Отдельно выделяют логическую информацию, адекватно отображающую объективные закономерности природы, общества и мышления. Это *научная* информация. Огромный объем составляет *управленческая* информация, которая обслуживает процессы производства, распределения, обмена и потребления материальных благ, обеспечивает решение задач органи-

зационно-экономического управления народным хозяйством и его звеньями. Это разнообразные сведения экономического, технологического, социального, юридического, демографического и другого содержания. В информационном процессе, каким является управленческая деятельность, информация выступает как один из важных ресурсов наряду с энергетическими, материальными, трудовыми, финансовыми и другими ресурсами. Она позволяет принимать как краткосрочные решения, так и стратегические решения на длительную перспективу.

Свойства информации. Для любой информации можно указать три объекта взаимодействия: источник информации, приемник информации (ее потребитель) и объект или явление, которые данная информация отражает. Им соответствуют три группы внешних свойств. Первой и важнейшей группой являются свойства информации с точки зрения потребителя.

Релевантность – способность информации соответствовать нуждам потребителя.

Полнота – свойство информации исчерпывающе характеризующее отображаемый объект или процесс (для данного потребителя).

Своевременность – способность информации соответствовать нуждам потребителя в нужный для него момент времени.

Доступность – свойство информации, характеризующее возможность ее получения данным потребителем.

Защищенность – свойство, характеризующее невозможность несанкционированного использования или изменения информации.

Эргономичность – свойство, характеризующее удобство формы или объема информации с точки зрения данного потребителя.

Достоверность – свойство информации не иметь скрытых ошибок. Это особенно важно для алгоритмов и компьютерных программ.

Научная информация характеризует не взаимоотношение «информация – потребитель», а взаимоотношение «информация – отражаемый объект или явление». Это будет вторая группа внешних свойств информации. Здесь наиболее важным является свойство *адекватности*.

Адекватность – свойство информации однозначно соответствовать отображаемому объекту или явлению. Адекватность для потребителя – внутреннее свойство информации, проявляющее себя через релевантность и достоверность.

Среди внутренних свойств информации важнейшими являются также объем (количество) информации и ее внутренняя организация (структура). О количестве и структуре информации более подробно мы будем говорить далее. Здесь же кратко заметим, что по способу внутренней организации информацию делят на две группы:

- ♦ данные, т. е. простой, логически не упорядоченный набор сведений;
- ♦ логически упорядоченные, организованные определенным образом наборы данных (таблицы, массивы, базы данных и т. д.). Во второй группе выделяют особым образом организованную информацию – *знания*. Знания, в отличие от данных, представляют собой информацию не о каком-то единичном и конкретном факте, а о том, как устроены все факты определенного типа.

К третьей группе можно отнести свойства информации, связанные с процессом ее хранения. Здесь важнейшим свойством является *живучесть* – спо-

способность информации сохранять свое качество с течением времени. К этому можно добавить еще свойство *уникальности*. Уникальной называют информацию, хранящуюся в единственном экземпляре.

Важнейшей составляющей управленческой информации является информация *экономическая*. Она отражает социально-экономические процессы как в сфере производства, так и в непроизводственной сфере, во всех органах и на всех уровнях управления: государственного, отраслевого, регионального, местного, управления концернами, холдингами, фирмами.

Особенностью экономической информации является отражение актов производственно-хозяйственной деятельности с помощью системы натуральных и стоимостных показателей (*цифровые значения*). Это предопределяет возможность широкого применения компьютерной техники в экономике.

Вторая особенность экономической информации – ее *цикличность*. Для большинства производственных и хозяйственных процессов характерна повторяемость составляющих их стадий и информации, отражающей эти процессы. Цикличность позволяет, однажды создав программу машинного счета, использовать ее многократно в течение длительного времени.

Экономическая информация отражается в материальных носителях или передается по каналам связи. Для повышения ее достоверности ведутся только передача и обработка юридически оформленной информации (с подписями ответственных лиц, с кодами, подтверждающими достоверность передающих сообщений).

Еще одна особенность экономической информации – ее *объемность*. Качественное управление экономическими процессами невозможно без детальной информации о них. Экономическая информация описывает разные сущности, как простые, так и сложные, имеющие определенные свойства. Например, материал обладает массой, габаритами, имеет цену, относится к конкретному виду материальных ресурсов.

Совокупность сведений, отражающих какую-либо сущность, называют *информационной совокупностью*. Информационные совокупности могут быть разного уровня и иметь иерархическую структуру. Так, данные о поставщике включают его наименование и адрес, номенклатуру поставляемой продукции или услуг, условия поставки, наличие продукции на складе, максимальные сроки поставки, фактические сведения о произведенных поставках и т. д. Информационная совокупность, неделимая далее на более мелкие смысловые единицы, получила название *реквизита*. Различают два вида реквизитов: *реквизиты-признаки* и *реквизиты-основания*. Первые характеризуют качественные свойства отражаемых сущностей. Вторые представляют собой количественные величины, характеризующие данную сущность. Сочетание одного реквизита-основания с одним или несколькими соответствующими ему реквизитами-признаками образует *показатель*, являющийся информационной совокупностью наименьшего состава, достаточной для образования самостоятельного сообщения или формирования документа. Например, информационная совокупность «10 цистерн мазута» состоит из реквизита-основания «10» и реквизитов-признаков

«цистерн» и «мазута», что вполне отражает экономический смысл сообщения и потому является показателем.

Умение определить количество и состав реквизитов в документе позволяет оценить его уровень информативности, рассчитать при необходимости объемы информации в документе. Зная максимальную разрядность каждого реквизита и число таких документов, можно рассчитать общий объем информации.

Изобретение и освоение человеческого языка, устная речь выделили человека из мира животных. Это позволило ему хранить, передавать, совершенствовать и увеличивать приобретенную информацию (знания). С этого момента в развитии процессов накопления, хранения, обработки и передачи информации выделяют **четыре революционных этапа**, давших мощный толчок новым информационным технологиям:

- возникновение письменности – появилась возможность фиксации, накопления и передачи знаний от поколения к поколению;

- изобретение книгопечатания позволило радикальным образом изменить культуру и организацию деятельности в обществе, поскольку воспроизведение информации было поставлено на поток, на промышленную основу. Это повысило сохранность информации, доступность ее для большого числа людей и точность ее воспроизведения;

- открытие электричества – появились радио, телефон, телеграф, магнитная запись, позволившие оперативно передавать информацию на большие расстояния и накапливать ее в больших объемах;

- появление полвека тому назад компьютеров – появилась возможность индустриальной обработки информации и создания телекоммуникационных компьютерных систем.

Усложнение индустриального производства, социальной, экономической и политической жизни, быстрое изменение динамики процессов во всех сферах жизнедеятельности человека привели к росту потребностей *в знаниях*, т. е. к росту научно-технической информации. Быстрыми темпами расширяется познание человеком окружающего нас мира. Чуть ли не ежедневно мы становимся свидетелями все новых открытий, синтеза новейших материалов, совершеннейшей уникальной техники и технологий в промышленности. Очень быстрыми темпами идут освоение космоса, развитие микроэлектроники, внедрение нанотехнологий.

Велики потоки информации в производстве. Для управления любым экономическим объектом необходимо располагать и манипулировать определенными сведениями о его фактическом и желаемом состояниях. Ежегодно выпускаются миллионы наименований и типоразмеров продукции, с которой связаны огромные объемы конструкторской, технологической, потребительской и другой информации. Чтобы представить себе эти объемы информации, приведем пример. К концу 80-х гг. в СССР ассортимент выпускаемых в народном хозяйстве изделий насчитывал около 20 млн. наименований и типоразмеров. Среди производителей было 44 тыс. промышленных предприятий, 32 тыс. строительных организаций, 47 тыс. сельхозпредприятий. Легко подсчитать общее число межхозяйст-

венных связей, требующих полной и достоверной информации. Разрабатываются тысячи комплектов конструкторской и технологической документации. Составляются миллиарды различных нормативных, учетных документов, планов, прогнозов, финансово-бухгалтерских документов, отчетов, калькуляций, расчетных ведомостей, накладных на отпуск товаров, счетов на оплату.

Огромны потоки статистической информации о развитии народного хозяйства для координации всех отраслей промышленности и изучения развития отдельных регионов страны. Добавьте общественно-политическую и рекламную информацию на телевидении, радио, в печатных изданиях, информацию в сфере искусства, художественную литературу и получите безбрежный информационный океан.

При резком увеличении коммуникабельности людей информационные потоки растут лавинообразно. Если общая сумма человеческих знаний еще совсем недавно, в начале XX в., удваивалась каждые 50 лет, то к 1950 г. удвоение происходило уже каждые 10 лет, к 1970 г. – каждые пять лет, а к началу XXI в. – каждые 1,5–2 года.

Сегодня речь идет уже о так называемом *информационном взрыве (кризисе перепроизводства информации)*, негативными чертами которого являются:

- избыточность информации, затрудняющая восприятие потребителем полезных, необходимых ему данных;
- противоречия между ограниченными информационными возможностями человека и мощными потоками информации;
- возникновение экономических, политических и других социальных барьеров (например, по причине секретности), препятствующих распространению информации.

Если до середины XX в. основными предметами труда были материальные объекты, а экономическая мощь государства определялась его материальными, энергетическими, финансовыми, природными, трудовыми и другими ресурсами, то сейчас появилось понятие *информационных ресурсов* государства. Это научные теории, открытия, изобретения, сведения о недрах земли и океанов, проекты новейшего оборудования и приборов, математические модели экономики и человеческого общества и многое другое, в том числе программное обеспечение компьютеров. *Это то, что определяет интеллектуальную мощь современного государства. Это знания предыдущих и настоящего поколений людей.* В отличие от других ресурсов, например природных, информационные ресурсы со временем не убывают, а наоборот, накапливаются все в больших объемах. Еще раз подчеркнем, что сейчас именно **информационные ресурсы становятся движущей силой общества.**

Общество становится все более информационным. Информационное общество – это общество, структуры, техническая база и человеческий потенциал которого приспособлены для оптимального превращения знаний в информационный ресурс и переработки последнего с целью перевода пассивных форм его (книги, журналы, технические отчеты о выполненных научно-исследовательских работах, базы данных) в активные формы (модели, алгоритмы,

проекты, программы). Движущей силой развития общества должно стать производство информационного, а не материального продукта. Материальный же продукт станет более информационно емким, т. е. в его стоимости произойдет увеличение доли инноваций, дизайна и маркетинга. По сравнению с индустриальным обществом, где все направлено на производство и потребление товаров, в информационном обществе производятся и потребляются интеллект, знания, что приводит к увеличению доли умственного труда. Деятельность людей сосредоточивается главным образом на сборе и обработке информации, а материальное производство и производство энергии все больше переключаются на роботов, автоматизированные линии и системы, станки и машины с числовым программным управлением. Подтверждение этому – отчет Бюро трудовой статистики США об уменьшении численности рабочих на сборочных линиях в станкостроительной отрасли с 330 тыс. человек в 1992 г. до 14 тыс. человек в 2005 г. без уменьшения объема производства. Уже к середине 1990-х гг. около 60 % экономически активного населения США было занято в отраслях, производящих информационные технологии, либо активно использующих их в своей работе. Добавим, что за те же годы производительность труда в США за счет научно-технического прогресса (а информатизация – его главная движущая сила) в целом выросла на 37 %.

Влияние информатизации на производственные отношения и характер труда. Информатика оказывает очень сильное воздействие на экономическую, социальную, научно-техническую и культурную сферы жизни современного общества, что влечет за собой радикальные изменения не только в сфере производства и деловой активности людей, но и во всей социальной сфере. Благодаря информатизации реализуется принцип дальнего действия, осуществляется планирование программ целевого управления на основе эффективного использования прошлого опыта и высокоорганизованных знаний. Внедрение информационных технологий, как правило, ведет к изменению промышленных технологий, переоснащению производства, созданию и внедрению новых видов энергии, созданию и распространению новых видов продукции и даже к пере профилированию отдельных производств. Результатом утверждения нового технологического уклада является обесценивание старых технологий и продуктов их применения. Перспективной становится наукоемкая информационно-технологическая промышленная активность.

В США, например, влияние информационных технологий на сферу труда можно назвать поистине революционным. Здесь имеется в виду широкое использование персональных компьютеров и компьютерного оборудования, контролирующего работу производственных систем и, прежде всего, робототехники. Использование компьютеров, управляющих товаро-материальными запасами позволило радикально трансформировать систему материально-технического снабжения предприятий, установив надежные связи между поставщиками и оптовыми покупателями продукции в режиме on-line. Широкое применение компьютерных систем проектирования и дизайна, многократно увеличивающих производительность инженерного труда, компьютеров и телекоммуникационных технологий на

финансовых рынках и в банковской сфере, обеспечивает все виды финансовых транзакций без участия огромного числа банковских служащих. Число пользователей Интернета в США составляет более 70 %, населения, а объем продаж через «глобальную паутину» еще в 2002 г. превысил триллион долларов (это в три с лишним раза больше объема продаж всей автомобильной промышленности США). В России же в настоящее время аудитория Интернета занимает (8-е место в мире)[♦] и составляет около 30 % населения.

В информационном обществе изменяется не только производство, но и весь уклад жизни, возрастает значимость культурного досуга по отношению к материальным ценностям, появляются большие возможности духовного и культурного развития личности. Меняются характер и принципы организации труда. Прежняя привязанность к рабочему месту во многих случаях становится не обязательной, равно как и жесткий временной режим труда. На смену им приходят работа на дому, графики гибкого рабочего времени. Зачем, например, программисту или специалисту по рекламе идти на фирму и сидеть там за компьютером, разрабатывая программы и рекламные ролики, если это можно делать и за домашним компьютером, а с фирмой при необходимости связываться по сети? По данным социологического исследования, проведенного в США, уже сейчас более 30 млн. работающих осуществляют свою деятельность, не выходя из дома.

Новые режимы труда создают немало преимуществ для работников: появляется возможность более гибко планировать свое рабочее и личное время, уделять больше внимания самообразованию, а также семье и воспитанию детей, избегать часов пик на транспорте.

Информатизация привела не только к появлению массы новых профессий (операторы локальных сетей, создатели периферийного оборудования, программисты, компьютерные дизайнеры, рекламные агенты), но и повлияла на абсолютное большинство традиционных профессий. Без компьютера уже давно не обходятся инженеры и ученые, финансовые аналитики и банковские служащие, врачи и адвокаты, транспортные и торговые агенты. Автомобильные механики сегодня вряд ли найдут работу, если они не в состоянии провести компьютерную диагностику машины. Это же касается подавляющего большинства специалистов по ремонту и обслуживанию бытовой техники.

Вместе с тем, и это отмечают многие американские исследователи, подобная организационная и временная гибкость открывает возможности для усиления эксплуатации работников, пренебрежения их законными интересами, поскольку позволяет, пользуясь их изолированностью и отрывом от коллег и руководства, увеличивать сверхурочное время, ограничивать в информации, в частности о возможностях служебного роста и повышения заработной платы. Это требует принятия соответствующих законов, защищающих права трудящихся в новых условиях.

Новые технологии могут отрицательно влиять и на другие права и интересы не только конкретных работников, но и общества в целом. *Особую тревогу*

[♦] Подсчет аудитории в российском секторе Интернет можно посмотреть по адресу: <http://www.spylog.ru>.

вызывает угроза вмешательства в их частную жизнь. По данным Американской ассоциации управления, 45 % американских компаний прослушивают и записывают телефонные разговоры своих сотрудников, просматривают их электронную почту и компьютерные файлы. Это же происходит и в России. Информация стала товаром, а конкуренция сделала ее дорогим товаром. Например, такие сведения, как технические новинки фирмы, новые разработки, сведения о финансовом положении, о готовящихся сделках пользуются на черном рынке большим спросом. Свободно можно получить данные о том, какие финансовые или имущественные ресурсы имеются у того или иного гражданина, так как свободно можно купить базы данных ГИБДД, налоговой службы, службы регистрации недвижимости, банков. Такая информация может интересовать конкурентов и преступные сообщества. Поэтому руководители фирм вынуждены прибегать к тотальной слежке за своими подчиненными.

Таким образом, информатизация порождает целый ряд во многом новых проблем, касающихся как занятости и квалификации работников, так и их законных прав и интересов, требующих адекватной реакции общества.

Следует отметить и гигантское развитие **средств массовой информации**. Используя современные научно-технические разработки, они перешли на качественно новый уровень. Объединенные средствами связи в мировые информационно-коммуникационные сети, *они оказывают необыкновенно сильное влияние на психологию громадной массы людей во всем мире*. С помощью средств массовой информации возможны манипулирование общественным мнением, создание необходимых психологических предпосылок для формирования политических решений и целенаправленных воздействий на общество. От их воздействия в значительной степени зависят формирование общественно-политических движений и партий, социальная стабильность в обществе.

Влияние средств массовой информации особенно велико должно быть на нравственное изменение жизни молодых людей – жизни личной и общественной. В поведении многих молодых людей сегодня присутствуют, черствость, безответственность, безнравственность, хищническое отношение к природе, погоня за деньгами, за удовольствиями и т.д. Поэтому средства массовой информации должны воспитывать в молодом поколении трудолюбие, заботу о сохранении окружающей среды, противостояние алкоголизму, наркомании, преступности, развращенности. Все эти бедствия коренятся в человеческой душе. Исцеление человеческой души – это и оздоровление экологии и экономики и развитие у молодежи истинных нравственных ценностей и т.д. Это оздоровление общества в целом.

Сегодня практически каждый человек, так или иначе, связан с обработкой информации. Гибкая и оперативная обработка информации в таких огромных объемах уже не может осуществляться вручную, с помощью простейших технических средств. Необходим процесс *индустриализации* обработки информации с использованием современной компьютерной техники и систем передачи данных (как это было с развитием промышленного производства).

Понятие об информатике и информационных технологиях. Рынок информационных продуктов и услуг. Индустрия информатики

Для получения своевременной и достоверной информации небывало высокими темпами развиваются интегральные информационно-вычислительные сети связи, новые информационные технологии, появилась *информатика*, ставшая новой научной дисциплиной, изучающей информационные процессы во всех сферах человеческого общества. Как наука информатика изучает общие закономерности, свойственные информационным процессам. Когда разрабатываются новые носители информации, каналы передачи ее, приемы кодирования, визуального отображения информации и многое другое, то конкретная природа информации почти не имеет значения. Для разработчика системы управления базами данных (СУБД) важны общие принципы организации и эффективность поиска данных, а не то, какие конкретно данные будут заложены в базу многочисленными пользователями. Но информатика не только «чистая» наука. Важная ее особенность – широчайшие приложения, охватывающие почти все виды человеческой деятельности: производство, управление, образование, науку, проектные разработки, торговлю, финансовую сферу, медицину, охрану окружающей среды, социальную сферу и т. д.

Информатика является фундаментом процесса информатизации общества. Основная задача информатики – создание материально-технической базы для удовлетворения информационных потребностей промышленной и деловой сфер, органов государственного управления и других областей человеческой деятельности.

Что такое современная информатика, определил Международный конгресс по информатике в 1978 г. Он отметил, что понятие информатики охватывает области, связанные с разработкой, созданием, использованием и материально-техническим обслуживанием систем обработки информации, включая ЭВМ, оборудование связи, математическое обеспечение, организационные аспекты, а также комплекс промышленного, коммерческого, административного и социального воздействия.

Информатика одновременно молодая и одна из старейших наук. Бурное развитие ее произошло в последние 30 - 40 лет, но корни уходят далеко вглубь веков. Обращаясь к истории проблемы, один из «отцов» вычислительной техники, академик В.М. Глушков писал: «Задача накопления, обработки и распространения информации стояла перед человечеством на всех этапах его развития. В течение долгого времени основными инструментами для ее решения были мозг, язык и слух человека. Первое кардинальное изменение произошло в связи с при-

ходом письменности, а затем – с изобретением книгопечатания. Эти два этапа создали принципиально новую технологию накопления и распространения информации, избавившую человечество от необходимости всецело полагаться на такой зыбкий и ненадежный инструмент, каким является человеческая память».

Но настало новое время, и возникли качественно новые технологии. Подобно тому, как изобретение механического, а затем электрического двигателя открыло эру комплексной механизации и автоматизации физического труда, изобретение ЭВМ сделало то же самое в отношении труда умственного. Именно с появлением первых ЭВМ принято связывать возникновение информатики как науки. На базе ЭВМ и телекоммуникационной техники рождаются *информационные системы*, реализующие специальные информационные технологии для всех информационных процессов. Любая технология включает комплекс научных и инженерных знаний, воплощенных в приемах труда, наборах материальных, технических, энергетических, трудовых факторов производства, способов их соединения для создания продукта или услуги, отвечающих определенным требованиям или стандартам, т. е. технология представляет собой *совокупность последовательно выполняемых этапов преобразования исходного сырья в готовый продукт*. Разные технологии позволяют из одного и того же сырья получать различные виды продуктов. Например, используя разные технологии, из молока можно получить сметану и сливки, сыр и творог, кисломолочные продукты и сливочное масло. Аналогично и с информационными технологиями.

*Информационная технология представляет собой процесс использования технических средств, научных методов и правил сбора, накопления, хранения, обработки и передачи данных для получения **качественно новой** (вторичной) информации или информационного продукта с целью его воздействия на объект.*

В зависимости от того, какими средствами, методами, приемами информация передается, накапливается, обрабатывается, представляется пользователю, и определяется информационная технология. В отличие от производственных, энергопреобразующих технологий, информационные технологии относятся к социальным, знакопреобразующим технологиям.

Переход к информационным технологиям как *технической базе автоматизированных информационно-управляющих систем* выдвинул на передний план общие проблемы содержания информационных процессов и значения информационных технологий. Поэтому страны, использующие высокотехнологичные производства и информационные технологии, закладывают фундамент развития всей своей экономики на будущее.

Информационные технологии развиваются в несколько раз быстрее любых других технологий. В странах Западной Европы, США и Японии информационные технологии – главная инфраструктура, способствующая росту экономики, создающая новые рабочие места. В этих странах отрасли, взявшие на вооружение информационные технологии, значительно увеличили производительность труда и стали более конкурентоспособными. Информационные технологии способствуют уменьшению накладных расходов и снижают порог вхождения в общемировой рынок. Снижение расходов очень важно, но еще важнее то, что ин-

формационные технологии дают людям возможность, избавившись от рутинной работы, генерировать новые идеи и претворять их в жизнь.

Информационные технологии породили *объект* информатики – *информационные системы*, что является основным содержанием современной научно-технической революции. Информационные технологии выступают в качестве нового средства превращения знаний в информационный ресурс общества, стали средством его эффективного использования. Информационный ресурс является основным ресурсом человечества, главной ценностью современной цивилизации. *Предметом* информатики как новой фундаментальной науки выступает именно *информационный ресурс* как симбиоз знания и информации – его сущность, законы функционирования, механизмы взаимодействия с другими ресурсами общества и воздействия на социальный прогресс. Информатика как наука о законах получения, передачи и использования информационного ресурса в общественной практике подводит теоретический фундамент под использование ЭВМ и автоматизированных систем, которые и предназначены для усиления информационных процессов в обществе, использования информационного ресурса. Речь идет, прежде всего, о специальных информационных ресурсах, основанных на компьютерной технике и реализующих информационные технологии, то есть инженерную обработку знаний. Без ЭВМ нет информатики, но нельзя объявлять информатику наукой об ЭВМ. «Оттолкнувшись от ЭВМ», информатика во главу угла ставит новые понятия – информационный ресурс и его социальную полезность, отдачу.

На сегодняшний день информатика представляет собой комплексную научно-техническую дисциплину. Информатика под своим названием объединяет довольно обширный комплекс наук, каждая из которых занимается изучением одного из аспектов понятия «информация». Информатика – научная дисциплина, поэтому она тесно связана с математикой (теория вероятностей, теория множеств, математическая логика), с теорией алгоритмов и абстрактных машин, с лингвистикой и формальной логикой, с кибернетикой и теорией автоматического управления, с теорией познания и моделирования процессов окружающего нас мира, с теорией ЭВМ и программирования, радиосвязью, теорией кодирования, теорией формальных языков и грамматик и др. Она дает мощный импульс их развитию и на базе этого бурно развивается сама. Информатика превратилась в особую отрасль народного хозяйства, науки, своеобразное и важное подразделение экономики промышленно развитых стран. В широком смысле слова современная информатика – это раздел фундаментальной естественной науки, изучающей законы и методы производства, переработки, хранения и распространения всех видов информации в человеческом обществе, природе и технических устройствах.

Само слово «информатика» возникло в начале 1960-х гг. во французском языке для обозначения области *автоматизированной переработки* информации в обществе. Именно от слов «информация» и «автоматика» оно и произошло путем слияния. Практически одновременно с французским в 1963 г. в журнале «Известия вузов. Электромеханика» был предложен и русский вариант. Со временем

этот термин появился во всех языках мира, кроме, может быть, английского, в котором он звучит как «computer science» – вычислительная наука.

В настоящее время информатика – это наука, техника и технологии использования информационного ресурса, являющегося, как указывалось, *предметом* информатики, это организационные формы в обработке, хранении и передаче информации, это огромный рынок информационных технологий и услуг.

Нельзя смешивать информатику с компьютеризацией и под термином «информатика» понимать вычислительную технику и программирование. В отличие от *компьютеризации*, где основное внимание уделяется созданию и внедрению компьютерной техники во все сферы социально-экономической и духовной деятельности, *информатика* основное внимание уделяет комплексу мер, направленных на получение и полное использование достоверного, исчерпывающего и своевременного *знания* во всех сферах человеческой деятельности.

Хотя информатика еще совсем молодая наука, тем не менее, в ней четко прослеживается грань между двумя составляющими: теоретической и прикладной информатикой, аналогично другим наукам, например физике.

Теоретическая информатика рассматривает все вопросы разработки автоматизированных информационных систем: их проектирования, создания и использования не только с формально-технической, но и содержательной стороны, а также комплекс экономического, политического и культурного воздействия их на социальную динамику. В орбиту анализа теоретической информатики попадают и традиционные системы преобразования информации и распространения знаний: средства и системы массовой информации, система лекционной пропаганды, кино, театры, справочные службы и т. д. Но теоретическая информатика рассматривает их с определенной стороны – с позиций получения и использования информационного ресурса, форм и способов воздействия указанных систем на общественный прогресс, возможной их технологизации.

Теоретическая информатика изучает информационный ресурс, законы его функционирования и использования как движущей силы социального прогресса, а также общие, фундаментальные проблемы информационных технологий как исторического феномена, выводящего общество на новую ступень развития. Решающее значение для рождения теоретической информатики имеет появление информационных технологий высшего уровня, основанных на искусственном интеллекте.

Прикладная информатика изучает *конкретные разновидности* информационных технологий, которые формируются с помощью специальных информационных систем (управляющих, обучающих, криминалистических, медицинских и т. д.). Очевидно, что такие информационные технологии, как, например, управление (АСУП, АСУТП), проектные разработки (САПР) или криминалистика, имея общие черты, в то же время существенно различаются между собой. Они имеют разные операции и процедуры, различное оборудование, специализацию критериев и показателей, у них разная степень замкнутости информационных контуров, т. е. разные информационные среды, – все это становится объектом изучения конкретных функциональных и отраслевых информатик. Так рождают-

ся ветви прикладной информатики, обслуживающие создание различных систем и информатизацию разных сфер социальной и экономической практики.

Рынок информационных продуктов и услуг. Лавинообразные потоки информации к концу XX в. захлестнули мир. Океан информации закономерно порождает величайшие трудности для каждого, кто ею пользуется. Попробуйте найти в этом океане что-то конкретное, достаточно специфичное, нужное лично вам сейчас – вы сразу в этом убедитесь. *Получается, что нужной информации мало, потому что ее слишком много, – вот к какому парадоксу ведет информационный взрыв.* Чтобы помочь ученым и специалистам лучше справляться с могучими волнами океана информации, возникли научно-информационные службы. Это довольно многочисленная и сложная сеть различных научно-исследовательских институтов технико-экономической информации, областных и городских центров научно-технической информации. Это специализированные фонды, научно-технические библиотеки, специальные подразделения крупных промышленных, сбытовых, торговых, научных организаций по сбору, накоплению и распространению научно-технической информации. Появились различные информационно-коммерческие и рекламно-маркетинговые центры, центры аналитических исследований и научного прогнозирования. *Появился рынок информационных продуктов и услуг.* Информация, наряду с материальными продуктами, стала предметом купли-продажи, т. е. продуктом. Этот рынок огромен. По оценкам специалистов более 15 % общемировой торговли составляют информационные продукты. В первую очередь, это научно-техническая информация: патенты на открытия и изобретения, отчеты о новейших научно-исследовательских, проектно-технологических и опытно-конструкторских разработках, позволяющие создавать новые производства на базе *высоких технологий*, производить новые виды промышленных продуктов, энергии и многое другое.

Не менее важны информационные продукты и услуги в сфере экономики, поддерживающие принятие как оперативных, так и стратегических решений. Прежде всего, это финансовая, банковская, биржевая, коммерческая и другая информация. Например, сведения об экономическом потенциале того или иного региона позволяют планомерно развивать внешнеэкономическую деятельность предприятия. Сведения о районах с высоким уровнем доходов населения или с высокой концентрацией покупателей позволяют поставщику продукции выбрать наилучшее расположение дилерских точек. Сведения о спросе и конкурентоспособности выпускаемой предприятием продукции позволяют оценить насколько обеспечен рынок сбыта. Наконец, это и большой рынок программных продуктов для обработки разнообразной информации на компьютерах.

Информационные службы накапливают научно-техническую информацию по отраслям науки и техники, готовят специализированные аналитические и библиографические издания, распространяют реферативные журналы, каталоги, рекламные и обзорные выпуски, отражающие поток оригинальной (первичной) научной информации, ведут картотеки, готовят выпуски экспресс-информации, осуществляют переводы научно-технической литературы и доку-

ментации с целью повысить эффективность научных исследований и разработок.

Произошло выделение научно-информационной деятельности в *особый вид труда*. Соответственно этому сложилась особая область знания – *теория научной информации и научно-информационной деятельности*, изучающая *метаинформацию* (информацию о научно-технической информации: структуру, ее свойства и др.), т. е. организацию и развитие *вторичной* научно-технической информации и способы автоматизации ее обработки.

Львиную долю в обработке вторичной информации занимает поиск информации. Для автоматизированного быстрого поиска информации создаются *информационно-поисковые системы*, которые строятся на базе специальных информационно-поисковых языков и мощных высокопроизводительных компьютеров.

Индустрия информатики. Информационная революция наших дней выдвигает на первый план новую отрасль народного хозяйства – *информационную индустрию*, выпускающую технические средства (прежде всего, компьютерную технику и средства связи), программное и информационное обеспечение и разрабатывающую методы и технологии для получения новых знаний.

Рассматривая индустрию информатики, следует выделить ее составляющие.

- Наличие и дальнейшее развитие всех звеньев, непосредственно реализующих переработку информации: вычислительные центры (ВЦ) и их сети, исследовательские и проектные институты по созданию информационных систем, организации по обслуживанию ВЦ, пусконаладочные организации и их объединения, обслуживающие ВЦ и АСУ и т. д. Все это превращается в особую инфраструктуру, обслуживающую сферу народного хозяйства, требующую единой технической политики и организационного оформления.

- Создание новых производств и целых промышленных отраслей по выпуску средств вычислительной техники и автоматизации информационных процессов: ЭВМ, периферийного оборудования, элементной базы, микроэлектронной техники, контроллеров, дисковых подсистем, принтеров, сканеров, множительной техники, других электронных и электромеханических узлов и блоков, магнитных носителей информации, эксплуатационных материалов. Создание производств по выпуску средств связи и телекоммуникаций: коммутаторов, маршрутизаторов, линейных усилителей, межсетевых шлюзов, брандмауэров различного типа кабелей и многого другого.

- Создание предприятий по выпуску вспомогательного оборудования (например, кондиционеров, вентиляционного оборудования) и материалов, специальной мебели для ВЦ, средств магнитной записи, оборудования по выработке бумажных носителей и пр.

- Создание организаций и предприятий по разработке математического и программного обеспечения (поскольку создание программных средств переводится на промышленную основу), по накоплению программных ресурсов (фонды алгоритмов и программ), организаций по массовому тиражированию программ общего назначения.

- Разработку и создание систем и сетей связи и телекоммуникаций, без которых немислим машинно-информационный процесс.

Рост индустрии информатики резко опережает рост других отраслей народного хозяйства. В большинстве ведущих промышленно - развитых стран по числу занятых в этой отрасли, объемам выпускаемой продукции она уже выдвинулась на первое место среди всех других отраслей. Этой исключительно динамичной, технически развитой, наукоемкой сфере народного хозяйства придается стратегическое значение. Эта сфера деятельности уже превратилась в область массового приложения труда. Рождаются новые машины, создаются уникальные программы, растет машинный интеллект – появляются новые возможности для исследования и познания окружающего нас мира. Новые информационные технологии с применением компьютеров позволяют не только накапливать, хранить, обрабатывать имеющуюся информацию, но и получать новую информацию, новые знания. *В получении новых сведений, новых данных, количественно и качественно отличающихся от исходных, вводимых в компьютер, и состоит сущность толкования компьютера как усилителя, а если учесть его быстроедействие, то и ускорителя интеллекта.*

Важная особенность индустрии информатики состоит в том, что она является не монолитной, а *инфраструктурной* отраслью (как транспорт и связь), обслуживающей другие отрасли материального производства и непродуцированной сферы, создавая условия для их развития.

Мы будем понимать под информатикой быстро развивающуюся научную дисциплину, в которой объединены соответствующие разделы математики, физики, кибернетики, электроники и связи, необходимые для исследования и решения задач получения, накопления, хранения, обработки и передачи информации с помощью *компьютера*.

Лекция 3

Связь между информацией, сообщением и сигналом. Виды сигналов и их особенности

Термин *информация* произошел от латинского слова *informatio*, что означает «сведения, разъяснения или изложение». Слово «информация» во множественном числе так и произносится – сведения. Действительно, в быту это понимается как передача *сведений* между людьми. В теории связи под информацией принято понимать *любую последовательность передаваемых символов*, не учитывая их смысл. В ячейках памяти компьютера под любой информацией понимается совокупность *записанных двух знаков 0 и 1*. В неживой природе понятие информации связывают с понятием *отражения, отображения* каких-то явлений, процессов, объектов. Кратко рассмотрим на следующих примерах значение слова «информация»:

- в автоматизированной системе управления технологическим процессом объект характеризуется рядом параметров, значения которых заранее неизвест-

ны и непостоянны во времени. С помощью датчиков, измерительных и регистрирующих приборов непрерывно или периодически производятся измерения этих параметров – собирается *первичная информация* о состоянии объекта для передачи в управляющее устройство, которое на основе выбранных критериев и алгоритмов обрабатывает полученные данные и передает *новую (вторичную) информацию* – команды, воздействующие на объект управления через исполнительные органы;

- радиоизлучение звезд – рассеянные по Галактике шумы, являющиеся для линий космической связи помехами в передаче информации, но для ученого за радиотелескопом это *полезная информация* о звездах и развитии Вселенной;

- ученый-биолог, за микроскопом изучая растение, *получает информацию* об особенностях жизни и его развития.

Примеров приводить можно бесконечно много. Они позволяют сделать обобщение и дать для обиходной речи такое нестрогое определение информации.

Информация – это совокупность сведений о всевозможных объектах, явлениях и процессах окружающего нас мира, это отражение реального мира с помощью человеческой речи, знаков и сигналов.

С практической точки зрения информация – это нечто абстрактное. *Абстрактная информация всегда представляется в виде конкретного сообщения.* Можно сказать, что сообщение выступает в качестве *материальной оболочки* для представления информации и ее передачи. Сообщение служит переносчиком информации, а информация – содержанием сообщения.

Эти два понятия являются основными в информатике. Информационное сообщение от источника сообщения поступает на кодирующее устройство, передается по каналу связи, декодируется и поступает получателю сообщения. Научные значения этих понятий несколько отличаются от понимания в обиходной речи и связаны с теорией вероятности[♦].

Соответствие между сообщением и информацией не является взаимно однозначным. Одна и та же информация может быть передана с помощью различных сообщений. Например, общеизвестно, что о начале войны с Германией Сталину поступали самые различные сообщения от разведчиков и дипломатов, от торговых представителей и простых граждан, но все они несли одну и ту же информацию: летом 1941 г. Германия нападет на СССР. И наоборот, одно и то же сообщение может нести различную информацию. Например, сообщение по радио или телевидению о падении самолета с пассажирами на борту несет совершенно разную информацию для обывателя, руководства авиакомпании и родственников погибших. Все зависит от того, как сообщение *интерпретируется* получателем, как получатель понимает и соотносит полученное сообщение применительно к себе.

Процесс интерпретации может осуществляться человеком или группой лиц, биологическими объектами (например, рефлекс собаки) или техническими устройствами в автоматизированных системах управления и, конечно, в ком-

[♦] Вся теория информации построена на использовании теории вероятности, которая выходит за рамки нашего изложения и изучается в математике.

пьютере. Он зависит от многих факторов: что за получатель интерпретирует сообщение, какой информацией он уже располагает, с каких позиций рассматривает сообщение и т. д. Процесс интерпретации в общем случае не может быть определен формально, в то время как сообщение существует всегда в какой-то определенной форме.




Человек воспринимает сообщения посредством органов чувств. Приемники информации в технике воспринимают сообщения с помощью различной измерительной и регистрирующей аппаратуры. В обоих случаях с приемом информации связано изменение во времени какой-либо величины, характеризующей состояние приемника. Сообщение от источника к приемнику передается в материально-энергетической форме, виде сигналов электрических, световых, звуковых и т. д.

В этом и заключается абстрактность информации в отличие от конкретности сообщения.

Приведем примеры, которые показывают, насколько разнообразна связь между сообщением и информацией.

- «Поздно не приду» / «Поздно, не приду» – пример того, как дополнительный знак препинания в сообщении существенно изменяет передаваемую информацию.

- Начиная с третьего слова справа, запишем первые буквы (они подчеркнуты) слов следующего сообщения: «такой анализ говорит о наличии практически ежедневных нагрузок». Получим слово ПЕНТАГОН – пример того, как в открытом сообщении содержится скрытая информация.

- Сообщения службы дорожного движения в виде следующих графических знаков , ,  на языке правил дорожного движения несут информацию для водителей о повороте, особом внимании на участке дороги и запрете движения.

Для существования информации обязателен какой-то материальный объект (физическая среда), хранящий или передающий ее. Его будем называть *носителем информации*. В качестве носителей информации могут быть объекты любой физической природы. Кроме бумаги современная техника в качестве носителей при передаче информации чаще всего использует:

- механическое движение;
- электрические заряды, напряжение и ток;
- электромагнитные волны, в том числе и световые;
- волны движения в жидкостях и газах, включая звуковые волны.

В качестве долговременных носителей информации при хранении, помимо бумаги, наиболее часто используются намагниченные пленки и покрытия, светочувствительные фотопленки и киноплёнки, а также специальные немагнитные полимерные покрытия, обрабатываемые лучом лазера.

Информация на носителе существует в виде сигнала.

Сигнал – это материальное воплощение сообщения в виде, используемом в информационных системах при передаче, обработке и хранении информации. При передаче и обработке информации сигнал представляет собой процесс из-

менения во времени и пространстве некоторой физической величины (например, импульсы тока), характеризующей состояние информационной системы. Иногда такие сигналы называют *динамическими*. В системах хранения информации сигналы преобразуются в *состояния* некоторого материального носителя (записи в виде отверстий на перфоленте, намагниченностей магнитных материалов, изображений на фотопленках и кинопленках) и т. д. Эти сигналы иногда называют *статическими*.

Та характеристика сигнала, которая служит для представления информации, называется параметром сигнала. Для примера возьмем радио. Передатчик, входящий в состав радиопередающего устройства, вырабатывает ток высокой частоты и питает им передающую антенну (несущая частота).

Управляющее устройство изменяет амплитуду или частоту этого тока в соответствии с передаваемыми сигналами. Данный процесс называется *модуляцией*, а получающийся при этом высокочастотный ток называется модулированным током. В качестве простейшего управляющего устройства может использоваться микрофон. Модулированный ток высокой частоты, действующий в передающей антенне, вызывает излучение электромагнитной энергии в окружающее пространство в виде радиоволн. Следовательно, сигналом являются электромагнитные волны. В диапазоне средних волн (СВ) сообщения передаются путем изменения амплитуды колебаний, а в диапазоне ультракоротких волн (УКВ) – путем изменения частоты колебаний (амплитудная и частотная модуляция). В первом случае на СВ параметром сигнала является амплитуда, а во втором случае на УКВ – частота колебаний.

Сигнал, воспроизводящий все мгновенные значения физической величины на отрезке времени его существования, называется *непрерывным, или аналоговым*. В общем случае можно сказать, что характеристики всех явлений реального мира описываются с помощью аналоговых сигналов (световое излучение, звук, давление, температура, электромагнитные волны и т. д.). Это сигналы, передаваемые по радио и телевидению, телефонные сигналы, многие сигналы систем телемеханики и автоматического управления, измерения, регулирования и другие. Поэтому на первых этапах развития вычислительной техники широкое применение находили не цифровые, а так называемые *аналоговые вычислительные машины* (АВМ), или моделирующие установки.

Если параметр сигнала может принимать лишь конечное число значений и существовать лишь в конечном числе моментов времени, то такой сигнал называется *дискретным*. Аналоговый сигнал, изменяясь во времени, принимает огромное множество *информационных* значений. В отличие от этого дискретный сигнал имеет только два *информационных* значения.

Вид такого сигнала представляет собой скачкообразное изменение во времени с одного уровня на другой, т. е. сигнал имеет *импульсную* форму. В настоящее время происходит бурное развитие импульсной техники и внедрение импульсных методов работы в радиолокацию, радиосвязь, радионавигацию, радиотелеметрию, радиоизмерения, телефонию, телевидение и т. д. Дискретные сигналы используются для представления любой информации в памяти компь-

ютеров, в компьютерных сетях для передачи сообщений, представляющих собой последовательности двоичных символов.

Как ни странно на первый взгляд, это отличие дает дискретным сигналам, по сравнению с непрерывными, большие преимущества. Аппаратура для приема, обработки и передачи дискретных сигналов значительно проще, чем для аналоговых, а значит, надежнее и дешевле. В каждом дискретном сигнале легче исправлять ошибки, неизбежно появляющиеся вследствие различных помех. Для дискретных сигналов разработаны специальные методы автоматического обнаружения и исправления ошибок при передаче и обработке информации за счет избыточно введенных сигналов в сообщение на случай пропадания одного или нескольких импульсов в сообщении. Так как дискретный сигнал имеет ступенчатую форму, т. е. принимает два значения, то незначительные изменения сигнала по величине можно игнорировать. При этом значение несущей им информации не изменяется. Дискретные устройства более компактны, потребляют значительно меньше электроэнергии, чем аналоговые.

Любой аналоговый сигнал может быть с той или иной степенью точности преобразован в дискретный путем квантования (деления) по уровню или по времени. Точно так же дискретные сигналы могут быть преобразованы в аналоговые. Для этого используются *аналого-цифровые и цифро-аналоговые преобразователи (АЦП и ЦАП)*. Учитывая преимущества дискретных сигналов, любой физический процесс может быть представлен посредством дискретных сигналов либо смоделирован с их помощью.

В настоящее время во всех развитых странах мира бурно ведутся разработки бытовой и промышленной аппаратуры, передающей, обрабатывающей и принимающей дискретную информацию. Такая техника получила название *цифровой*. Создание цифровой промышленной и бытовой информационной техники открывает возможности для *единого* подхода к ее проектированию и изготовлению, а также созданию общих хранилищ информации больших объемов. В этих хранилищах могут храниться любые виды информации: телевизионные передачи, кинофильмы, тексты документов, чертежи, картины и рисунки, музыка и т. д. Обработка и представление этой информации на компьютерах производится одновременно. Поэтому сейчас бурно развиваются новые *мультимедийные* компьютерные технологии, позволяющие представлять пользователю на экране монитора одновременно движущиеся объемные картинки, сопровождаемые музыкой и пояснительным текстом.

Лекция 4

Общие процессы восприятия, сбора, передачи и представления информации человеком

Восприятие информации – это процесс преобразования сведений, поступающих в техническую систему или живой организм из внешнего мира, в форму, пригодную для дальнейшего использования. Благодаря восприятию информации обеспечивается связь системы с внешней средой.

Для развитых систем восприятия можно выделить несколько этапов переработки поступающей информации:

- предварительная обработка для приведения входных данных к стандартному для данной системы виду;
- выделение в поступающей информации смысловых и практически значимых фрагментов;
- распознавание объектов и ситуаций;
- коррекция внутренней модели мира.

Живые существа воспринимают информацию с помощью анализаторов.

Анализаторы в биологии – это сложные системы нервных клеток (нейронов), осуществляющие восприятие и анализ информации об окружающей организм среде (или внутри самого организма) и формирующие специфические для данного анализатора ощущения. Анализатор получает название по тому виду сенсорной информации, для восприятия которого он специально приспособлен. Раздражители, несущие нам информацию, бывают зрительные, слуховые, осязательные, вкусовые и обонятельные, а также сила земного тяготения. Каждый анализатор состоит из воспринимающего отдела – рецептора, проводниковой части и высших нервных центров в коре головного мозга.

Следовательно, познание и восприятие человеком окружающего мира начинается с восприятия его органами чувств: зрением, слухом, обонянием, вкусом, осязанием. Именно они доносят до сознания информацию о самых разнообразных свойствах предметов, явлений и процессов, окружающих человека. После того как физические раздражители (световые или звуковые волны, запахи, жар или холод и т. д.) преобразуются рецептором в нервные импульсы, они существуют в виде кода нервных импульсов в специфических сенсорных каналах нервной системы. Впоследствии мозг реконструирует образ события, складывая вместе всю информацию, получаемую в данный момент от каждого из активированных рецепторов. Эта совокупность информации и интерпретируется мозгом для создания той конструкции, которая называется восприятием события.

Мозг человека способен из всего обилия поступающей информации отобрать ту, которая существенна в данный момент, в данной ситуации. Эта информация превращается в признаки, позволяющие мозгу в дальнейшем распознавать объекты, процессы, явления, с которыми человеку приходится иметь дело. Поэтому свойство мозга отражать и познавать внешний мир предстает как звено в развитии процессов, связанных с передачей и обработкой информации. С этой точки зрения мозг человека представляет собой исключительно сложную кибернетическую систему, хранящую и обрабатывающую поступающую из внешнего мира самую разнообразную информацию.

Однако формирование знаний человека происходит не только на основе получаемой им информации от органов чувств. Овладение ядерной энергией, выход в открытый космос, синтез все новых и новых материалов и видов энергии, бурное развитие микроэлектроники и на ее основе компьютеров – все это лежит за пределами чувственного познания человека. Это продукт коллективной интеллектуальной деятельности. Наличие *предшествующей* информации

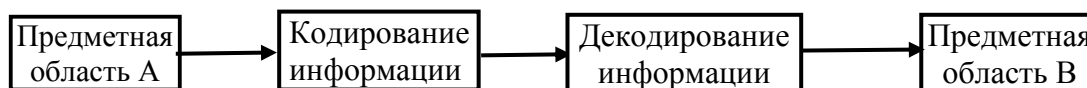
способствует более полному и достоверному восприятию *последующей* информации, что в познании имеет решающее значение. «Если я видел дальше других, то это потому, что стоял на плечах гигантов», – говорил Исаак Ньютон, имея в виду использование им ранее накопленных человечеством знаний.

Возможность передачи знаний от одного человека другому – вот в чем заключается прогресс человечества в целом и каждого его представителя. Но чтобы передавать знания, нужно как-то обозначать окружающие нас предметы, явления, процессы и их признаки.

Понятие о знаках. С появлением жизни на нашей планете появилось общение. Первобытные люди быстро пришли к осознанию необходимости накопления и фиксирования приобретенных еще незначительных знаний, чтобы воспользоваться ими в дальнейшем. Люди начинают делать рисунки на скалах, на песке, зарубки на дереве, складывать в кучки камешки, т. е. отражать и фиксировать полученную информацию при помощи *знаков*. Таким образом, не осознавая важность события, человечество шагнуло в мир информации.

Возможность передачи знаний одного человека другому появилась, видимо, именно тогда, когда первобытный человек научился каким-то образом обозначать предметы и их признаки, давать им имена и фиксировать свои сообщения, т. е. научился различать отдельные предметные области. Ведь и сейчас предмет, его признаки или явления «присутствуют» в человеческом мышлении только тогда, когда они как-то обозначены, – человек видит предмет, знает его имя, слышит о нем что-то и т. д.

Процесс передачи информации можно представить следующим образом:



Для того чтобы передаваемое сообщение было понятным, необходимо, чтобы:

- 1) предметная область А содержалась в предметной области В;
- 2) кодирование и декодирование информации были взаимнообратными операциями.

Это настолько важно в информатике, что запишем два вывода.

- Количество информации, получаемой из сообщения, зависит от имеющихся предварительных знаний (семантический аспект) предметной области.
- Чтобы понять сообщение, необходимо знать алфавит кода (язык), в котором это сообщение передано (синтаксический аспект).

Разберем следующие примеры. Пусть нарушен первый вывод: «Дефицит кокарбоксилазы повышает уровень пировиноградной кислоты в крови, что в результате приводит к ацидозу». А теперь скажите, если наступит ацидоз – это хорошо или плохо? А пировиноградная кислота содержится в винограде? Не приводит ли поедание винограда к ацидозу? Вы ответите на эти вопросы, если владеете предметной областью, иначе – увы!

Пусть нарушен второй вывод. Вы получили телеграмму на китайском языке. Если вы его не знаете, то сообщение для вас будет всего лишь последовательностью символов, не несущих никакой информации.

Понятие знака занимает важное место в информатике, логике, лингвистике, психологии, философии и других разделах современной науки и человеческого знания. *Важнейшее качество знака – его ассоциативность.* Чем оно однозначнее, тем проще его восприятие. Возьмем, к примеру, буквы. Каждая из них воспринимается однозначно, чего нельзя сказать о жестах. Для понимания природы знака первостепенное значение имеет выделение особых социальных (знаковых) ситуаций, в которых происходит его использование. *Новые знаки появляются тогда, когда назревает социальная необходимость в них.* Одну из таких ситуаций мы назвали – это попытки первобытного человека осознать и как-то обозначить явления, процессы, предметы и их признаки в окружающем его мире. Другой пример. С появлением автотранспорта и стремительным ростом его числа, увеличением интенсивности движения на дорогах появилась необходимость в регулировании движения с целью его безопасности, которая вызвала к жизни знаки дорожного движения. Аналогичные ситуации возникали при развитии химии, математики, физики и т. д., когда человеку необходимо было как-то зафиксировать математические действия или химические реакции – появлялись системы специальных знаков. Для записи музыки и шахматных партий возникли нотная грамота и специальные знаки для записи шахматных партий.

Знак – минимальная, неделимая единица информации, выражающая определенное суждение о предмете, явлении или смысл чего-то.

Использование знаков неразрывно связано с передачей информации.

Важнейшее свойство знака – обозначать что-либо. В силу этого невозможно понимание знака без его значения: предметного (обозначаемый им предмет), смыслового (образ обозначенного им предмета) и экспрессивного (выражение с его помощью чувств). В понятие знака входят не только *фонемы и графемы*, т. е. звуки, буквы, знаки препинания, цифры, знаки операций и т. д. Это понятие значительно шире: кивок («да») или покачивание («нет») головой тоже знаки. На определенном уровне слова тоже могут быть знаками, например в стенографии, и даже целые предложения могут представлять собой знаки, например, при переводе с одного языка на другой.

Человек вскоре осознал, что окружающий мир безбрежен и необходимых знаков может быть бесчисленное множество и разнообразие. Все их человек просто не в состоянии запомнить и воспринять. Поэтому он начал создавать из них новые конструкции – слова. Появилась идея выделить некоторое ограниченное число простых знаков, из которых путем соединения их при соблюдении определенных правил получать произвольные слова, последовательности слов и т. д. Таким образом, возник *алфавит*. Важно отметить, что человечеству понадобилось не так уж много различных знаков-символов, чтобы написать миллионы книг самого разного содержания. Возникновение слов привело к появлению предложений, т. е. высказыванию мыслей. Это позволило большим массам людей общаться друг с другом, делиться знаниями. *Набор знаков, в*

котором определен их линейный порядок, есть алфавит. Кроме известных нам алфавитов языков народов мира, алфавитом являются, например, 12 знаков Зодиака, 10 арабских цифр для записи любых чисел (числовых слов), нотная грамота, знаки математических действий, двоичный алфавит для представления любой информации в информационной технике и компьютерах и т. д.

Возникновение алфавита – очень заметная веха в развитии информационных процессов, к сожалению, оно породила немало проблем. Дело в том, что, когда люди создавали алфавит, каждая община, регион или страна вкладывали в его создание различные соображения. В итоге появилось огромное количество алфавитов по всей планете, что затруднило быстрое распространение знаний, информации, затормозило развитие человечества. Некоторые алфавиты не только непонятны людям, использующим другой (свой) алфавит, но и трудны для восприятия даже тем, кто изучает их с детства. Например, китайский алфавит, насчитывающий более 10 000 знаков. Причем эти знаки обозначают не одну букву, как принято в большинстве алфавитов, а целые высказывания. Тем не менее, разные алфавиты обладают одинаковой «изобразительной силой», т. е. с помощью одного алфавита можно представить всю информацию, которая имеется в других алфавитах. А значит, информацию обо всем окружающем человека мире можно отобразить в алфавите, состоящем только из двух символов, например цифр 0 и 1.

Так осуществляется физическая запись любой информации в компьютере. В компьютерах на физическом уровне используется двоичный алфавит: цифры 1 и 0. Эти цифры соответствуют двум различным уровням напряжения или тока (потенциальный способ), либо наличию или отсутствию импульсов (импульсный способ). На рис. 1 высокий уровень напряжения или наличие импульса соответствует цифре 1, а противоположное положение – цифре 0. Цифры 1 и 0 могут различаться также потенциалами или импульсами противоположного знака. В схемах компьютера переменные величины и соответствующие им сигналы изменяются не непрерывно, а в дискретные моменты времени $t = 1, 2, \dots, i$.

Временной интервал между двумя соседними моментами дискретного времени называется *тактом*, или *периодом представления информации*. Отметка моментов дискретного времени осуществляется *тактовым генератором компьютера*, вырабатывающим *синхронизирующие импульсы (СИ)*.

Для представления и передачи двоичных машинных слов, состоящих из нескольких двоичных разрядов, применяют *последовательный* или *параллель-*

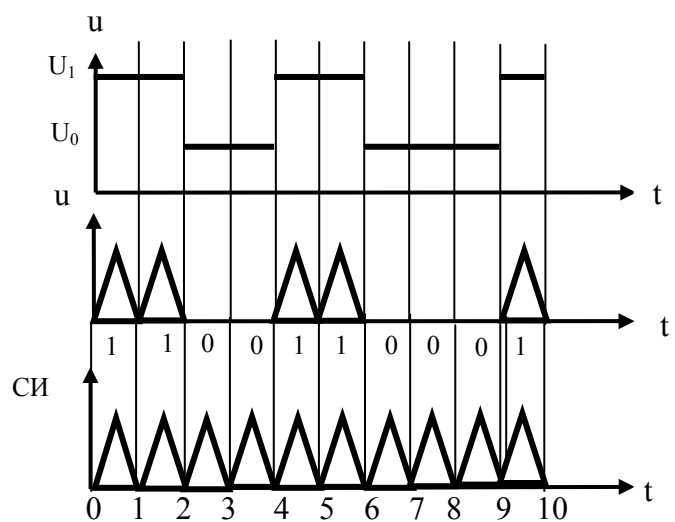


Рис. 1. Физическое представление информации в компьютере

ный способы. При последовательном способе каждый временной такт используется для отображения одного разряда слова, все разряды которого передаются последовательно и фиксируются одним и тем же элементом. Номер разряда определяется номером такта, который отсчитывается от нулевого положения, совпадающего с началом слова. Таким образом, двоичный код слова представляется в виде некоторой временной последовательности потенциальных или импульсных сигналов, соответствующих значениям цифр в разрядах слов. При параллельном способе все разряды двоичного кода слова передаются одновременно в одном такте, фиксируются отдельными элементами для каждого разряда и передаются по отдельным линиям связи, каждая из которых предназначена для представления и передачи только одного разряда слова. При этом код слова развертывается не во времени, а в пространстве.

Для достижения высокого быстродействия основные устройства современных компьютеров строятся параллельными. Хотя они требуют больше аппаратуры, так как при параллельном коде надо иметь столько шин, а также запоминающих и преобразующих элементов, сколько разрядов в слове, однако это не представляет трудности, так как современная микроэлектроника позволяет иметь несколько сложных функциональных схем на одном полупроводниковом кристалле в одном корпусе микросхемы.

На логическом уровне информация может представляться в других алфавитах. Следовательно, существует необходимость находить между разными алфавитами взаимное соответствие.

Использование алфавита для создания слов со временем стало подчиняться определенным правилам. Ими же было описано построение наборов слов – предложений. Возникла необходимость в дальнейшей систематизации этих структур. Это положило начало возникновению языка.

Понятие о языках. По сути, в любом алфавите можно составить сколь угодно много знаковых конструкций. Главное, чтобы слова несли определенный смысл, и чем он уже, тем однозначнее будет воспринято слово, тем яснее будет вложенный в него смысл. Таким образом, знаковые конструкции, составленные из определенного алфавита, обязаны подчиняться определенным правилам.

Множество всех слов и других конструкций, записываемых в данном алфавите согласно правилам, есть знаковая система, или язык.

Язык – это средство отражения и познания окружающего нас мира. Основными функциями любой знаковой системы являются:

- познавательная – передача сообщений или выражение смысла;
- общение – т. е. обеспечение понимания пользователем передаваемого сообщения;
- коммуникативная – побуждение его к действию.

Осуществление любой из них предполагает определенную внутреннюю организацию знаковой системы, т. е. наличие различных знаков и законов их сочетания. В соответствии с этим выделяются три основные группы правил:

1. *Семантика* изучает знаковые системы как средство выражения смысла. Правила *интерпретации* (выражения смысла) знаков и составленных из них

конструкций являются предметом рассмотрения семантики. Конструкция из символов алфавита называется *имеющей смысл*, если она поддается истолкованию с помощью правил интерпретации, принятых для данной знаковой системы.

2. *Синтаксис* изучает внутреннюю структуру знаковых систем относительно к выполняемым ими функциям. Правила построения различных конструкций из знаков алфавита, т. е. правильное написание их, и есть синтаксис знаковой системы, или языка.

3. *Прагматика* (от греческого прагма – дело, действие) определяет практическую значимость и ценность знаковых систем, т. е. значение их для практических приложений.

В настоящее время появилась новая дисциплина – *семиотика*, занимающаяся сравнительным изучением знаковых систем: от простейших систем сигнализации до естественных языков. Семиотические исследования способствуют *формализации* новых областей науки, что позволяет подходить с единых позиций к познанию и изучению различных явлений, процессов и объектов окружающего мира, подтверждая его материальное единство. Понятия и методы семиотики приобретают большое значение в связи с развитием теории и практики хранения и обработки информации.

Всякий информационный процесс может осуществляться только при наличии языка, описывающего объекты и связи между ними.

Языки бывают естественными и искусственными. *Естественный язык* – это язык повседневной жизни, служащий формой выражения мыслей и средством общения между людьми. Это языки устной и письменной речи народов мира. *Искусственный язык* создается людьми для каких-то узких потребностей. Это язык математических формул, язык записи шахматных партий, язык записи химических формул, нотная грамота, язык обозначений на электрических схемах, языки программирования высокого уровня, языки различных указателей, например знаков дорожного движения, и т. д.

Искусственный язык, предназначенный для записи информации в запоминающие устройства информационной машины или системы, называется *информационным*. Ни один из естественных языков не может быть применен в качестве информационного. Этому препятствуют недостаточно четкая и однозначная структура таких языков. Некоторые словосочетания в ходе развития и эволюции языка приобретают иной либо многозначный смысл. Так, иностранец, в совершенстве знающий грамматику русского языка и обладающий огромным словарным запасом, но не знающий смысла фразеологических сочетаний, столкнется с трудностью перевода таких из них, как «заварить кашу», «потерять голову», «намылить шею», «съесть собаку», «зарыть талант в землю» и т. д. Имеется большое количество грамматических исключений: слова-*синонимы* (различные слова, обозначающие один и тот же объект, например: дорога-путь); слова-*омонимы* (слова одинаково звучащие, но обозначающие различные понятия, например: линия, воля, авторитет). Логические связи во фразах реальных языков также выражены недостаточно отчетливо, что делает неудоб-

ным вывод логических следствий из посылок, заданных в форме фраз естественного языка.

Информационный язык должен быть приспособлен к использованию его в машине, которая, не обладая интеллектом, могла бы легко распознавать различные слова. Производя различные действия над текстами, помещаемыми в запоминающее устройство, машина не может обращаться к их интуитивному смыслу, она должна ограничиваться лишь анализом их формальной структуры. Поэтому информационный язык должен быть *однозначным* – каждое его выражение должно допускать единственное толкование и единственную форму представления.

Язык, в котором каждое выражение имеет только одно толкование или один смысл, называют формализованным.

Для создания и использования информационных систем на практике всегда оказывается необходима фиксация не только смысла выражения (семантический уровень формализации языка), но и правил построения слов, выражений из заданного алфавита, т. е. жесткое задание грамматики языка (синтаксический уровень формализации). Язык, который получается в этом случае, называют *формальным*. Формальными языками являются все языки общения с компьютером – языки программирования высокого уровня.

Рассуждая о понятии «язык» с точки зрения использования его в информационных системах, надо помнить и о существовании языка, на котором происходит обмен информацией между устройствами машины на физическом уровне.

Язык, правила интерпретации которого реализованы в компьютере аппаратно, называется машинным языком, или системой команд машины. Она зависит от конкретного аппаратного исполнения логики устройства, поэтому системы команд у разных типов компьютеров разные.

Понятие знаковой системы, или языка, играет основополагающую роль в построении *любой* информационной системы.

Подведем итоги сказанного.

1. Информация передается посредством сообщений, составленных на некотором языке.

2. Большинство языков задаются конечным алфавитом и правилами образования конструкций из символов этого алфавита.

3. Чтобы сообщения, передаваемые с помощью языка, были понятны, необходимо знать, какими именами обозначены объекты и отношения между ними в языке, на котором передается сообщение, – правила интерпретации. Для формальных языков правила должны быть заданы так, чтобы их могла реализовать машина.

Следует подчеркнуть, что только язык является средством фиксации и сохранения накопленных человечеством знаний и передачи их от поколения к поколению.

Формы представления информации. Многообразие источников и потребителей информации привело к существованию различных форм ее представления: символьной, текстовой, графической, звуковой и т. д.

Символьная форма информации основана на использовании символов: букв,

цифр, знаков математических и логических операций, других знаков. Эта информация является наиболее простой, поэтому на практике применяется для передачи сигналов о различных событиях. Например, уличный светофор или знаки дорожного движения несут несложную информацию о правилах движения. В компьютере это, например, признаки нажатия специальных управляющих клавиш на клавиатуре: *Ins*, *Caps Lock*, *Num Lock*, *Scroll Lock*, *Alt*, *Ctrl*, левый *Shift*, правый *Shift*. При нажатии любой из этих клавиш в соответствующий бит (в том порядке как перечислены клавиши) байта памяти с адресом 0040:0017h заносится 1, а не код символа в ячейку памяти, как это происходит при вводе символов. Символьная информация значима, прежде всего, по своим информационным качествам, при этом порядок расположения знаков относительно друг друга роли не играет.

Текстовая форма информации является более сложной формой. Здесь так же, как и в предыдущей форме, используются символы, цифры, математические и другие знаки. Однако информация заложена не только в этих символах, но и в их сочетаниях и зависит от расположения знаков относительно друг друга. Например, «кот–кто–ток», или 7285–2857–5782. Символы в словах и числах одни и те же, но в зависимости от расположения несут разную информацию.

Понятие «текст» в широком смысле не требует никаких ограничений не только на набор изображающих знаков, но и на правила интерпретации текста. Любая информация, представленная в символьной форме, может рассматриваться как текст. Это могут быть различные документы (анкеты, отчеты, статьи, письма, программы, написанные на языках программирования высокого уровня) или массив десятичных чисел (67,3; 453,7; 98,65; 104,8; 234,00; 93,73; 578,12; 326,12; 98,62...), вводимых в компьютер для расчетов. Все это текстовая информация. Она чрезвычайно удобна благодаря письменному отображению человеческой речи и наиболее распространена.

Графическая форма информации принципиально отличается от текстовой тем, что в ней используется единственный вид изображающего символа – точка на плоскости. Все изображения объектов конструируются из точек. *Графическая форма представления данных более информативна* и является самой емкой и сложной формой, в которой могут быть представлены графики, чертежи, фотографии, видеоролики, кадры фильмов или телепередач и т. д. Использование цвета еще более существенно повышает информационную емкость такой формы данных.

Звуковая форма информации также является очень емкой и сложной формой, в которой представляются речь, вокальные произведения, музыка, звуковые сигналы и др.

С широким распространением текстовой формы информации в виде представления различных структур связаны и формы ее обработки на компьютерах. Одними из первых наиболее распространенных и популярных были и остаются до сих пор системы обработки текстов, системы обработки электронных таблиц и системы управления базами данных.

Системы подготовки и представления текстовой информации[♦]. Появление дешевых персональных компьютеров и принтеров стимулировало их применение в таких сферах, где подготовка и представление информации в виде текстовой документации являются главным видом деятельности. В принципе любая работа так или иначе связана с составлением текстовых документов. *Текстовые процессоры, или редакторы* (так называются компьютерные программы), служат для подготовки самых разнообразных видов текстовой документации – статей и монографий, служебных писем и записок, технических описаний, отчетов, реферативных обзоров и приказов. Практически любой документ, который в течение многих десятилетий обычно готовился на пишущей машинке, сейчас создается на компьютере с помощью текстового редактора. При этом появляется возможность многократно и быстро исправлять отдельные слова, предложения и фрагменты, не вводя заново весь текст, вставлять в текст другие объекты, также подготовленные на компьютере (таблицы, рисунки, диаграммы и т. д.), выделять текст другим начертанием и размером шрифта, удобно располагать его на странице и печатать на принтере в нужном количестве экземпляров. Можно автоматически составлять оглавления документов, проверять орфографию и грамматику, составлять по заданным текстам словари и выполнять другие операции.

Таким образом, подготовка и представление информации в виде текстовых материалов на компьютере могут выполняться не только быстрее и эффективнее, чем на пишущей машинке, но и использовать недоступные ранее возможности. Пример представления текстовой информации – текст этой книги.

Системы подготовки и представления информации в виде таблиц. Для профессиональной работы с данными, расположенными в документах табличной формы, одной из самых распространенных и гибких информационных технологий является использование специальных программных продуктов – *табличных процессоров*, которые применяются в основном для автоматизации учрежденческой деятельности, где разработка различного рода таблиц и вычисления в них играют важнейшую роль. Они позволяют с большой степенью автоматизации создавать простые и сложные табличные документы, использовать различные встроенные функции для вычисления значений в таблицах по формулам, строить диаграммы и графики по цифровым данным таблиц. В документах рабочих таблиц можно использовать методы автоматического контроля ввода данных и поиска ошибок вычислений, средства поиска оптимального решения и подбора параметров, анализ достоверности вычислений.

Можно создавать пользовательские программы-макросы для выполнения часто используемых последовательностей команд, шаблоны форм различных документов, использовать элементы управления (раскрывающиеся списки, переключатели, флажки, счетчики и т. д.) для автоматизации заполнения форм документов. Можно автоматически провести анализ «а что, если...», или обработать статистические данные, переслать информацию из таблиц в базы дан-

[♦] Текстовые и табличные процессоры, системы управления базами данных и другие прикладные программы изучаются на лабораторных занятиях.

ных, или получить ее из имеющихся баз данных. Довольно мощным средством для быстрого анализа большого количества очень сложных данных, получения как бы различных срезов с этих данных в нескольких измерениях за счет быстрого перемещения полей таблицы с одного места на другое служат сводные таблицы. Пример представления информации в виде таблицы и графиков, созданных в табличном процессоре *Excel* приведен на рис. 2.

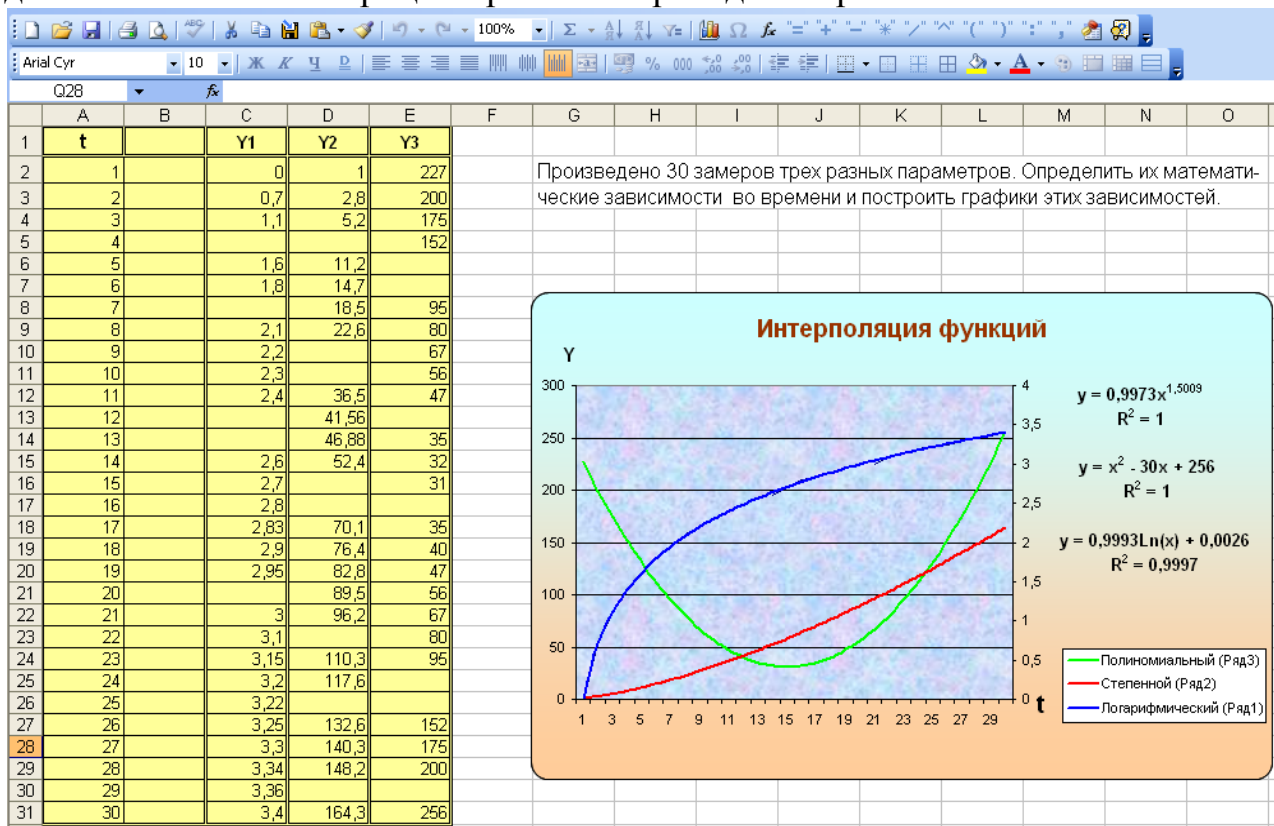


Рис. 2. Пример таблицы и графиков, созданных в Excel

Кратко о базах данных. База данных, как и любая база (стройматериалов, продуктов, овощей и т. д.), создается на длительный период использования и предназначена для накопления, хранения и выдачи данных в различные отчеты, справки, запросы и другие документы. Наиболее распространены *реляционные* базы данных. Это как бы многомерные базы: информация в них хранится в большом количестве таблиц, связанных между собой с помощью одного определенного признака. Информация в базу данных может поступать из разных источников: отчетов, таблиц, учетной документации т. д. Наиболее разумный способ группировки данных заключается в размещении всей связанной информации в одной строке таблицы, а однородной информации – в одном столбце. Например, в списке сотрудников фирмы полная информация о каждом сотруднике будет занимать одну строку, а столбцы будут содержать отдельные элементы этой информации: фамилия, имя, год рождения, образование, должность, стаж работы и т. д. Полные строки базы данных называют *записями*, а отдельные элементы ее (столбцы) – *полями записи*. Поля записи должны быть поименованными. Информация в базе данных должна быть полной, т. е. содержаться

необходимые компоненты для обработки и представления пользователям информации любой из рассмотренных форм.

Для обработки и представления информации других форм существует также множество различных программ: растровые и векторные графические редакторы, программы воспроизведения звуковых и музыкальных файлов и т. д.

Лекция 5

Процессы восприятия, сбора и передачи информации в информационных системах

Восприятие информации. Благодаря восприятию информации обеспечивается связь системы с внешней средой. Современные информационные системы, создаваемые, как правило, на базе ЭВМ, в качестве своей составной части имеют более или менее развитую систему восприятия, которая может представлять собой довольно сложный комплекс программных и технических средств.

Как уже говорилось, для развитых систем восприятия существует несколько этапов переработки поступающей информации. В зависимости от анализаторов, входящих в комплекс технических средств системы восприятия, организуется восприятие зрительной, акустической и других видов информации. Кроме того, различают статическое и динамическое восприятие. При динамическом восприятии особо выделяют системы, функционирующие в том же темпе, в каком происходят изменения в окружающей среде. Важнейшей проблемой восприятия информации является проблема интеграции информации, поступающей из различных источников и от анализаторов разного типа в пределах одной ситуации.

Очень кратко рассмотрим процесс восприятия наиболее важного вида информации – зрительной (напомним, что человек более 80 % информации получает именно через органы зрения). Можно выделить несколько уровней зрительного восприятия информации, приведенных на рис. 4.

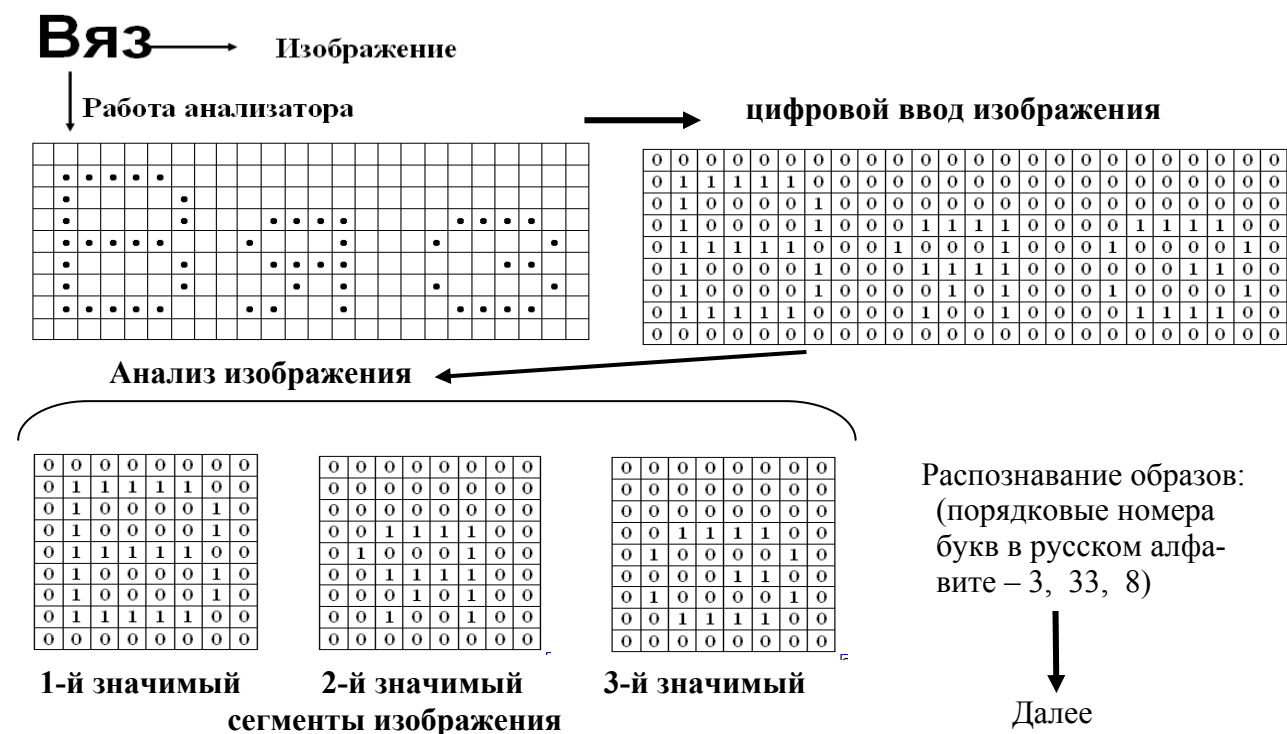


Рис. 4. Принцип работы системы зрительного восприятия текста

1. Получение изображения, поступающего от рецепторов. Как правило, к ЭВМ подключают специальные устройства цифрового ввода изображения, аналогичные широко распространенным сканерам, в которых яркость каждой точки изображения кодируется одним или несколькими двоичными числами.

2. Построение образной модели. На этом уровне с помощью специально разработанных алгоритмов происходит обнаружение объектов в описании изображения и разбиение его на значимые сегменты. Эффективность алгоритмов анализа объектов определяет скорость работы системы восприятия.

3. Построение образно-семантической модели. На этом уровне за счет информации, имеющейся во внутренней модели внешнего мира, и за счет знаний, хранящихся в ней, опознаются выделенные на предыдущем уровне объекты, и между ними устанавливаются пространственные, временные и другие виды отношений.

4. В технических системах на этом уровне восприятия используются методы распознавания образов. Полученные знания о текущей ситуации могут использоваться в дальнейшей работе.

5. Приведенный рис. 4 поясняет в общих чертах работу системы зрительного восприятия текстовой информации. Далее, после распознавания образов, возможно, например, обращение к словарю системы для обнаружения в нем считанного слова и определение его семантики.

С точки зрения информационной системы в целом система восприятия осуществляет первичную обработку собираемой извне информации. В свою очередь, для системы восприятия первичную обработку информации производит система сбора информации. На практике часто встречаются информационные системы, не обладающие развитой системой восприятия (из-за отсутствия необходимости в таковой). В этом случае система восприятия представляет собой систему сбора информации.

Сбор информации. Система сбора информации представляет собой сложный программно-аппаратный комплекс. Как правило, современные системы сбора информации не только обеспечивают кодирование информации, ее ввод в ЭВМ, но и выполняют первичную обработку этой информации. *Сбор информации – это процесс получения информации из внешнего мира и приведение ее к виду, стандартному для данной информационной системы.* Обмен информацией между воспринимающей системой и окружающей средой осуществляется, как уже упоминалось, посредством сигналов.

Сигнал можно определить как средство перенесения информации в пространстве и времени. Подобно живым организмам, воспринимающим сигналы из внешней среды с помощью специальных органов (обоняние, осязание, слух, зрение, вкус), технические системы для приема сигналов из внешней среды оснащаются специальными устройствами. Вне зависимости от носителя сигнала типичный процесс обработки информации (сигнала) состоит из следующих ша-

гов. На первом шаге исходный (первичный) сигнал с помощью специального устройства (датчика) преобразуется в эквивалентный ему электрический сигнал. На втором шаге вторичный (электрический) сигнал в некоторый выделенный момент времени оцифровывается специальным устройством – аналого-цифровым преобразователем (АЦП). АЦП устанавливает соответствие между значением электрического сигнала и некоторым числом из конечного множества таких чисел. Таким образом, датчик и АЦП, связанные вместе, составляют цифровой измерительный прибор. Если этот прибор оснастить некоторым устройством для хранения измеренной величины – регистром, то на следующем шаге по команде от ЭВМ можно это число ввести в машину и подвергнуть затем любой необходимой обработке. Следует заметить, что не все технические средства сбора информации работают по описанной схеме. Клавиатура ПК, например, не имеет АЦП. Здесь нажатие клавиши с определенным символом непосредственно программным путем преобразуется в стандартный цифровой код символа. Но в любом случае поступающая в ЭВМ информация представляется в виде цифрового кода – двоичного числа. Современные системы сбора информации (например, в составе автоматизированных систем управления) могут включать в себя тысячи цифровых измерительных приборов и всевозможных устройств ввода (от датчиков в ЭВМ, от человека в ЭВМ, от одной ЭВМ к другой и т.д.).

Совокупность технических средств ввода информации в ЭВМ, программ, управляющих всем комплексом технических средств, и программ, обеспечивающих ввод информации с отдельных устройств ввода (драйверов устройств), – вот что представляет собой современная развитая система сбора информации.

Сбор и регистрация информации происходят по-разному в различных экономических объектах. Наиболее сложна эта процедура в автоматизированных управленческих процессах промышленных предприятий и фирм, где производятся сбор и регистрация первичной учетной информации, отражающей производственно-хозяйственную деятельность объекта. Особое значение при этом придается достоверности, полноте и своевременности первичной информации. На предприятии сбор и регистрация информации происходят при выполнении различных хозяйственных операций (прием готовой продукции, получение и отпуск материалов, трудозатраты на выполнение определенного производственного процесса и т. д.). Сначала информацию собирают, затем ее фиксируют. Учетные данные могут возникать на рабочих местах в результате подсчета количества обработанных деталей, прошедших сборку узлов, изделий, выявление брака и т. д. Для сбора фактической информации производятся измерение, подсчет, взвешивание материальных объектов, получение временных и количественных характеристик работы отдельных исполнителей. Сбор информации всегда сопровождается ее регистрацией, т. е. фиксацией на материальном носителе (напомним, что информация всегда существует на каком-то материальном носителе). В системах организационно-экономического управления запись в первичные документы в основном осуществляется вручную, поэтому процедуры сбора и регистрации остаются пока наиболее трудоемкими. В технических системах авто-

матического управления эти операции автоматизированы за счет использования различных датчиков, измерительных и регистрирующих приборов, что значительно повышает скорость сбора и регистрации первичной информации.

Передача информации. Передача информации осуществляется различными способами: передача по телекоммуникационным каналам связи, с помощью курьера, пересылка по почте, доставка транспортными средствами и т.д. Некоторые технические средства сбора и регистрации, собирая автоматически информацию с датчиков, установленных на рабочих местах, передают ее непосредственно в ЭВМ. В современных условиях большое распространение получила распределенная обработка данных в информационно-вычислительных сетях (ИВС), которые представляют наиболее динамичную и эффективную отрасль автоматизированной технологии процессов ввода, передачи, обработки и выдачи информации. Важнейшим звеном ИВС является канал передачи данных, структурная схема которого представлена на рис. 5.

Непрерывный канал связи (НКС) совместно с функционирующими на его концах модемами образует дискретный канал связи (ДКС). В свою очередь, ДКС и устройства повышения достоверности (УПДс) образуют канал передачи данных.



Рис. 5. Структурная схема канала передачи данных:

УПД – устройство подготовки данных; *НКС* – непрерывный канал связи;
ДКС – дискретный канал связи; *УПДс* – устройство повышения достоверности;
Модем – модулятор/демодулятор – устройство преобразования данных

В НКС элементы данных передаются в виде физических сигналов, которые описываются непрерывными функциями времени. Большинство НКС оказываются непригодными для передачи сигналов, отображающих данные. С помощью модулятора информационный сигнал воздействует на некоторый параметр сигнала-переносчика, благодаря чему спектр сигнала смещается в область частот, для которых наблюдается наименьшее затухание в выбранном НКС. Обратную операцию, переход от модулированного сигнала (сигнала-переносчика) к модулирующему (информационному сигналу), осуществляет демодулятор. Понятие ДКС позволяет, отвлекаясь от физической природы процессов, происходящих в НКС, представлять совокупность НКС и модемов на его концах как некоторый «черный ящик», на вход которого подается последовательность кодовых символов – входное сообщение. Оно может представлять собой некоторый текст на русском языке, а может быть и двоичным кодом – последовательностью нулей и единиц. Чаще всего в ИВС рассматриваются ДКС с двоичным алфавитом, когда входное и выходное сообщения представляют собой двоичные кодовые последовательности.

Завершая общее описание канала связи, рассмотрим УПДс. УПДс может представлять собой специальную аппаратуру, предназначенную для повышения достоверности передачи данных, а может, особенно в современных ИВС, представлять собой специальную программу и ЭВМ, на которой она выполняется, может являться как элементом канала связи, так и элементом системы обработки информации. Повышение помехоустойчивости и защищенности передаваемой информации в каналах связи кратко рассмотрено в *лекции 7*.

Дистанционно может передаваться как первичная информация с мест ее возникновения, так и результатная в обратном направлении. В этом случае результатная информация отражается на различных устройствах: дисплеях, табло, печатающих устройствах. Дистанционная передача постоянно развивается и совершенствуется. Особое значение этот способ передачи информации имеет в многоуровневых информационных системах, где применение дистанционной передачи значительно ускоряет прохождение информации с одного уровня на другой и сокращает общее время обработки данных.

Лекция 6

Системы счисления

Системой счисления называется совокупность приемов и способов обозначения (записи) чисел. Эта разновидность знаковых систем – результат длительного исторического развития записи чисел, начавшегося, видимо, с возникновения так называемой *единичной* системы счисления, в которой для записи чисел применялся только один вид знаков – «палочка». Каждое число в ней состояло из «палочек», количество которых равнялось обозначаемому числу. С развитием человека палочки уже не устраивали его для представления чисел и, главное, действий над ними. Стали появляться различные *непозиционные* системы счисления с большим числом знаков для записи различных чисел (вавилонская, египетская, римская и др.).

Одна из немногих непозиционных систем счисления, которая сохранила свое значение в настоящее время, – это *римская* система счисления. Алфавитом ее являются несколько заглавных латинских букв: I, V, X, L, C, D и M, с помощью которых записываются соответственно числа: I = 1, V = 5, X = 10, L = 50, C = 100, D = 500, M = 1000. Конструкциями этого алфавита являются различные натуральные числа, причем одна и та же цифра, стоящая в разных местах (позициях) числа, не меняет своего значения, например: MCMXCIX = 1999, XLVIII = 48, XII = 12, и т. д. Здесь цифры M, C, X, I, независимо от их расположения в числе, сохраняют свое номинальное значение.

Отметим правила записи чисел в римской системе счисления:

- если более значимая цифра стоит перед меньшей цифрой или если подряд стоящие цифры одинаковы, то они арифметически складываются, например: XXXVI = 36;
- если меньшая цифра стоит перед большей, то она вычитается из боль-

шей, причем меньшая цифра в этом случае повторяться подряд не может, например: XL = 40, CDIV = 404 (но нельзя IIIV = 3);

- цифры M, C, X, I могут повторяться в записи числа не более трех раз подряд, а цифры D, L, V – по одному разу.

С помощью принятого в римской системе счисления алфавита максимальное число может быть таким: MMMCMXCIX = 3999. Для записи больших чисел надо расширять алфавит языка. По этой причине, а также из-за отсутствия цифры 0 римская система счисления не годится для записи действительных чисел. Попробуйте перемножить два числа CLVI и LXXIV, не прибегая к переводу их в привычную для нас десятичную систему счисления. Думаю, что это вам не удастся. Учитывая сказанное, сформулируем недостатки непозиционных систем счисления:

- в них нельзя записать любое число;
- запись чисел громоздка и неудобна;
- арифметические операции над числами крайне затруднены.

Сейчас римская система счисления используется в основном для обозначения веков, томов книг, на циферблатах часов и т. д.

Поэтому уже несколько веков используются *позиционные* системы счисления, в которых каждая цифра, содержащаяся в записи числа, занимает определенное место, называемое *разрядом*. Отсчет разрядов производится справа налево. Классический пример позиционной системы счисления – *десятичная*, к которой мы привыкли с детства и которая получила в мире самое широкое распространение. Каждая цифра позиционной системы счисления *несет в себе двойную нагрузку*: собственное значение и значение, принимаемое ею в зависимости от позиции в числе. Например, в числе 3983 одна и та же цифра 3 принимает разные значения (три и три тысячи) в зависимости от ее позиции (разряда) в числе. Количество цифр, используемых для записи любых чисел в позиционной системе счисления, носит название *основания системы*.

Десятичная система счисления возникла приблизительно в V в. н. э. в Индии. Первыми о ней узнали арабы, которые в VII – VIII вв. завоевали обширные районы Азии и Средиземноморья. Поскольку Европа узнала о ней от арабов, то алфавит десятичной системы счисления стали называть *арабскими* цифрами. Правила выполнения основных арифметических операций в ней впервые были разработаны выдающимся восточным математиком и астрономом Абу Джафар ибн Мусы аль-Хорезми и описаны им в труде «Книга о сложении и вычитании по исчислению индийцев».

В позиционной системе счисления любое действительное число и его значение можно представить в виде следующего полинома:

$$R_n R_{n-1} R_{n-2} \dots R_0 R_{-1} R_{-2} \dots R_{-m} = R_n \cdot K^n + R_{n-1} \cdot K^{n-1} + \dots + R_1 \cdot K^1 + R_0 \cdot K^0 + R_{-1} \cdot K^{-1} + R_{-2} \cdot K^{-2} + \dots + R_{-m} \cdot K^{-m},$$

где n – количество цифр в целой части; m – количество цифр в дробной части числа (после запятой), K – основание системы счисления, а R – любые цифры, принятые в данной системе счисления (*разряды*). Задавая различные значения

K , можно получить множество различных позиционных систем счисления. Откуда вытекают их общие свойства:

- в позиционных системах счисления запись чисел удобна и компактна: в них можно записать любое число – сколь угодно большое или сколь угодно малое, чего нельзя сделать в непозиционных;

- при переходе цифры в числе на один разряд влево (вправо) ее значение увеличивается (уменьшается) во столько раз, чему равно основание системы. Поэтому умножение любого числа на K^m , где K – основание данной системы счисления, а m – целое число, сводится просто к переносу в числе запятой на m разрядов влево или вправо, в зависимости от знака m .

- благодаря поразрядной организации записи чисел в позиционной системе счисления с ними легко проводить арифметические операции;

Несмотря на то, что исторически человек привык работать с числами в десятичной системе счисления, для хранения, обработки и передачи информации в технических устройствах она крайне неудобна.

Во-первых, числа в электронные схемы устройств не запишешь так, как ручкой или карандашом на бумаге. Это можно сделать, если *каждой цифре ставить в соответствие какое-то фиксированное состояние технического элемента*, причем таких состояний должно быть десять и они друг от друга должны *резко отличаться*, иначе устройство не определит, с какой цифрой имеет дело. Но, практически какой бы элемент технической базы устройств мы ни взяли, обнаруживается только *два* резко отличных друг от друга состояния. Примеры: магнитный элемент – намагничен/размагничен; электрический конденсатор – заряжен/разряжен; контакт реле – замкнут/разомкнут; полупроводниковый прибор – проводит/не проводит ток и т. д. Отсюда вывод: *с помощью имеющихся в настоящее время элементов цифровых электронных устройств можно отражать только две цифры – 0 и 1. При этом информация хранится до тех пор, пока хранится состояние элемента, соответствующее ей.*

Во-вторых, информацию между устройствами нужно передавать какими-то сигналами, причем устройство должно четко и однозначно определять, что один сигнал – это цифра 2, а другой сигнал – цифра 7. При использовании десятичной системы счисления таких сигналов должно быть десять, и они должны так же четко отличаться друг от друга, как и состояния элементов. Учитывая, что устройства информационной техники – это в основном электрические устройства (электронные), то из резко отличных друг от друга дискретных сигналов можно выбрать опять только два – есть импульс тока/нет импульса тока. *С помощью этих сигналов можно передавать только две цифры – 1 и 0.*

В-третьих, чем меньше различных сигналов в цепях устройств, тем проще схемы, тем менее они затратны и тем надежнее работают.

Именно поэтому в устройствах информационной техники применяется **двоичная система** счисления, использующая две цифры (0 и 1) для представления любых чисел. Числа в двоичной системе счисления записываются так же, как в десятичной системе счисления, один разряд за другим. Число и его значение по-прежнему представляются в виде полинома:

$$R_k R_{k-1} \dots R_1 R_0 R_{-1} \dots R_{-m} = R_k \cdot 2^k + R_{k-1} \cdot 2^{k-1} + \dots + R_1 \cdot 2^1 + R_0 \cdot 2^0 + R_{-1} \cdot 2^{-1} + \dots + R_{-m} \cdot 2^{-m}$$

Здесь отличие состоит только в том, что разряд R_n может принимать значения не от 0 до 9, а только два: 0 или 1. Например:

$$100111,101 = 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 39,625_{10}.$$

Все арифметические операции над числами в двоичной системе счисления выполняются точно так же, как и в десятичной (поразрядно, по тем же правилам), только нужно помнить, что 1 при переходе из соседнего разряда изменяет свое значение в два раза, а не в десять. Примеры:

$\begin{array}{r} 1001011101 \text{ (605)} \\ + 10111011 \text{ (187)} \\ \hline 1100011000 \text{ (792)} \end{array}$	$\begin{array}{r} 1001011101 \text{ (605)} \\ - 10111011 \text{ (187)} \\ \hline 110100010 \text{ (418)} \end{array}$	$\begin{array}{r} 1101 \text{ (13)} \\ \times 111 \text{ (7)} \\ \hline 1101 \\ + 1101 \\ \hline 1101 \\ \hline 1011011 \text{ (91)} \end{array}$	$\begin{array}{r} 1011011 \text{ (91)} \big 1101 \text{ (13)} \\ - 1101 \\ \hline 10011 \\ - 1101 \\ \hline 1101 \\ - 1101 \\ \hline 0000 \end{array}$
--	---	---	---

Очевидно, что сдвиг двоичного числа вправо или влево на n разрядов уменьшает или увеличивает его в 2^n раз. Это широко применяется в информационных устройствах путем использования специальных схем – сдвиговых регистров, особенно в операциях умножения и деления, которые заменяются многократными операциями сложения и вычитания, где сдвиги чисел необходимы. Основание системы число 2 записывается как 10, т. е. одна двойка и ноль единиц.

Перекодировка в двоичную систему счисления при вводе информации в компьютер и из двоичной системы – при выводе на экран или печать производится автоматически программным путем.

Простота действий над двоичными числами определяет простоту построения электронных схем для их выполнения. Кроме того, как мы увидим дальше, в двоичной системе счисления, т. е. с помощью цифр 0 и 1 можно записать в память компьютера любую информацию (числовые данные, текст, графику, музыку, видеоизображения, кинокадры или сигналы от устройств управления объектами и др.) и обработать ее.

Сравнивая записи чисел в десятичной и двоичной системах счисления, нетрудно заметить, что в двоичной системе счисления для записи чисел требуется значительно больше разрядов. Таким образом, выигрывая в простоте, мы проигрываем в компактности записи чисел и ее наглядности. Поэтому в вычислительной технике нашли широкое применение системы счисления, основания которых кратны двум, – *восьмеричная и шестнадцатеричная*. По существу, это разновидности записи двоичных чисел, выполненные более компактно другими символами.

В восьмеричной системе счисления используются первые восемь арабских цифр, и основание ее $8 = 2^3$ записывается в виде 10, то есть одна восьмерка и ноль единиц.

В шестнадцатеричной системе счисления используется 16 символов для записи чисел – это десять арабских цифр и шесть начальных букв латинского алфавита: 0,1,2,...,9,A,B,C,D,E,F. Основание ее $16 = 2^4$ записывается также в ви-

де 10, т. е. одна шестнадцатеричка и ноль единиц. Для обозначения шестнадцатеричных чисел обычно за числом записывают латинскую букву *H* или *h*, например: *6AB72EDC98h*.

Каждая цифра в восьмеричной системе счисления записывается тремя двоичными цифрами ($2^3 = 8$), а каждая цифра шестнадцатеричной системы счисления – четырьмя двоичными цифрами ($2^4 = 16$). Разбиение двоичного числа на триады или тетрады производится справа налево. Недостающие цифры слева в самой левой триаде или тетраде дополняются нулями, например:

$$255_{10} = 11111111_2 = 011\ 111\ 111 = 377_8 = 1111\ 1111 = FF_{16}.$$

Следует подчеркнуть, что эти системы счисления используются только для удобства пользователя – просматривать отображение на устройствах вывода находящейся в памяти информации. Внутри же компьютера информация не может быть записана иначе как в двоичной системе счисления.

Для перевода целых чисел из одной системы счисления в другую нужно исходное число и получающиеся частные от деления делить последовательно на основание системы счисления, в которую оно переводится, записанное в исходной системе счисления. Получающиеся остатки от деления и последнее частное, меньшее делителя, и есть цифры числа новой системы счисления, записанные в исходной. Последнее частное от деления будет старшим разрядом числа. Поэтому записывать число в новой системе счисления нужно, начиная с него, как показано стрелкой. Например, переведем десятичное число 23 в двоичную систему счисления. Получаем: $10111 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 23$.

Для перевода дробей нужно дробную часть числа и дробные части получающихся промежуточных произведений умножать на основание новой системы счисления, записанное в исходной, до тех пор, пока в дробной части промежуточного произведения не получатся все нули или пока не будет достигнута требуемая точность (количество цифр в новой системе счисления). Цифры в целой части промежуточных произведений и есть цифры новой системы счисления, записанные в исходной системе счисления. Записывая их последовательно от начала к концу, получим число в новой системе счисления. Например, переведем десятичное число 0,875 в восьмеричную систему счисления. Получаем: $0,7_8 = 7 \cdot 8^{-1} = 7 \cdot 0,125 = 0,875$.

$$\begin{array}{r} \times 0,875 \\ 7,1000 \\ \hline \end{array}$$

Перевод чисел из любой системы счисления в десятичную систему на практике проще и удобнее выполнять путем суммирования всех разрядов числа, записанного в виде полинома, как показано выше в примерах.

Лекция 7

Измерение количества информации и ее кодирование

Из физики известно, что за единицу измерения любой физической величины принимается ее количество, равное *эталонной мере*. Длину измеряют отрезком длиной 1 м, массу тела измеряют единицей массы 1 кг и т. д. Аналогично измеряют и информацию. Имеется несколько принципиально различных способов ее измерения.

Наиболее известны энтропийный[♦] и объемный способы. Информация – величина случайная. Количество информации зависит от вероятности события, о котором идет речь в сообщении. Пусть, например, по радио 4 августа сообщают в прогнозе погоды следующее: «Завтра осадков в виде снега не ожидается». Количество информации в этом сообщении практически равно нулю, так как вероятность снега 5 августа практически равна нулю. Если же вместо снега сообщали бы о дожде, то такое сообщение несло бы определенное количество информации, так как вероятность дождя может быть достаточно большой.

На практике наиболее распространен объемный способ. Объемное понятие количества информации было введено в 1928 г. американским инженером Р. Хартли. Он предложил определять количество информации как логарифм числа возможных сообщений N с одинаковым числом символов n из определенного алфавита с количеством символов m :

$$I = \log_a N = \log_a m^n = n \cdot \log_a m$$

Выбор основания логарифма может быть любым и скажется только на единицах измерения количества информации. При $a = 10$ количество информации измеряется в десятичных единицах, при $a = 2$, как принято в технических средствах информационных систем, количество информации измеряется в двоичных единицах. Формула Хартли дает $I = 1$ при $a = 2$, $m = 2$ и $n = 1$, т. е. информацией в 1 двоичную единицу будет обладать сообщение, составленное из одного символа двоичного алфавита. *Это элементарная информация, передаваемая сообщением об одном из двух равновероятных исходов события.* Такая информация называется *двоичной*, или *бит* (от **binary digit**). Пусть объект может находиться в одном из восьми равновероятных состояний. Количество информации, поступающей в сообщении о том, в каком именно состоянии он находится, будет равно $I = \log_2(8) = 3$ бита. (Состояния: 000, 001, 010, 011, 100, ..., 111).

Измерение информации в двоичных единицах особенно удобно для технических приложений, так как в системах, хранящих и преобразующих информацию, проще всего оперировать с величинами, выраженными в двоичной системе счисления, а одна двоичная единица информации есть как раз то количество ее, которое необходимо для указания одной двоичной цифры. Этот же принцип используется в нервной системе человека и животных (два состояния нервной

[♦] Энтропия в теории информации – мера неопределенности ситуации с конечным или счетным числом исходов. Пример неопределенной ситуации – бросание игрального шестигранного кубика с шестью возможными исходами. Чем больше возможных исходов в ситуации, тем больше начальная неопределенность и тем больше количество информации, получаемое после проведения опыта.

клетки – возбуждение и торможение, т. е. релейная система передачи нервного импульса «замкнуто/разомкнуто»). Кроме того, одна двоичная единица информации есть количество информации, необходимое для задания одного значения булевой* функции, что удобно для математико-логического анализа вопросов переработки информации. Основной единицей измерять физические величины не всегда удобно, поэтому вводятся производные единицы от основной, например, для измерения длины – 1 мм, 1 см, 1 км и т. д. Точно так же и для измерения информации 1 бит – слишком маленькая единица измерения, поэтому введены более крупные единицы, но кратные не 10, как мы привыкли, а двум: 1 байт = $2^3 = 8$ бит, 1 Килобайт (Кб) = $2^{10} = 1024$ байта, 1 Мегабайт (Мб) = $2^{20} = 1024$ Кбайта, 1 Гигабайт (Гб) = $2^{30} = 1024$ Мбайта и т. д. В этих единицах измеряется объем оперативной и внешней памяти компьютеров, размеры компьютерных документов – файлов и количество передаваемой информации по каналам связи.

Для представления в компьютерах и передачи по каналам связи любая информация *кодируется*, т. е. преобразуется в специальную систему символов – *код*. Кодами являются числа. Поставим в соответствие символам русского алфавита цифры: А → 0; Б → 1; В → 2; Г → 3; Д → 4 и т. д. Распознавание техническим устройством даже десяти цифр – задача неразрешимая. Поэтому для упрощения задачи распознавания такой системой символов являются две двоичные цифры 0 и 1, т. е. *любая информация в компьютере кодируется в виде двоичных чисел*. Устройствам компьютера распознавать нужно только две цифры, соответствующие состояниям электронного переключателя, а при передаче – наличие или отсутствие импульса тока.

Некоторое множество слов одного алфавита можно называть кодом, если оно поставлено во взаимно однозначное соответствие с множеством слов другого алфавита.

По ходу заметим, что если каждый образ при кодировании является отдельным знаком, то такое отображение называют *шифровкой*, а образы – *шифрами*. Например, вместо слова ИНФОРМАТИКА можно получить его шифр – МРЧСУПГХМНГ, записывая взамен каждой буквы этого слова следующую за ней в алфавите третью букву.

Код называется *различимым*, если существует взаимно однозначное соответствие между множествами всех последовательностей слов данного кода и всех последовательностей слов первоначального алфавита, т. е. последовательностей элементов сообщений. Каждое слово, входящее в код, называется *кодовой комбинацией*, например совокупность двоичных цифр, образующих двоичное число, которое соответствует какому-либо символу русского алфавита или знаку. Одну двоичную цифру (0 или 1), входящую в кодovou комбинацию, называют *элементом кода*. Число элементов кода в кодовой комбинации называется ее *длиной* и определяет значность кода: 5-значный, 7-значный и т. д. Кодовые комбинации для разных символов одного и того же алфавита могут быть различной длины. Такой код будет *неравномерным*. В качестве примера приведем код Морзе:

* Функции Буля изучает алгебра логики, которая кратко рассмотрена в лекции 9.

А ··	Б ····	В ···	Г ---·	Д -···	Е ·
Ж ···-	З ----·	И ··	Й ·----	К ---·	Л ····
М --	Н ··	О ---	П ····	Р ···	С ···
Т -	У ···	Ф ····	Х ····	Ц ····	Ч ----·
Ш ----	Щ ----	Ъ,Ь ····	Ы ----	Э ····	Ю ····
Я ····	1 ·····	2 ·····	3 ·····	4 ·····	5 ·····
6 ·····	7 ·····	8 ·····	9 ·····	0 ·····	

В коде Морзе для обозначения букв, цифр, знаков препинания и различных служебных сигналов используются комбинации точек и тире. Кодовому знаку «точка» соответствует сигнал, состоящий из посылки (импульса) тока и паузы одинаковой длительности, а кодовому знаку «тире» соответствует сигнал из посылки тока тройной длительности и паузы. Кодовые комбинации отделяются друг от друга специальным разделительным знаком, представляющим собой двойную паузу. Как видно, код Морзе *неравномерный*. Здесь применен принцип частоты использования букв в английском алфавите (чем чаще используется буква, тем короче ее код).

Чтобы уменьшить значность кода для передачи большого числа символов, их делают многопрограммными. Примером может служить международный телеграфный код Бодо – равномерный пятизначный двоичный код. Число различных кодовых комбинаций в коде Бодо, следовательно, равно: $2^5 = 32$, что недостаточно для передачи всех букв, цифр, знаков препинания и т.д. Поэтому код Бодо является *двухпрограммным*. Одна из программ (буквенный регистр) объединяет все буквы за исключением *ъ, э, й, щ*, а другая – оставшиеся буквы, цифры, знаки препинания и т. д. (цифровой регистр). При передаче данных выбор регистра принимаемым аппаратом осуществляется автоматически по кодовой комбинации, передаваемой перед сообщением. В коде Бодо кодовая комбинация 11110 соответствует первой программе, а кодовая комбинация 11101 – второй программе.

При кодировании и передаче информации по каналам связи очень существенное значение имеет *помехозащищенность и помехоустойчивость* кода, так как в каналах связи всегда имеются атмосферные, промышленные и другие помехи*, которые могут исказить передаваемые коды. Если в кодовой комбинации пропадет одна единица или появится лишняя, то сообщение искажается, так как вместо одного символа появляется другой. С целью повышения помехоустойчивости код усложняют: к информационным знакам добавляют дополнительные знаки – проверочные. Для этого выбирают из общего возможного числа кодовых комбинаций лишь те, которые отличаются друг от друга не менее чем двумя элементами. В этом случае искажение сигнала в одном элементе легко обнаруживается, вследствие чего наступает защитный отказ. Например, из всех 16 кодовых комбинаций, состоящих из 4 элементов, выделим для кодирования только те, которые содержат по две единицы: 0011, 0101, 0110, 1001, 1010, 1100. Здесь искажение любого кода в одном элементе приводит к образованию недопустимой кодовой комбинации, что ведет к наступлению защитного отказа.

* Помехи – это электрические возмущения, которые могут быть в виде излучений генераторов высокой частоты, грозных разрядов, от работы сторонних радиостанций и др.

Приведем другой пример: система пятиэлементных кодовых комбинаций, отличающихся друг от друга не менее чем тремя элементами. Полезную информацию несут второй и третий элемент справа. В этом случае появляется возможность автоматически обнаруживать и корректировать искаженный в одном элементе код и реализовать его как правильный.

- 1) 01001 Пусть принята недопустимая кодовая комбинация 01011.
- 2) 11010 Сравнивая ее с приведенными, видим, что она отличается от
- 3) 10101 первой комбинации одним элементом, от второй – двумя, от
- 4) 00110 третьей – четырьмя и от четвертой – тремя элементами.

Какой же код был передан? Скорее всего, первый, так как искажение в одном элементе наиболее вероятно. Вероятность искажения других комбинаций значительно меньше – они различаются в двух и больше элементах. Специальные схемы это определяют и скорректируют код под первую комбинацию.

Применение кодов, обеспечивающих исправление нескольких ошибок принципиально возможно, но такие коды практической ценности не представляют, так как повышение помехоустойчивости ведет к избыточному увеличению элементов кода и, как следствие, к увеличению времени передачи информации, усложнению аппаратуры и снижению ее надежности.

Избыточность широко применяется и в обиходе для повышения надежности и достоверности передаваемых сообщений. Например, текст телеграммы составлен так: «Вылетаю 3 марта, вторник, рейсом 523 восемь тридцать пять утра». Этот текст содержит избыточную информацию: получившему телеграмму и так известно, что 3 марта – вторник и что самолет рейсом 523 вылетает в 8 ч 35 мин утра. Однако при возникновении ошибки телеграфа из-за помех при передаче телеграммы дополнительные сведения позволят получить нужную информацию.

По такому принципу строятся коды *обнаружения и исправления ошибки*, например код Хэмминга. Корректирующие коды строятся так, что появление или потеря единицы вследствие помех в любой кодовой комбинации приводит к появлению недопустимой в данном коде комбинации, которая сразу же обнаруживается или исправляется.

Рассмотрим кодирование информации в компьютерах. *В компьютере любая информация не может быть физически записана иначе, кроме как в двоичной системе счисления в виде двоичного кода (двоичных чисел).*

Кодирование текстового сообщения производится заменой каждого символа соответствующим числом – двоичным кодом. Количество символов, кодируемых n разрядами двоичного числа, равно $N = 2^n$. Если взять восемь двоичных разрядов (один байт), то с их помощью можно закодировать $2^8 = 256$ различных символов. Этого вполне достаточно, чтобы закодировать 66 прописных и строчных букв русского алфавита (кириллицы), 52 прописные и строчные буквы латинского алфавита, 10 арабских цифр, 9 знаков препинания, знаки арифметических операций и т. д.

Таким образом, каждый знак в компьютере можно кодировать *равномерным восьмизначным кодом*, что очень удобно для записи в память, так как стандартом, введенным фирмой *IBM*, вся память представляется в виде восьми-

битовых ячеек. Кроме перечисленных выше знаков, из 256 двоичных комбинаций выделяют коды для кодирования знаков псевдографики и специальных управляющих знаков. Примеры: ¶ = 11001110, Ж = 10000110, 7 = 00110111, № = 11111100. Подчеркнем, что десятичные цифры, как и любые символы, представляются не их значениями в двоичной системе счисления, а двоичными кодами (сравните: значение семи равно 111, а код семи равен 00110111).

Возникает вопрос: почему, например, прописная русская буква А кодируется из 0 и 1 в виде комбинации 10000000, а не какой-то другой? Очевидно, что разработчики вычислительной техники договорились о том, как кодировать символы. Иначе было бы «вавилонское столпотворение» и информацией между различными компьютерами обмениваться было бы нельзя. Так было на первых ЭВМ, каждый тип из которых имел свою кодировку, так случилось и в конце 80-х гг. прошлого века, когда появившиеся советские персональные компьютеры типа *Искра 1030*, *Нейрон*, *ЕС-1840* . . . *ЕС-1849*, *Поиск-2* и др. имели одну (ГОСТ) кодировку, а IBM-совместимые – другую. Пользователям пришлось разрабатывать специальные программы перекодировки информации и предварительно использовать их при переходе на другой тип компьютера. Еще и сейчас в мире существует множество 8-битовых кодов (КОИ-8, МІС, ДКОИ-8 и др.). Это создает, естественно, дополнительные трудности в работе из-за необходимости разработки специальных программ перекодировки информации при переходе с одного типа компьютера на другой.

В 1961 г. в США был разработан «Универсальный стандартный код обмена информацией» – *ASCII (American Standard Code for Information Interchange)*. Этот код используется на всех персональных компьютерах с операционной системой Windows 9x. Таблица состоит из двух частей. *Общая часть* содержит от 0 до 31 – управляющие коды, от 32 до 127 – стандартные коды десятичных цифр, прописных и строчных латинских букв, специальных символов (клавиш) и т. д. Управляющие коды первоначально были разработаны в виде команд, сообщающих печатающему устройству, как форматировать печатную страницу и как распознавать конец файла. К таким командам относились, например, команды «конец строки» в печатаемом тексте или «перевод каретки» печатающего устройства в начало следующей строки.

Затем их назначение было распространено на все устройства вывода, особенно на сетевые. Например, код 17 используется для возобновления передачи данных между двумя компьютерами при асинхронной передаче после прекращения передачи. Код 21 – контрольный код, посылаемый компьютером, принимающим данные по сети. Отсутствие подтверждения приема пакета, например при сбое передачи, требует повторить передачу данного пакета.

Фактически общая часть представляет собой *стандартный семизначный равномерный код*, дополняемый нулем в старшем (восьмом) разряде.

Дополнительная часть – это коды от 128 до 255. Сюда входят символы для рисования рамок таблиц (псевдографика), буквы русского алфавита, специальные символы, такие, например, как § – 10100111, ® – 10101110 и т. д. Следует подчеркнуть, что прописные и строчные буквы как русского, так и латинского

алфавитов имеют свои собственные коды, например: **Б** – 10000001 и **б** – 10100001 или **D** – 01000100 и **d** – 01100100.

Дополнительная часть может изменяться в зависимости от типа операционной системы компьютера, страны использования и т. д. Фактически эта часть является *расширением* основного кода *ASCII* на основе международного стандарта *ISO (International Standards Organization)*. Возможность замены дополнительной части кода *ASCII* делает компьютер пригодным для использования в любой стране мира, так как именно в этой части располагают, помимо кодов специальных символов, коды национальных алфавитов.

Пользователи IBM PC в разных странах вначале вынуждены были в MS-DOS дополнительную часть таблицы приспособлять под особенности своей страны. Со временем фирма *Microsoft* решила упорядочить этот процесс и обеспечить его стандартными средствами. Были введены средства, позволяющие загружать в знакогенератор компьютера кодировки символов для разных стран – *кодовые страницы* – и переключаться между ними. Начиная с версии MS-DOS 6.22, такие средства были введены и для России. Следует заметить, что коды символов русского алфавита (кириллицы) в дополнительной части таблицы для MS-DOS и Windows9x различаются и зависят от используемой *страницы*. В MS-DOS используется 866 страница (символы кириллицы занимают коды 128÷175 и 224÷239), а в Windows9x – 1251 страница (символы кириллицы занимают коды 192–255). Кодировка, используемая в первых версиях Windows, называлась *ANSI-кодировкой*.

В *Windows NT* и основанных на ней операционных системах *Windows 2000*, *Windows XP* и др. используется 16 - разрядный *Unicode* – стандарт кодирования символов, позволяющий представить знаки практически всех письменных языков мира. Коды в стандарте *Unicode* разделены на несколько областей. Область с кодами от U+0000 до U+007F содержит символы набора *ASCII*. Далее расположены области знаков различных письменностей, знаки пунктуации и технические символы. Часть кодов зарезервирована для использования в будущем. Под символы кириллицы выделены коды от U+0400 до U+052F. Операционные системы *Windows* имеют встроенные средства, позволяющие преобразовывать символы из одной кодировки в другую, так что пользователь, переходя от одной версии *Windows* к другой, этого практически не замечает.

Существует много возможностей просмотра на экране имеющейся в компьютере таблицы кодов. В *Windows* для просмотра символов и соответствующих им кодов можно воспользоваться программой *Таблица символов* из группы программ *Стандартные*. В большинстве приложений (*Word*, *Excel* и др.) окно таблицы символов появляется при выполнении команды *Символ* из пункта меню *Вставка*. Зная шестнадцатеричный код символа, его можно набрать и при нажатой клавише *Alt* нажать клавишу с символами X/q – появится символ. Аналогично и обратное преобразование – при установке курсора справа от символа и нажатии клавиш $Alt+X/q$ вместо символа появится его шестнадцатеричный код. Зная десятичный код символа, на цифровой части клавиатуры его можно набрать и будет введен символ, соответствующий этому коду. Это так называемый *Alt-набор*.

Для больших вычислительных систем существует другой международный стандарт: 8-битовый код *EBCDIC* (*Extended Binary Coded Decimal Interchange Code*) – расширенный двоично-кодированный десятичный код для обмена информацией, разработанный фирмой *IBM*.

Кодирование графической информации. Для кодирования графической информации в двоичные коды используется несколько способов.

Кодирование изображения по точкам (растровый способ). На первом этапе вертикальными и горизонтальными линиями делят изображение на клетки одинакового размера. Чем меньше размер клетки (клетка превращается в точку) и чем их больше, тем точнее будет передано изображение и меньше потеря информации при кодировании. По ходу заметим, что при черно-белом изображении снимки в газете содержат 9 клеток в 1 кв. мм. Для наглядного представления на рис. 6 приведен увеличенный фрагмент изображения, взятый из окна графического редактора (каждая клеточка равна одному пикселю – наименьшей точке экрана), который в нормальном масштабе показан в верхнем левом углу. На втором этапе записывают в двоичном виде параметры каждой клетки: ее координаты по горизонтали и по вертикали, а также ее цвет. *Эти*

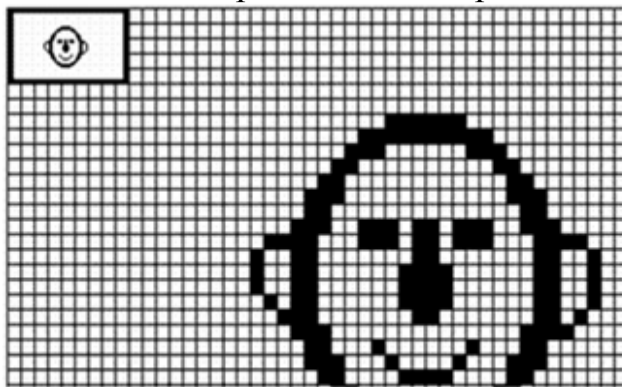


Рис. 6. Принцип кодирования рисунка отдельными точками

описания каждой клетки и будут кодами графического изображения. Такой код изображения занимает очень большой объем памяти.

Из физики известно, что любой цвет можно получить путем смешивания в различных пропорциях яркостей только трех цветов: красного (*Red*), зеленого (*Green*) и синего (*Blue*). В зависимости от используемой на компьютере видеосистемы яркость каждого из трех цветов разбивается на 16 – 256

градаций. Чем больше градаций яркости каждого цвета, тем качественнее изображение. При 256 градациях каждого из трех цветов можно получить примерно 16,78 млн. различных оттенков цвета. Для задания 16,78 млн. палитр цвета необходимо 24 двоичных разряда только для признака цвета клетки, поэтому *графическая информация очень велика по объему.*

Это же изображение можно закодировать более экономично. Поскольку номера клеток идут по порядку, то при описании изображения можно координаты клеток не указывать, а задать лишь размеры изображения (количество клеток по вертикали и горизонтали) и адреса первой и последней клеток. Естественно, что существуют стандарты такой кодировки и различные алгоритмы, позволяющие уменьшить объем графической информации.

Тем не менее обработка графической информации стала находить широкое применение только сейчас, когда оперативная и дисковая память компьютеров стала исчисляться десятками и сотнями Гбайт.

Таким образом, при растровом способе графическое изображение на экране однозначно отображается в памяти компьютера в виде определенной области с записанными двоичными кодами, описывающими каждую точку этого изображения. *В памяти получается как бы слепок изображения.* При линейном считывании памяти точки сканируются и с большой скоростью воспроизводятся на экране. Запись графической информации в память компьютера осуществляется автоматически, путем считывания с экрана координат и признаков цвета каждой точки изображения программой графического редактора, с помощью которого создается графическое изображение, или путем сканирования с листа специальным считывающим устройством – сканером. Растровый рисунок легко редактировать, он быстро воспроизводится на экране, но имеет и недостатки. *Растровая графика хранится с фиксированным разрешением и определяется количеством и цветом пикселей.* Растровый рисунок всегда привязан к конкретному устройству вывода информации. При изменении масштаба рисунка или выводе его на монитор, разрешение или цветовые возможности которого отличаются от параметров монитора, на котором рисунок создавался, возможны его искажения. Растровой графикой обычно пользуются для изображения очень мелких деталей или тонких оттенков в рисунке. Имеется множество растровых графических редакторов, например: *Adobe Photoshop, Paint Shop Pro, Microsoft Image Composer*, входящий в состав *Microsoft FrontPage* и др. Файлы растровых рисунков имеют обычно расширение *BMP, TIFF, PNG, JPEG (JPG), GIF*. Последние два формата наиболее распространены в Интернете.

В появившемся в конце XX в. новом формате представления растровых рисунков *DIB (device independent bitmap* – аппаратно-независимый битовый массив) изображение растрового рисунка не зависит от аппаратуры. В этом формате производится сжатие изображения и с ним сохраняется информация об использованной цветовой палитре и разрешении исходного устройства.

Другим способом кодирования графической информации является *векторный способ*. В *векторной графике* рисунок составляется из векторных схем – простейших графических элементов: линий, овалов, прямоугольников, окружностей и т. д. Векторная графика включает совокупность команд рисования, которые описывают положение элемента, его размеры, форму, цвет и толщину линий, вид заполнения каждого простейшего графического элемента и т.д. Например, для кодирования треугольника задаются точки координат трех его вершин, толщина и цвет линий. Таким образом, векторное изображение хранится в памяти машины в виде совокупности команд – программы, занимает значительно меньше места, но выводится медленнее, так как готового изображения в памяти нет, а оно формируется при выполнении команд, описывающих его. Такой рисунок при изменении масштаба или выводе на устройства с различной разрешающей способностью не искажается в силу того, что разрешение его не фиксировано и определяется только устройством вывода. К векторным графическим редакторам относятся: *CorelDraw, Adobe Illustrator, Micrografx Designer and Micrografx Draw*, редактор входящий в *AutoCAD, Windows Enhanced Metafile* и др.

На практике при создании графических изображений редакторы растровой, векторной и трехмерной графики часто комбинируют. Например, фотографии объектов обрабатываются и комбинируются в *Adobe Photoshop*, а на это накладываются надписи и иные рисунки, созданные с помощью *Corel Draw*, и виды трехмерных объектов, созданные в *AutoDesk 3D Studio*.

Кодирование графических изображений можно осуществлять и в виде описаний изображения в виде последовательности команд на языке программирования высокого уровня. Во всех языках программирования имеются специальные операторы рисования графических примитивов, таких как прямая линия, овал или окружность, дуга и др. Нужно в операторах задать координаты начальной и конечной точек, толщину и направление линий, точки центров окружностей и другие параметры. Составляется программа рисования изображения, которая является кодом этого изображения. Пример программы в Турбо Бейсик:

```
10 SCREEN 9
20 FOR N=1 TO 125
30 CIRCLE(160+100*COS(N/20), 90+80*SIN(N/20)), 45-N/3, INT(N/10+1)
40 NEXT
```

Эта программка размером всего 128 байт рисует цветной рог из окружностей размером в четверть экрана. Еще раз отметим, что вывод на экран закодированного таким образом изображения происходит медленнее, так как на выполнение программы, рисующей изображение, требуется какое-то время.

Кодирование звука. Звук – это воспринимаемые человеческим ухом колебания воздуха с частотами от 16 Гц до 20 КГц. В природе не бывает чистых (однотонных) звуков, то есть звуков одной частоты. Каждый звук является сложной смесью колебаний разной частоты, интенсивности и фазы и представляет собой аналоговый сигнал, который может быть преобразован в изменения напряжения или тока. Амплитуда колебаний определяет силу звука (громкость), а частота колебаний – высоту звука.

Имеется три способа кодирования звука: запись речи – письменность; запись музыки – ноты; запись произвольных звуковых колебаний – применяется в проигрывателях, магнитофонах, музыкальных центрах с компакт-дисками и т.д. При этом сигнал может быть как аналоговым, так и дискретным. Последний из этих способов и используется при кодировании звука в компьютерах.

Аналоговый сигнал, поступающий в компьютер от микрофона, проигрывателя или магнитофона, преобразуется аналого-цифровым преобразователем в дискретный сигнал. Частота дискретизации стандартизирована и составляет 44,1 КГц. Каждое измеренное значение аналогового сигнала представляется, чаще всего, 16-ю битами (в зависимости от качества звуковой платы измеренное значение может быть представлено 24 или 32 битами). Результаты преобразования в сжатом виде записываются в память компьютера. *Это также очень большая по объему информация.* Например, чтобы закодировать звуковой фрагмент звучащий 5 минут с помощью звуковой 16 битной платы, требуется памяти 12,6 Мбайт.

Другой способ кодирования звука состоит в том, что каждой ноте, высоте ее звучания, каждой октаве, паузе между нотами и т. д. присваивается определенный двоичный код. Эти коды обрабатываются специальными командами, имеющимися в каждом языке программирования высокого уровня. Поэтому звук как таковой не записывается, а составляется соответствующая программа, его воспроизводящая. Процесс аналогичен записи музыки композитором в виде нот, воспроизводимых музыкантами. *Эта программа и является своего рода кодом звукового фрагмента.*

В современных звуковых платах в виде библиотек по стандарту MIDI в ПЗУ карты записаны электронные аналоги нот. *MIDI – Musical Instruments Digital Interface – цифровой интерфейс музыкальных инструментов*, это стандартная система команд, используемая для управления электромusикальными инструментами, встроенными в звуковую карту. Запись музыки в виде MIDI-файлов в сотни раз компактнее файлов звуковых колебаний. Однако для воспроизведения музыки звуковая карта должна знать, как играет каждый музыкальный инструмент. Ведь одна и та же нота в исполнении скрипки, фортепиано, флейты, гобоя или всего оркестра звучит совершенно по-разному. А этих инструментов много: в стандарте *General MIDI* определено 128 обычных и 46 ударных инструментов. Для генерации звучания MIDI-инструментов используются, как правило, два способа: частотный и волновой синтез.

В заключение следует подчеркнуть, что, представляя любую информацию в виде двоичных кодов, мы абстрагируем ее, т. е. отвлекаемся от ее семантики, формализуем информацию. Работать с формализованной информацией значительно проще, так как можно применять одни и те же методы, способы и законы хранения, обработки и передачи ее.

Для сведения.

Имеются специальные программы типа *Score* и *Encore*, которые позволяют получать партитуру записываемой музыки. Музыка записывается с помощью музыкальной клавиатуры в виде нот, а программа проигрывает их, позволяя прослушивать музыку, оформляет и печатает партитуру, освобождая композитора от этой рутинной работы.

Фирма IBM начала промышленные поставки систем распознавания речи – *Via Voice под Windows*. Эта единственная продаваемая на рынке система позволяет пользователям диктовать намного быстрее, чем многие могут печатать на клавиатуре. Системы ориентированы на большинство языков мира (даже арабский), обладают большим словарным запасом, дают возможность говорить естественно, без пауз между словами, хранят информацию об акценте каждого пользователя, учатся словам от пользователя и накапливают их в своем словарном запасе. Система позволяет осуществлять голосовую навигацию по рабочему пространству и прямое начитывание текста в большинстве популярных приложений *Windows*.

Лекция 8

Представление чисел при обработке их в компьютере

Любая компьютерная программа оперирует определенными элементами (константами, числовыми или символьными переменными, массивами и т. д.), которые принято называть *данными*. Человек, оперируя какими-то данными, способен быстро разобраться в них и в тех операциях, которые допустимы для определенных данных. Он не будет извлекать корень квадратный из слова и не запишет число с заглавной буквы. Быстро разобраться ему в этом позволяет различие в записи **разных типов** данных. *Компьютер же любые данные воспринимает как последовательность двоичных чисел*. Поэтому ему необходимо указывать тип данных, обрабатываемых программой. В современных языках программирования в начале программы указывают (декларируют) тип данных, используемых в программе. Любой тип данных определяет три важные для транслятора (компилятора или интерпретатора) вещи:

- множество допустимых значений для заданного типа данных;
- объем памяти, выделяемый для размещения элемента программы заданного типа;
- набор допустимых действий с элементами данного типа.

Существует несколько типов данных: числовые, логические, символьные или строковые и т. д. Мы будем рассматривать только числовые данные. Код каждого символа, в том числе и любой десятичной цифры, в памяти занимает один байт, а в Unicode – два байта. Если каждую цифру числа представлять в виде кода, то для записи, например, десятизначного числа потребовалось бы 10 или 20 байт памяти. Для обработки сравнительно небольшого массива размером 1000 x 1000 таких чисел памяти уже нужно было бы до 20 Мбайт. *Это большая трата памяти*. Поэтому транслятор записывает числа по-другому – в виде разных **форматов**, в которых для представления числа используется определенное количество байт, определяемых данным форматом.

*Совокупность разрядов (бит), отводимая для представления числа определенного формата, образует **разрядную сетку** компьютера.*

Различают две формы представления чисел:

- а) естественную (с фиксированной запятой);
- б) экспоненциальную (с плавающей запятой).

В современных компьютерах естественная форма используется, как правило, для представления целых чисел. Экспоненциальная форма применяется для изображения вещественных и очень больших или маленьких целых чисел.

Машинное представление целых чисел. Обычно для представления целых чисел используют один, два или четыре байта. Существуют две формы представления: *беззнаковая* и *со знаком*. Например, в языке программирования Turbo Pascal 7.0 из пяти целочисленных типов два имеют беззнаковую форму, три – со знаком. В таблице представлены все пять указанных форматов.

Тип	Диапазон чисел	Формат
<i>ShortInt</i>	$-128 (-2^7) \dots 127 (2^7-1)$	1 байт со знаком
<i>Integer</i>	$-32768 (-2^{15}) \dots 32767 (2^{15}-1)$	2 байта со знаком
<i>LongInt</i>	$-2^{31} \dots 2^{31}-1$	4 байта со знаком

<i>Byte</i>	0 . . . 255 (2^8-1)	1 байт без знака
<i>Word</i>	0 . . . 65535 ($2^{16}-1$)	2 байта без знака

Разнообразие типов данных позволяет экономно расходовать память при разработке программ. Беззнаковое представление используется лишь для неотрицательных чисел. При этом все биты разрядной сетки отводятся под представление числа.

При представлении чисел со знаком под знак отводится старший (левый) бит разрядной сетки. Остальные разряды отводятся под само число. Для положительных чисел знаковый бит содержит 0, а для отрицательных 1. Например, под тип *Integer* отводится 2 байта со знаком, т. е. 1 разряд – под знак числа и 15 разрядов – под само число. Максимальное положительное число, представленное в 15 разрядах, равно $2^{15} - 1 = 32\,767$. Для представления же отрицательного числа используются все 16 разрядов. Минимальное отрицательное число равно: $-2^{15} = -32\,768$.

В разрядной сетке одного и того же размера можно представить беззнаковых чисел в два раза больше, чем чисел со знаком. Поэтому, если известно, что некоторая величина является неотрицательной, то выгоднее ее представить как беззнаковую.

Представление целых чисел в прямом, обратном и дополнительном кодах. Прямой код числа – это обычное изображение числа со знаком. В компьютерах прямой код для записи отрицательных чисел не используется. Во-первых, он неудобен из-за двух различных представлений нуля: +0 и -0. Во-вторых, использование прямого кода при выполнении арифметических операций привело бы к ряду дополнительных действий. Например, при сложении пришлось бы определять знаки обоих слагаемых и знак результата. При различных знаках из большего по абсолютной величине числа пришлось бы вычесть меньшее и присвоить результату знак большего числа. Это потребовало бы дополнительного выполнения нескольких машинных команд и значительного усложнения функциональных логических схем.

В связи с этим отрицательные числа представляют в компьютерах в обратном или дополнительном кодах. Для получения обратного кода число следует *инвертировать* (заменить 1 на 0, а 0 – на 1) за исключением знакового разряда. *В обратном коде, как и в прямом, ноль имеет также два представления.*

Поэтому в основном используется дополнительный код, для получения которого к младшему разряду обратного кода следует добавить единицу. При представлении отрицательных чисел в дополнительном коде имеется единственное изображение нуля, а все арифметические операции сводятся к поразрядному сложению и сдвигам. Это связано с тем, что логические схемы арифметического устройства, в частности сумматора, состоят из двоичных переключателей, на которых двоичные слагаемые представляются состояниями переключателей. Поэтому можно выполнять только сложение двоичных чисел, т. е. накапливать итоговое состояние ключей при подаче на них слагаемых (последова-

тельностью импульсов). Умножение сводится к многократному сдвигу и сложению, деление сводится к операциям сравнения и вычитания, а вычитание – к сложению в обратном или дополнительном кодах. Это позволяет для реализации всех арифметических операций использовать лишь одно устройство – *сумматор*. Положительные числа во всех трех кодах имеют одинаковый вид.

В дополнительном коде представление положительного нуля совпадает с прямым кодом. В разрядной сетке длиной 1 байт положительный ноль будет таким: 0 0 0 0 0 0 0 0. Для получения дополнительного кода отрицательного нуля следует к его обратному коду добавить единицу в младший разряд:

$$\begin{array}{r} 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\ + \quad \quad \quad \quad \quad 1 \\ \hline \boxed{1}0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \end{array}$$

Полученная единица выходит за разрядную сетку и теряется. Таким образом, ноль в дополнительном коде имеет *единственное* представление.

Пусть разрядная сетка составляет один байт. Представим число -11 в прямом, обратном и дополнительном кодах.

1 0 0 0 1 0 1 1 Прямой код числа -11. В крайнем левом разряде *1* указывает, что число отрицательное.

1 1 1 1 0 1 0 0 Обратный код числа -11. Выполнили инверсию (кроме знака).

1 1 1 1 0 1 0 1 Дополнительный код числа -11. К младшему разряду обратного кода добавлена единица.

В дополнительном коде число -11 интерпретируется следующим образом: $-2^7 + 2^6 + 2^5 + 2^4 + 2^2 + 2^0 = -128 + 64 + 32 + 16 + 4 + 1 = -128 + 117 = -11$.

Отсюда следует, что в разрядах с нулевого по шестой записанное число 117 является дополнением модуля числа -11 до переполнения разрядной сетки, то есть до числа 128. Поэтому код и называется *дополнительным*.

При сложении положительного и отрицательного чисел в обратном и дополнительном кодах, если получаемый результат является *положительным*, то он представляется *в прямом коде*, то есть таким, каким должен быть, а если *отрицательным*, то – *в обратном коде*. После инвертирования обратного кода получается прямой код результата.

Покажем это на примерах.

Пусть разрядная сетка длиной 1 байт и крайний левый бит будет знаковым.

При положительном результате: 34 – 19 = 15.

В прямом коде: В обратном коде: В дополнительном коде:

$$\begin{array}{r} 00100010 \\ -00010011 \\ \hline 00001111 \end{array} \quad \begin{array}{r} 00100010 \\ +11101100 \\ \hline 100001110 \\ \boxed{1} \xrightarrow{+} \quad \quad \quad 1 \\ \hline 00001111 \end{array} \quad \begin{array}{r} 00100010 \\ +11101101 \\ \hline \boxed{1}00001111 \end{array}$$

*В обратном коде единица, вышедшая за разрядную сетку, переносится в младший разряд и добавляется к результату. В дополнительном коде эта единица теряется – происходит переполнение разрядной сетки. За переносами и переполнением разрядной сетки следят индикаторы переноса и переполнения. Индикаторами являются биты специального регистра центрального процессора, который носит название **регистра флажков**. В зависимости от состояния*

флажков центральное устройство управления, которое следит за их состоянием, осуществляет соответственно операции переноса и переполнения.

Пусть результат будет отрицательным: $19 - 34 = -15$.

В прямом коде операция не реализуема.

В обратном коде:

$$\begin{array}{r} 00010011 \\ + 11011110 \\ \hline 11110000 \end{array}$$

В дополнительном коде:

$$\begin{array}{r} 00010011 \\ + 11011110 \\ \hline 11110001 \\ - \quad \quad 1 \\ \hline 11110000 \end{array}$$

В дополнительном коде из полученного результата вычитаем единицу из младшего разряда, чтобы получить обратный код (мы ее добавляли к обратному коду отрицательного числа для получения дополнительного кода).

Выполнив инвертирование результатов сложения, получаем результат в прямом коде со знаком минус, о чем говорит единица в знаковом разряде.

Аналогично можно показать выполнение этих операций и в десятичной системе счисления.

Машинное представление вещественных чисел. Для изображения вещественных чисел, а также очень маленьких и очень больших целых чисел используется экспоненциальная форма представления, например: $62,478 \cdot 10^5$; $-0,23 \cdot 10^{-10}$. Записывается это следующим образом: $62,478E+05$; $-0,23E-10$, где символ E читается как «умножить на десять в степени». При такой записи: $478 \cdot 10^2 = 47,8 \cdot 10^3 = 4,48 \cdot 10^4 = 0,478 \cdot 10^5$ число не изменяется по величине, а запятая в числе как бы плавает из одной позиции в другую, поэтому такие числа еще называют «числа с плавающей запятой». Для однозначности представления чисел, а также с целью минимизации погрешности при вычислениях и эффективного использования памяти в компьютерах к таким числам применяют процедуру **нормализации справа**. Число называют нормализованным справа, если целая часть числа равна нулю, а после запятой стоит не нуль. Нормализованная форма представления числа имеет следующий вид: $N = \pm m \cdot q^p$.

Здесь: m – мантисса числа, равная $q^{-1} \leq |m| < 1$;

q – основание системы счисления;

p – порядок числа.

При $q = 10$ $0.1 \leq |m| < 1$. При $q = 2$ $0.5 \leq |m| < 1$.

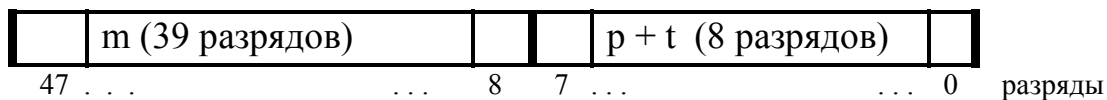
Для примера возьмем снова язык программирования Turbo Pascal 7.0. В нем существует пять стандартных типов для вещественных чисел, приведенных в таблице, из которых наиболее распространен тип *Real*.

Тип	Диапазон	Значащие цифры	Формат
<i>Real</i>	$1.47 \cdot 10^{-39} \dots 1.7 \cdot 10^{38}$	11 – 12	6 байт
<i>Single</i>	$1.5 \cdot 10^{-45} \dots 3.4 \cdot 10^{38}$	7 – 8	4 байта
<i>Double</i>	$5.0 \cdot 10^{-324} \dots 1.7 \cdot 10^{308}$	15 -16	8 байт
<i>Extended</i>	$3.4 \cdot 10^{-4932} \dots 1.1 \cdot 10^{4932}$	19 -20	10 байт
<i>Comp</i>	$-2^{63} + 1 \dots 2^{63} - 1$	19 -20	8 байт

Основное различие между типами состоит в числе разрядов, отводимых под переменную определенного типа. Поэтому длина мантииссы и порядка в различных типах не одинаковы. Чем больше длина, тем выше точность вычислений, но тем больше расход оперативной памяти, отводимой под описание переменных. Переменные типов *Single*, *Double* и *Extended* рассчитаны на аппаратную поддержку арифметики с плавающей запятой (сопроцессор, или математический процессор – специализированная логическая схема, ранее устанавливаемая на системной плате в виде отдельной микросхемы. В современных компьютерах она выполнена совместно с центральным процессором в одной микросхеме и работает в 10 – 50 раз быстрее его). Тип *Real* оптимизирован для работы без сопроцессора. Поэтому если установлен сопроцессор, то тип *Real* будет преобразовываться к *Extended*, а это ведет к увеличению расходуемой памяти. При разработке программ, критичных ко времени счета, следует использовать типы *Single* или *Double*. В этом случае скорость счета с сопроцессором увеличится в 2 – 5 раз.

При представлении чисел с плавающей запятой часть разрядной сетки отводится для записи порядка числа, остальные – для записи мантииссы. По одному разряду в каждой группе отводится под знак – порядка и мантииссы.

Для примера рассмотрим схему представления чисел для переменных типа *Real* в компьютерах без сопроцессора. Переменная такого типа занимает в оперативной памяти 6 байт, или 48 разрядов. Они распределены следующим образом:



В сопроцессорных вещественных типах мантиисса и порядок меняются местами, за исключением знакового разряда, а принципиальное отличие их типа *Real* заключается в основном лишь в количестве разрядов, отводимых под порядок и мантииссу.

Любое число X типа *Real* в двоичной форме имеет следующий нормализованный вид: $X = \pm 0.1 \cdot m \cdot 2^p$, $0 \leq m < 2^{39}$, $-128 \leq p \leq 127$.

Величина мантииссы равна $0.1 \cdot m$, так как после запятой всегда стоит единица. В записи числа она не участвует, но учитывается при выполнении арифметических операций.

Величина $t = 128$ – это смещение порядка p . Оно введено для получения положительного порядка при любых числах. При записи числа с минимально возможным порядком, равным -128 , все разряды, отводимые под порядок, будут содержать нули: $-128 + 128 = 0$. Для числа с максимально допустимым порядком, равным 127 , будет $p + t = 127 + 128 = 255$, т. е. все разряды содержат единицы. Таким образом, порядок для переменных *Real* всегда представляется положительным беззнаковым числом.

В разрядной сетке на месте мантииссы размещается ее модуль, а знак ее, как и знак всего числа, определяется значением 47-го разряда: 0 для положительных чисел и 1 – для отрицательных чисел.

Минимальное значение мантииссы равно: 0.1_2 . При этом все 39 разрядов, отведенных под мантииссу, содержат нули. Максимальное же значение мантииссы равно: $0.1 \underbrace{(\dots 1)}_{39 \text{ разрядов}} = 2^{-1} + 2^{-2} + \dots + 2^{-40} = 1 - 2^{-40}$.

Машинным нулем для переменной типа *Real* называется неотрицательное число, равное: $0.1_2 \cdot 2^{-128}$. При его записи все 48 разрядов содержат нули.

При представлении порядка со смещением ($p + t$) упрощается операция сравнения произвольного числа с нулем, которая реализована аппаратно и выполняется в программах довольно часто. Однако положительный нуль (+0) и отрицательный нуль (-0) оказываются различными. У отрицательного нуля в старшем 47 разряде записана 1, а в остальных – нули. Поэтому при программировании сравнивать на точное равенство вещественные числа в большинстве случаев некорректно. Вместо $X - Y = 0$, следует писать $|X - Y| < \varepsilon$, где ε – достаточно малое число, соответствующее погрешности вычислений.

Максимально представимое число типа *Real* равно:
 $(1 - 2^{-40}) \cdot 2^{127} = 2^{127} - 2^{87} = 2^{(10 \times 12.7)} - 2^{(10 \times 8.7)} = 1024^{12.7} - 1024^{8.7} = 1,7 \cdot 10^{38}$.

Минимально представимое число равно:
 $0.1 \cdot 2^{-128} = 1 \cdot 2^{-129} = 2^{(-10 \times 12.9)} = 1024^{-12.9} = 1,47 \cdot 10^{-39}$.

Арифметика нормализованных чисел. Суть операций сложения и вычитания состоит в выравнивании порядков чисел до большего, сложении либо вычитании мантиисс и присваивании результату соответствующего порядка, при этом мантииссы представляются как **целые числа**. При вычитании меньшая по модулю мантиисса представляется в обратном коде, и производится их сложение. Знак результата будет таким, как у большей по модулю мантииссы. Примеры сложения и вычитания для наглядности покажем в десятичной системе счисления. Пусть разрядная сетка равна заданной длине мантииссы (6-ти разрядам):

$$C = A + B = 0.397621 \cdot 10^3 + 0.237912 \cdot 10^1$$

Порядок второго слагаемого выравнивается до большего: $0.237912 \cdot 10^1 = 0.002379 \cdot 10^3$

$$C = A + B = (0.397621 + 0.002379) \cdot 10^3 = 0.4 \cdot 10^3 = 400$$

$$C = A - B = 0.645623 \cdot 10^{-2} - 0.254811 \cdot 10^2$$

Выровняем порядок 1-го числа: $0.645623 \cdot 10^{-2} = 0.0000645623 \cdot 10^2 \approx 0.000065 \cdot 10^2$

$$C = A - B = (0.000065 - 0.254811) \cdot 10^2 = -0.254746 \cdot 10^2 = -25.4746$$

В двоичной системе счисления модуль мантииссы вычитаемого равен 111110001101011011, а модуль мантииссы уменьшаемого равен 00000000001000001. Эта мантиисса по модулю меньше мантииссы вычитаемого, поэтому ее представим в обратном коде 11111111110111110 и сложим мантииссы:

$$\begin{array}{r} 111110001101011011 \\ + 11111111110111110 \\ \hline \boxed{1}111110001100011001 \\ \hline \rightarrow + 1 \\ \hline 111110001100011010 = 254746 \end{array}$$

Получаем результирующую мантииссу 0,254746. Присвоим знак (-) большей по модулю мантииссы и добавим общий порядок. Получим результат: -25.4746.

При умножении нормализованных чисел следует сложить порядки, перемножить мантииссы и результату присвоить суммарный порядок.

При делении нормализованных чисел следует из порядка делимого вычесть порядок делителя, а над мантииссами выполнить обычную операцию деления.

После выполнения операций при необходимости нужно провести нормализацию результатов, что приводит к изменению порядков, так как каждый сдвиг на один разряд влево уменьшает порядок на единицу, а сдвиг вправо – увеличивает его на единицу. Введение термина «плавающая запятая» как раз и объясняется тем, что двоичный порядок, определяющий фактическое положение запятой в числе, корректируется после выполнения каждой арифметической операции, т. е. запятая в изображении числа плавает по мере изменения его.

Таким образом, в компьютере все арифметические операции сводятся к серии сложений и сдвигов двоичных чисел. Для чисел с плавающей запятой они значительно сложнее и выполняются медленнее, чем для целых чисел. Поэтому и используется дополнительно специальный математический сопроцессор, который ускоряет выполнение этих операций в 10 – 50 раз.

При работе с вещественными числами следует учитывать следующие особенности.

1. Ошибки округления возникают уже на этапе записи чисел в компьютер. При выполнении арифметических операций они возрастают. Рассмотрим для простоты действий примеры в десятичной системе счисления: пусть под мантиссы отведено 5 разрядов $X = 0.87654 \cdot 10^1$; $Y = 0.94567 \cdot 10^2$. После выравнивания порядка до большего получим $X = 0.08765 \cdot 10^2$. Последнюю цифру 4 потеряли. Далее просуммируем мантиссы:

$$\begin{array}{r} 0.08765 \\ + 0.94567 \\ \hline 1.03332 \end{array}$$

Получили мантиссу больше 1, поэтому произойдет нормализация ее вправо, т. е. у результата порядок возрастет на единицу при сдвиге вправо, а последняя цифра 2 пропадет: $0.10333 \cdot 10^3$. Итого потеряли две последние цифры 2 и 4.

2. Добавление или вычитание малого числа может не сказаться на результате. Пример: $X = 0.32165 \cdot 10^3$; $Y = 0.48434 \cdot 10^{-3}$. После выравнивания порядков: $Y = 0.00000048434 \cdot 10^3$. Так как под мантиссу у нас отведено 5 разрядов, то при сложении мантисс у второго слагаемого последние значащие цифры пропадут и результат будет равен первому слагаемому.

3. Получение очень больших чисел может вызвать переполнение порядка, а очень малых – отрицательное переполнение или превращение его в ноль. Это может привести к аварийному завершению программы.

4. При вычитании близких по модулю чисел могут возникнуть ненадежные значащие цифры, приводящие к серьезной потере точности.

Изначальной причиной возникновения погрешностей обработки чисел является ограниченность разрядной сетки.

У целых чисел появляется понятие максимального числа, хотя на числа ограничений в принципе нет. По этой же причине числовая система ЭВМ циклична. Пусть разрядная сетка имеет 4 разряда, в которых помещается максимальное число 1111 и минимальное число 0000. Последовательно прибавляя по 1 к минимальному числу, получим 1111, а затем снова 0000.

Из математики известно, что множество вещественных чисел на числовой оси упорядоченное, всюду плотное и непрерывное, т. е. каждая точка числовой оси имеет действительную координату. Поэтому вещественные числа в компь-

ютере заменяются их кодами, которые образуют конечное *дискретное множество*. Из этого положения вытекают следующие выводы:

- строгие отношения между числами превращаются в нестрогие для их компьютерных кодов;
- поскольку код вещественного числа в компьютере является *приблизительным представителем многих чисел из интервала*, то и результаты вычислений также будут заведомо неточными. Это главная особенность обработки вещественных чисел;
- появляется понятие «машинного нуля», числа меньшего некоторой определенной величины в зависимости от разрядной сетки.

Лекция 9

Понятие об алгебре логики

Для дальнейшего рассмотрения вопросов обработки информации большое значение имеет знакомство с *алгеброй логики* – одной из основных частей математической логики, основанной на применении алгебраических методов к логике. (Математическая логика – это наука, изучающая предмет формальной или «обычной» логики методом формализованных языков.)

Алгебра логики является одним из основных инструментов в теории релейных схем, в теории ЭВМ и дискретных автоматов, в прикладной информатике и кибернетике. Она представляет собой, прежде всего, *алгебру высказываний*. В алгебре логики под высказыванием понимают всякое предложение, которое либо истинно (имеет значение *T-true*), либо ложно (имеет значение *F-false*). Отдельные высказывания обозначаются буквами какого-либо алфавита, например: *A, B, C, D...* Истинность или ложность высказываний называется *их значениями истинности*. Истинность высказывания отождествляют с числом 1, а ложность – с числом 0. Запись $A = 1, C = 0$ означает, что высказывание *A* истинно (например, $A =$ Екатеринбург – крупный промышленный центр Урала), а *C* – ложно (например, $C =$ снег черный).

Предметом изучения алгебры логики являются двоичные функции, т. е. функции, которые в результате выполнения принимают только два значения: 1 – истинно и 0 – ложно.

Из одного или нескольких высказываний, принимаемых за простые, можно составлять сложные высказывания, которые будут являться логическими или двоичными функциями простых высказываний. Объединение простых высказываний в сложные производится без учета внутреннего содержания (смысла) этих высказываний, а только по признаку истинности. К числу основных логических операций для получения сложных высказываний относятся операции отрицания, конъюнкции, дизъюнкции, эквивалентности и импликации. Если

под переменными A, B, C , и т. д. понимать не только высказывания, а вообще любую систему элементов, для которой определены действия логического умножения, сложения и отрицания и которая удовлетворяет основным законам алгебры логики, то получим абстрактную алгебру, называемую *алгеброй Буля*, названную в честь ирландского математика Д. Буля. В частности, высказывания и основные логические операции представляют собой частный случай алгебры Буля. Поэтому в теории алгоритмов и ЭВМ, в теории релейных схем, в теории двоичных переключательных схем и автоматов и т. д. чаще говорят не «алгебра логики», а «алгебра Буля» и не «двоичная логическая функция», а «булева функция» или «переключательная функция».

Булева алгебра позволяет не только описывать логические связи между различными узлами цифровых устройств, но и преобразовывать их, делая возможным реализацию устройств на функционально различных типах элементарных схем.

Поскольку аргументы переключательной функции могут принимать (как и она сама) только два значения 0 и 1, то область определения любой переключательной функции всегда конечна. Совокупность значений истинности аргументов называется *набором*. Если имеется n переменных (аргументов функции – простых высказываний), то существует $Z = 2^n$ различных наборов значений их истинности. Так как переключательная функция (сложное высказывание) определена на этих наборах и сама может принимать значения 0 или 1, то число различных булевых функций M от n переменных равно $M = 2^Z = 2^{2^n}$. Например, пусть имеем функции от одного аргумента, т. е. $n = 1$, тогда $Z = 2^1 = 2$ набора истинности аргумента – 1 и 0 и количество *сингулярных* функций (функций от одного аргумента) будет равно $M = 2^Z = 2^2 = 4$.

Основные логические операции. Рассмотрим кратко основные логические операции и получаемые с их помощью логические функции.

1. *Отрицание, или инверсия*, является функцией одной переменной, т. е. отрицание, одна из четырех сингулярных функций, может иметь также два значения: если $X = 1$, то $f(X) = \bar{X} = 0$ и, если $X = 0$, то $f(X) = \bar{X} = 1$. Читается «не X», пишется – *NOT*. Это высказывание противоположное *по значению* исходному. Обозначается операция чертой над аргументом. Аргументом отрицания может быть и сложное высказывание, т. е. другая булева функция. Это мы увидим чуть дальше по тексту. Три других сингулярных функций – это сама величина X , т. е. $f(X) = X$, и две константы, не зависящие от значения X : $f(X) = 1$ и $f(X) = 0$.

При $n = 2$ имеем четыре набора значений их истинности. Следовательно, булевых функций от двух аргументов (*бинарных*) будет 16. Бинарные переключательные функции получаются с помощью следующих логических операций.

2. *Конъюнкция, или логическое умножение*, обозначается значками \wedge , $\&$ или точкой и читается как «И» (*AND*). Получаемая функция $f(X_1, X_2) = X_1 \wedge X_2 = X_1 \cdot X_2$ (читается « X_1 и X_2 ») имеет значение истинности (1) только тогда, когда аргументы X_1 и X_2 истинны, т. е. при $X_1 = 1$ и $X_2 = 1$ функция $f(X_1, X_2) = 1$. Во всех других случаях она имеет значение ложности (0). Аргументами конъюнк-

ции могут быть другие переключательные функции. Сама конъюнкция также может быть аргументом другой логической функции. Например, если имеем функцию, $Y = X_1 \wedge X_2$, то, применив отрицание, получим $f(Y) = \overline{Y} = \overline{X_1 \wedge X_2}$. Кроме того, аргументов функции может быть не обязательно два, а больше, например: $f(X_1, X_2, X_3, X_4) = X_1 \wedge X_2 \wedge X_3 \wedge X_4$.

3. *Дизъюнкция, или логическое сложение*, обозначается значком \vee или $+$ и читается как «или» (*OR*). Это *неразделительный* союз «или» (при звоне будильника Дима или Коля проснутся – не разделяется кто: или тот, или другой, или оба вместе). Есть еще *исключающее* «или» (выбирай: он или я). Этот союз не может быть дизъюнкцией, хотя другой логической операцией может быть. Получаемая функция $f(X_1, X_2) = X_1 \vee X_2 = X_1 + X_2$ читается « X_1 или X_2 » и имеет значение ложности только тогда, когда X_1 и X_2 ложны, т. е. при $X_1 = 0$ и $X_2 = 0$ функция $f(X_1, X_2) = 0$. Во всех остальных случаях она истинна. Аргументами дизъюнкции также могут быть другие переключательные функции. Например, если $Y = X_1 \wedge X_2$, а $Z = \overline{X_3}$, то $f(Y, Z) = Y \vee Z = (X_1 \wedge X_2) \vee \overline{X_3}$. Сама дизъюнкция также может быть аргументом другой логической функции.

4. *Эквивалентность*. С помощью этой логической операции может быть получена булева функция, имеющая значение истинности тогда, когда значения истинности ее аргументов одинаковы, и значение ложности – во всех остальных случаях. Эта логическая операция обозначается значками \sim или \Leftrightarrow и читается « X_1 эквивалентно X_2 ». Если $X_1 = X_2 = 1$ или $X_1 = X_2 = 0$, то функция $f(X_1, X_2) = 1$; при $X_1 = 1$ и $X_2 = 0$ или $X_1 = 0$ и $X_2 = 1$ функция $f(X_1, X_2) = 0$. Для эквивалентности справедливы выражения: $A \sim 1 = A$ и $A \sim 0 = \overline{A}$.

5. *Импликация*. С помощью этой логической операции может быть получена булева функция, имеющая ложное значение в том и только в том случае, когда аргументы X_1 – истинно, а X_2 – ложно. При $X_1 = 1$ и $X_2 = 0$ функция $f(X_1, X_2) = X_1 \rightarrow X_2 = 0$. Во всех остальных случаях ее значение истинно. Обозначается эта функция значком \rightarrow и читается как «если X_1 , то X_2 ». Другими словами « X_1 ложно или X_2 истинно». Здесь союз «или» – не исключающее.

Рассмотрим еще три важнейшие переключательные функции, получаемые путем применения отрицания к рассмотренным выше операциям.

6. Применив операцию отрицания к функции эквивалентности, получим новую логическую функцию $f(X_1, X_2) = \overline{X_1 \sim X_2}$. Нетрудно видеть, что она имеет смысл исключающего «или» (для проверки подставьте значения истинности). Эта логическая операция имеет важнейшее значение в теории и практике ЭВМ, так как представляет собой *сложение двоичных чисел по модулю два* (без переносов из разряда в соседний разряд) и обозначается значком \oplus , т. е.: $f(X_1, X_2) = \overline{X_1 \sim X_2} = X_1 \oplus X_2$.

7. Применив операцию отрицания к функции, полученной путем конъюнкции $f(X_1, X_2) = X_1 \wedge X_2$, получим новую булеву функцию – *штрих Шеффера*

$f(X_1, X_2) = \overline{X_1 \wedge X_2} = X_1 | X_2 = X_1 \uparrow X_2$. Эта логическая операция имеет важнейшее значение в теории и практике ЭВМ и логических схем, так как через нее может быть выражена любая функция Буля. Электронная схема, реализующая ее, является функциональным универсальным элементом, при помощи которого в принципе могут быть построены любые функциональные схемы автоматов.

8. Применяя операцию отрицания к функции, полученной путем дизъюнкции $f(X_1, X_2) = \overline{X_1 \vee X_2}$, получим новую логическую функцию – *стрелка Пирса* $f(X_1, X_2) = \overline{X_1 \vee X_2} = X_1 \downarrow X_2$.

Остальные бинарные функции рассматривать не будем, так как они не имеют большого практического применения.

Преобразование логических функций. Из рассмотренного выше возникает вопрос: возможен ли минимальный набор простых булевых функций, с помощью которых можно представить все функции алгебры Буля или сколь угодно сложную булеву функцию? Этот вопрос связан с одним из основных понятий булевой алгебры – понятием *функциональной полноты системы булевых функций*. Оказывается, можно привести несколько систем булевых функций, обладающих функциональной полнотой. Одной из таких систем является система, состоящая из операций конъюнкции, дизъюнкции и отрицания. Из нее можно получить еще две системы, исключая поочередно функции «конъюнкция» и «дизъюнкция». Наконец, можно привести несколько систем, состоящих всего лишь из одной функции, которые будут обладать также функциональной полнотой. Это функции штрих Шеффера и стрелка Пирса. Недостающие функции конъюнкции, дизъюнкции и отрицания для этих систем можно получить на основе основных правил алгебры логики, приводимых ниже.

Правила приводятся без доказательств в качестве справочных данных, но легко можно убедиться в их справедливости, подставив в правые и левые части равенств значения переменных, равные 0 и 1. Эти правила используются для преобразования сложных логических выражений к более удобному или простому виду.

1. $\overline{\overline{A}} = A$; 2. $A \wedge A = A$; 3. $A \vee A = A$; 4. $A \wedge 1 = A$; 5. $A \vee 1 = 1$;
6. $A \vee \overline{A} = 1$; 7. $A \wedge \overline{A} = 0$; 8. $A \wedge 0 = 0$ 9. $A \vee 0 = A$;
10. $A \wedge B = B \wedge A$; 11. $A \vee B = B \vee A$; 12. $(A \wedge B) \wedge C = A \wedge (B \wedge C)$;
13. $(A \vee B) \vee C = A \vee (B \vee C)$;
14. $\overline{(A \vee B)} = \overline{A} \wedge \overline{B} = A \downarrow B$; 15. $\overline{(A \wedge B)} = \overline{A} \vee \overline{B} = A \uparrow B$;
16. $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$; 17. $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$;

Соотношения 10, 11, 12, 13 показывают, что для операций конъюнкции и дизъюнкции справедливы *переместительные* (10, 11) и *сочетательные* (12, 13) законы, в силу чего многочисленные конъюнкции и дизъюнкции, например $[(A \wedge B) \wedge C] \wedge D = A \wedge B \wedge C \wedge D$, можно писать без скобок. Такое выражение часто называют произведением, а члены его – множителями. Выражение вида $A \vee B \vee C \vee D \dots$ называют суммой, а члены его – слагаемыми. Соотношение 17 показывает, что в алгебре логики, как и в арифметике, справедлив распределительный закон.

тельный закон умножения относительно сложения – закон *дистрибутивности*. В отличие от арифметики, в алгебре логики есть еще распределительный закон сложения относительно умножения, выражаемый соотношением **16**. Соотношения **14** и **15** называются *законами де Моргана* (или законами инверсии). Они вместе с соотношением **1** позволяют преобразовывать логические выражения в такой вид, что операции отрицания будут относиться только к простым высказываниям (аргументам булевых функций). Соотношения **3** и **4** – законы *тождества* для сложения и умножения. Соотношение **6** – закон *исключенного третьего*, а **7** – закон *противоречия*.

Весьма полезными для преобразования логических выражений являются следующие формулы, приводимые также без доказательства (нумерация формул продолжается). Использование соотношений **28** ÷ **37** позволяет любые выражения, содержащие знаки эквивалентности, импликации и штрих Шеффера, приводить к выражениям, содержащим только знаки конъюнкции, дизъюнкции и инверсии.

- | | |
|---|---|
| 18. $A \vee (\bar{A} \wedge B) = A \vee B;$ | 19. $A \wedge (\bar{A} \vee B) = A \wedge B;$ |
| 20. $\bar{A} \vee (A \wedge B) = \bar{A} \vee B;$ | 21. $\bar{A} \wedge (A \vee B) = \bar{A} \wedge B;$ |
| 22. $A \vee (A \wedge B) = A;$ | 23. $A \wedge (A \vee B) = A;$ |
| 24. $(A \wedge B) \vee (A \wedge \bar{B}) = A;$ | 25. $(A \vee B) \wedge (A \vee \bar{B}) = A;$ |
| 26. $(A \wedge B) \vee (\bar{A} \wedge C) = (A \wedge B) \vee (\bar{A} \wedge C) \vee (B \wedge C);$ | |
| 27. $(A \vee B) \wedge (\bar{A} \vee C) = (A \vee B) \wedge (\bar{A} \vee C) \wedge (B \vee C);$ | |
| 28. $A \sim B = A \wedge B \vee \bar{A} \wedge \bar{B};$ | 29. $\overline{A \sim B} = A \wedge \bar{B} \vee \bar{A} \wedge B = A \oplus B;$ |
| 30. $A \rightarrow B = \bar{A} \vee B = \overline{A \wedge \bar{B}};$ | 31. $\overline{A \rightarrow B} = A \wedge \bar{B};$ |
| 32. $A \rightarrow \bar{B} = \bar{A} \vee \bar{B};$ | |
| 33. $\bar{A} \rightarrow B = A \vee B;$ | 34. $A \uparrow \bar{A} = A;$ |
| 35. $A \uparrow B = A \wedge B;$ | |
| 36. $A \uparrow B = \overline{A \wedge B} = \bar{A} \vee \bar{B};$ | 37. $\bar{A} \uparrow \bar{B} = A \vee B;$ |

Применение алгебры логики. Перечисленные в предыдущем разделе правила преобразования логических выражений далеко не исчерпывают всех возможных булевых равенств. Они являются основными, и их знание необходимо для овладения техникой преобразования булевых функций. Использование аппарата алгебры логики в теории устройств дискретного действия открывает возможности для анализа и синтеза переключательных схем, являющихся основой построения всех компьютеров.

Простейшие электронные схемы (интегральные схемы – ИС), реализующие элементарные булевы функции, называются *логическими элементами* и являются *элементной базой дискретных устройств*. При описании цифровых устройств на уровне функциональных схем и при описании цифровых интегральных схем логический элемент обозначается прямоугольником, внутри которого указывается символ выполняемой функции. Инверсия по входу (выходу) функционального элемента обозначается кружком на входе (выходе) прямоугольника. При этом логическая операция отрицания применительно к преобразованию сигналов означает, что сигнал на выходе элемента **НЕ** будет тогда,

когда не будет сигналов на его входах, и наоборот. Для логической операции конъюнкции функциональный элемент обозначается знаком $\&$ в верхнем левом углу прямоугольника. Сигнал на выходе элемента **И** будет только тогда, когда сигналы будут одновременно на всех его входах. Если хотя бы одного входного сигнала не будет, то на выходе сигнала тоже не будет. Для логической операции дизъюнкции функциональный элемент обозначается цифрой 1 в верхнем левом углу прямоугольника. Сигнала на выходе элемента **ИЛИ** не будет, если не будет ни одного входного сигнала. Если будет хотя бы один из всех входных сигналов, будет сигнал и на выходе элемента.

В настоящее время промышленность выпускает большое количество *серий* интегральных микросхем самого различного назначения, которые можно найти в любом справочнике по цифровым бытовым и промышленным интегральным схемам.

Задача *анализа* схем состоит в том, что, имея какую-либо готовую схему, требуется описать ее работу логическим выражением, задающим некоторую функцию алгебры логики. Затем, путем преобразования ее, исследовать вопрос: нельзя ли получить более простую схему, содержащую меньшее количество элементов, которая бы реализовала данную функцию алгебры логики? Это очень важный вопрос. Чем меньше элементов содержит схема, тем она надежнее работает, потребляет меньше энергии, конструктивно более компактна, более проста в изготовлении и т. д., т. е. менее затратна.

Рассмотрим пример. Пусть имеем фрагмент некоторой логической схемы, изображенный на рис. 7 а.

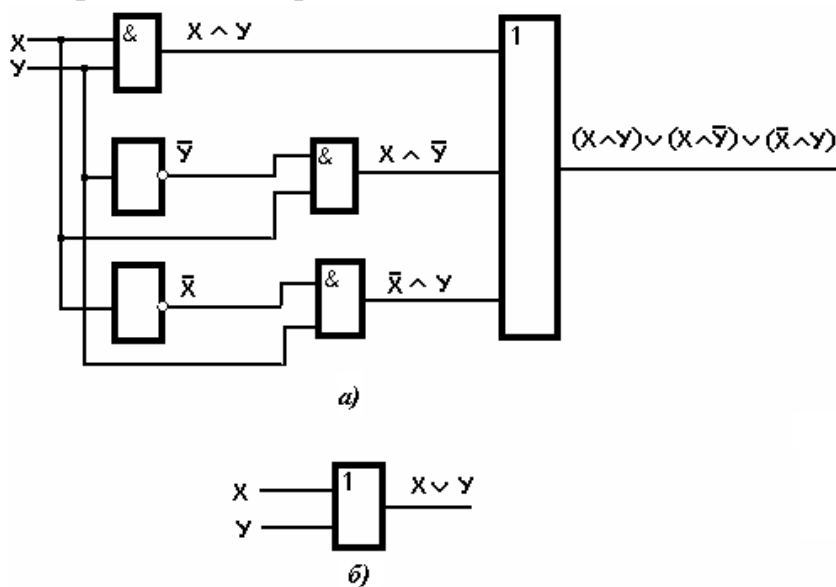


Рис. 7. Пример упрощения логической схемы

Проанализируем этот фрагмент с целью выяснить: нельзя ли схему упростить, сохраняя ее функциональные возможности? Запишем значение выходной функции в соответствии с логическими элементами в виде следующего логического выражения:

$$(X \wedge Y) \vee (X \wedge \bar{Y}) \vee (\bar{X} \wedge Y)$$

Применяя приведенные ранее соотношения 4, 6, 17 к первым двум скобкам, получим следующее логическое выражение:

$(X \wedge Y) \vee (X \wedge \bar{Y}) = X \wedge (Y \vee \bar{Y}) = X$. Подставив полученное значение в исходное выражение, и применяя снова соотношения 4, 6, 16, получим окончательный результат в виде: $X \vee (\bar{X} \wedge Y) = (X \vee \bar{X}) \wedge (X \vee Y) = X \vee Y$. Таким образом, упростив выходное логическое выражение, видим, что приведенный на рис. 7 (а)

фрагмент схемы может быть представлен в виде одного простейшего логического элемента *ИЛИ*, изображенного на рис. 7 б.

Задача *синтеза* состоит в том, что, имея логическое выражение, описывающее некоторую логическую функцию, необходимо определить, из каких элементарных логических операций она состоит, чтобы знать, какие логические элементы необходимы для построения схемы, реализующей заданное выражение, и как они должны быть объединены. Для этого заданное логическое выражение преобразуют и расчлениают на отдельные элементарные функции так, чтобы число их было минимальным.

Методы алгебры логики также находят широкое применение в теории множеств и при составлении логических схем программ. Конъюнкция часто используется для конкатенации – слияния двух выражений (обычно строковых), чтобы образовать одно более длинное выражение. Часто ее используют для выделения определенной части двоичного кода с помощью специально подобранных кодов из нулей и единиц, чтобы проверить какое-либо условие. Например: по выражению *если X AND 8=1 то* проверяется третий разряд переменной *X* на равенство 1 и выполняется какое-то условие. Здесь слово *AND* является логическим умножением.

Дизъюнкция используется для формирования двоичного кода из двух имеющихся кодов (чаще всего выделенных с помощью логического умножения).

Сложение по модулю два используется как операция сравнения двух чисел одинаковой длины. Если числа тождественны, то результат будет представлять набор нулей.

В систему команд любого компьютера обязательно входит полный набор логических команд.

Электронные схемы логических элементов. Кратко рассмотрим, на каких электронных схемах и как реализуются логические функции. В логических схемах двоичных переключательных автоматов, в частности в компьютере, используются в основном полупроводниковые диоды и транзисторы из-за наличия у них резко выраженных переключающих свойств (открыт – проводит ток, закрыт – ток не проводит).

Сначала познакомимся с работой транзистора. На рис. 8 приведена схема, реализующая логическую операцию «NOT» (отрицания). При отсутствии $U_{вх}$

транзистор T закрыт напряжением смещения на базе $+E_{б}$. Ток через него не проходит и $U_{вых} = -E_{к}$. Если подать на вход отрицательный сигнал $|U_{вх}| > E_{б}$, возникает ток базы и, следовательно, ток коллектора. На делителе напряжения (сопротивление $R_{к}$ – транзистор) протекающий ток большую часть $-E_{к}$ создаст на $R_{к}$, а на открытом транзисторе, с которого снимается выходное напряжение,

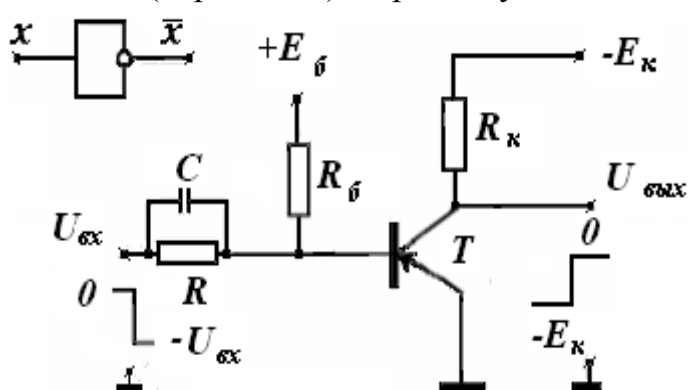


Рис. 8. Схема включения транзистора

оно будет по абсолютной величине уменьшаться. После насыщения транзистора $U_{вых}$ будет близко к 0. Если считать низкий уровень напряжения за двоичную единицу, а отсутствие напряжения – за двоичный нуль, то эта схема, реализует логическую операцию отрицания «NOT» (инверсию).

В компьютерах к электронным логическим схемам предъявляется требование, чтобы данная схема могла управлять большим количеством других схем, а для этого необходимо, чтобы схема обеспечивала большой коэффициент усиления по току. Поэтому практически всегда используется схема включения транзистора с общим эмиттером (заземленным).

Рассмотрим две простейшие схемы логического умножения «AND» (конъюнкции), реализованные на диодах и транзисторах, рис. 9. Если на входах диодов A, B, C левой схемы будут высокие напряжения, равные напряжению питания $+E$ (1) или больше, то диоды будут закрыты и на выходе схемы напряжение U будет равно $+E$ (1). Если же хотя бы на одном из входов напряжение будет отсутствовать, то через сопротивление R и открытый диод будет протекать ток, создавая на R падение напряжения почти равно $+E$ (1). Если же хотя бы на одном из входов напряжение будет отсутствовать, то через сопротивление R и открытый диод будет протекать ток, создавая на R падение напряжения почти равное $+E$. На выходе напряжение U будет близко к 0.

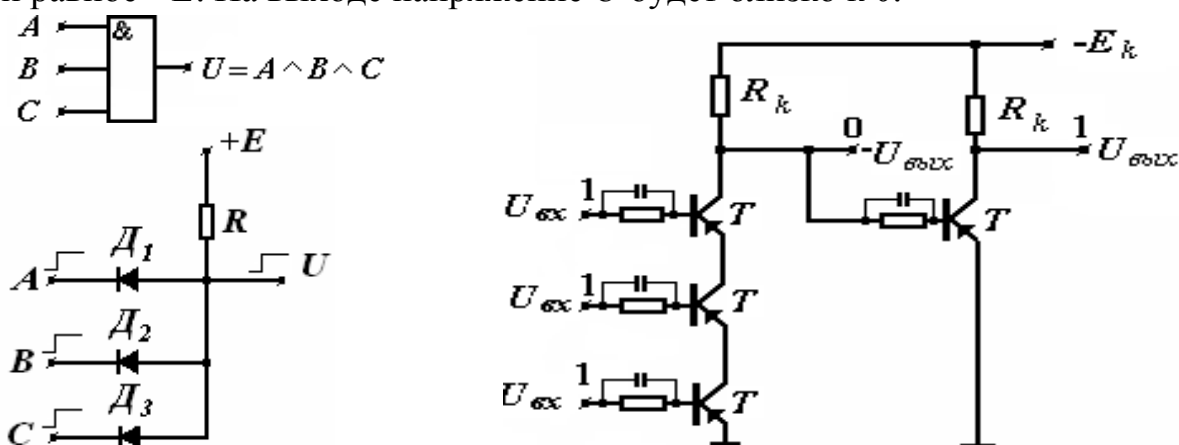


Рис. 9. Логическая схема «И» на диодах и транзисторах

Следовательно, эта схема реализует операцию логического умножения. Рядом справа приведена такая же последовательная схема «И» на транзисторах. На практике используются более сложные схемы логического умножения, как на диодах, так и на транзисторах.

Аналогично логическому умножению выполняются схемы логического сложения «OR» (дизъюнкции). На рис. 10 приведены простейшие схемы, реализованные на диодах и транзисторах. Если хотя бы на одном из входов схемы будет подан импульс тока, он будет и на выходе. Аналогично и на транзисторной схеме.

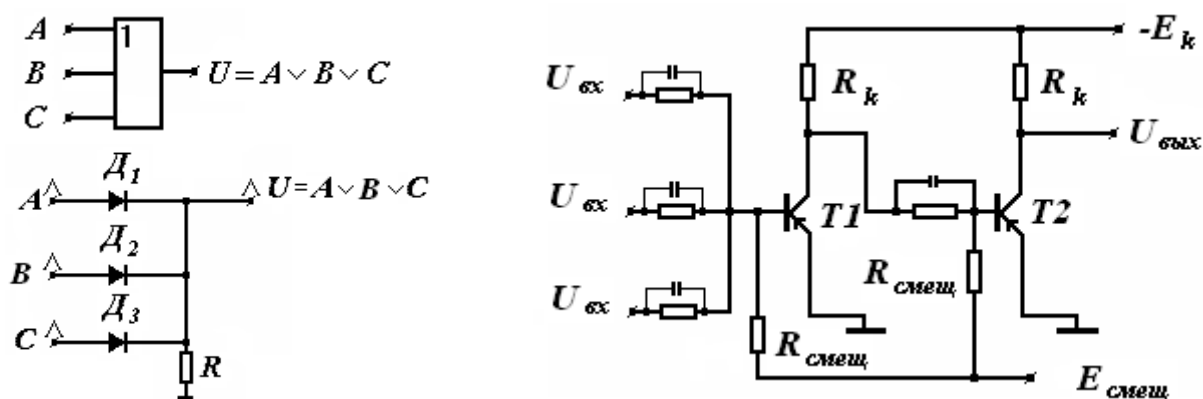


Рис. 10. Логическая схема «ИЛИ» на диодах и транзисторах

Очень важную роль играет схема основного двоичного переключателя – *триггера*. Схема на рис. 11 содержит две схемы логического отрицания, объединенные через положительные обратные связи. Схема, обладая двумя устойчивыми состояниями, способна при определенных внешних воздействиях скачком переходить из одного состояния в другое. Основная часть узлов современных цифровых вычислительных и логических устройств выполняется на триггерах. На них строятся регистры, счетчики, дешифраторы, сумматоры и т. д. При включении питания один из транзисторов откроется быстрее, чем другой и за счет петли обратной связи будет держать другой транзистор в закрытом состоянии. Для установки триггера в определенное состояние имеются два установочных входа: для установки в единичное состояние и для установки в нулевое состояние. Таким образом, триггер может хранить только 1 или 0.

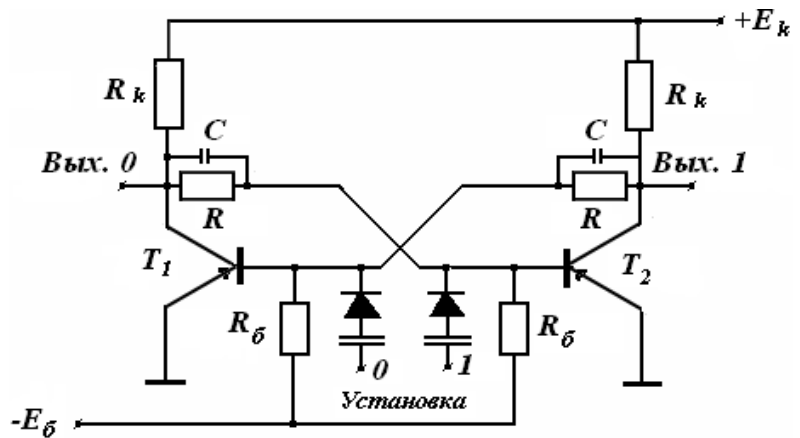
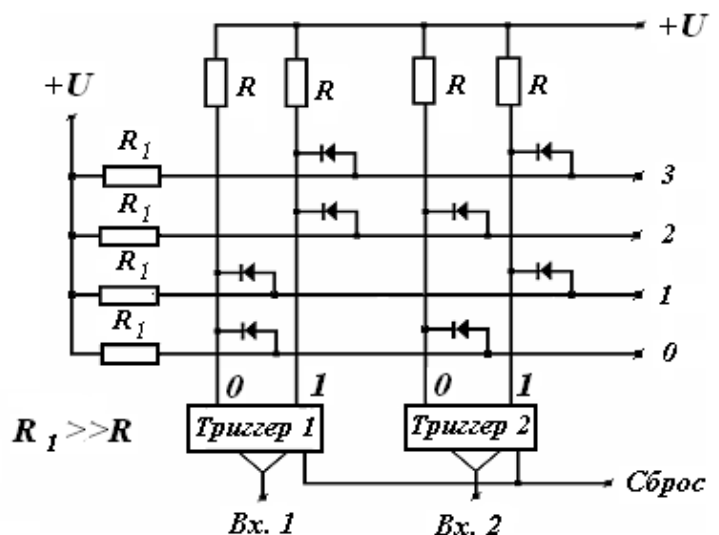
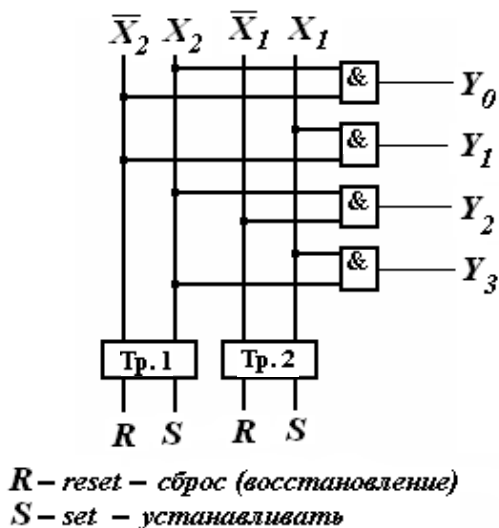


Рис. 11. Схема простейшего триггера

В настоящее время в интегральной схемотехнике применяются десятки разновидностей схем триггерных устройств, отличающихся способом записи информации, схемотехнической реализацией и т. д. По способу записи информации триггеры подразделяются на *асинхронные* и *тактируемые*. В асинхронных триггерах запись информации осуществляется непосредственно с поступлением информационного сигнала на его вход. Запись информации в тактируемые триггеры, имеющие информационные и тактовые (разрешающие) входы, осуществляется только при одновременной подаче тактирующего импульса.

Для выработки управляющих сигналов в зависимости от поступивших двоичных кодов, и для направления их к нужным схемам используются логические схемы – *дешифраторы*, рис. 12.

Дешифраторами называют комбинационные логические структуры, пре-



образующие код двоичного числа, поступающий на входы, в управляющий сигнал только на одном из выходов. *Подчеркнем, что дешифратор не расшифровывает в обычном понимании код числа, поступивший на его вход, а именно*

Рис. 12. Логическая и электрическая схемы дешифраторов

вырабатывает на определенном выходе управляющий сигнал для других логических схем, связанных с этим выходом. При дешифрации M -разрядного двоичного кода и реализации всех комбинаций этого кода число выходов дешифратора будет равно $N=2^M$.

Число входов дешифратора зависит от формы подачи входного двоичного кода: в однофазном виде или в парафазном виде. В зависимости от разрядности дешифрируемого кода и функциональных возможностей логических интегральных схем, находящихся в распоряжении разработчика, дешифратор может быть выполнен на основе одноступенчатой (линейной), как на рис. 12, или многоступенчатой схем дешифрации. На рис. 12 на вход может быть подан только двухразрядный двоичный код, дающий четыре сочетания: 00, 01, 10, 11. Поэтому у данного дешифратора всего четыре выходных шины. Для расшифровки многоразрядных кодов обычно используют многоступенчатые прямоугольные (матричные) и пирамидальные схемы построения дешифраторов.

Лекция 10

Понятие об алгоритмах

Почти во всех сферах жизни нам приходится ежедневно иметь дело с различными правилами, инструкциями, предписаниями, и т. д., в соответствии с которыми можно или нужно что-то сделать. Когда такие инструкции удовлетворяют определенным минимальным требованиям, то говорят *об алгоритме*. Название это введено в честь знаменитого ученого Мухаммеда аль-Хорезми, жившего во «дворце мудрости» багдадского халифа (780–850 гг. н. э.), о котором мы уже упоминали в *Лекции 6*. Латинская форма написания имени великого математика IX в. – algorithmi. *Понятие алгоритма – одно из фундаментальных понятий информатики.* Алгоритмизация наряду с моделированием выступает в качестве *метода информатики.* Алгоритмы являются объектом систематического исследования пограничной между математикой и информатикой научной дисциплины, примыкающей к математической логике – *теории алгоритмов.* Теория алгоритмов является основополагающей теорией для наук, связанных с вычислительной техникой – *основным инструментарием информатики.*

Таким образом, все мы живем в мире алгоритмов, не замечая этого. Алгоритмы экономят силы и время человека, так как однажды усвоенным алгоритмом можно пользоваться всю жизнь. Одни алгоритмы органично «сидят» в нас – заложены природой, воспитанием, обучением, тренировкой, выработаны на основе собственного опыта. К другим приходится обращаться, когда мы хотим выполнить что-то сложное. Например, собирая сложную электронную игрушку, мы

вынуждены пользоваться не только принципиальной электрической схемой, но и монтажной схемой и инструкцией по сборке.

Исходя из сказанного можно дать такое определение алгоритма.

Алгоритм есть точное, сформулированное на определенном языке конечное описание, определяющее содержание и последовательность исполнимых элементарных операций обработки, переводящих исходные данные в искомый результат.

Описание должно быть составлено настолько точно, чтобы исключить его неоднозначное понимание. Такому пониманию не должны мешать, вообще говоря, ни орфографические ошибки, ни безобидные опечатки. Один и тот же алгоритм может быть записан на любом языке, в том числе и на языке пиктограмм. При этом неважно, описаны ли единичные операции обработки словесно, в виде формул или графически.

Многие инструкции и правила, составленные для исполнителя-человека, не соответствуют данному требованию, так как предполагают наличие у него некоторого дополнительного объема знаний, умений, интуиции и здравого смысла.

Создание и использование в качестве исполнителей различных автоматов, роботов и компьютеров требуют *точного и однозначного* описания алгоритмов их работы. Это связано с тем, что каждое из этих устройств может выполнять ограниченный, строго определенный набор законченных простейших действий (и только их). *Указание выполнить конкретное законченное простейшее действие называется командой.* Совокупность всех команд, которые может выполнять исполнитель, называется *системой команд* данного исполнителя. Ограниченные возможности исполнителей – одна из причин трудностей, возникающих при создании алгоритмов. Поэтому, например, чем разнообразнее в компьютере система команд, тем более широкий класс задач можно решать с его помощью.

Поскольку алгоритмы могут применяться к самым произвольным объектам (числа, буквы, слова, логические выражения, станки с числовым программным управлением и т. д.), то в определении алгоритма используется термин, показывающий его назначение. В вычислительных алгоритмах объектами являются числа, в геометрических задачах – точки, линии, плоскости, многомерные фигуры, а в алгоритмах работы станков с числовым программным управлением объектами являются режущий инструмент, его пространственное положение, скорость вращения, перемещения и резания и другие параметры.

Выполняя алгоритм, исполнитель может не вникать в смысл того, что он делает и вместе с тем получить нужный результат. В таком случае говорят, что исполнитель действует *формально*, отвлекается от смысла (содержания) поставленной задачи и только строго выполняет соответствующие команды алгоритма. Для примера рассмотрим простейший алгоритм:

1. Пусть $X=34$, $Y=85$.
2. Если $X=Y$, то перейти к выполнению шага 8.
3. Если $X > Y$, то перейти к выполнению шага 6.
4. Значение Y уменьшить на величину X .
5. Перейти к шагу 2.
6. Значение X уменьшить на величину Y .
7. Перейти к шагу 2.

8. Величине D присвоить значение, равное значению X .

Не зная, что представляет собой эта задача, мы получим $D = 17$ – правильный ответ, если строго выполним все пункты последовательно друг за другом. А между тем, это алгоритм решения задачи по определению наибольшего общего делителя двух натуральных чисел X и Y – знаменитый *алгоритм Евклида*.

Поскольку алгоритм как система простейших законченных операций носит формальный характер, то на его основе всегда можно разработать программу действий для программно-управляемого электронного автомата, в частности компьютера, и осуществить машинное решение задачи. Таким образом, *решение задач на основе алгоритмов может быть автоматизировано*.

В связи с этим особое значение приобрели работы по описанию процессов управления автоматизированными объектами в виде алгоритмов. Получаемые при описании алгоритмы не всегда пригодны для непосредственного использования. Поэтому большое значение имеет проблема равносильных преобразований алгоритмов с целью их «улучшения». Важными проблемами являются оценка качества полученного алгоритма, т. е. эквивалентность алгоритма исходному процессу; его экономичность (алгоритм должен позволять программе разместиться в памяти); эффективность, например, по быстродействию (никому не нужен прогноз погоды на позавчера) и нахождение в том или ином смысле оптимальных алгоритмов.

Свойства алгоритмов. Алгоритм должен быть составлен таким образом, чтобы исполнитель, в расчете на которого он создан, мог однозначно и точно следовать его командам и эффективно получать нужный результат. Алгоритмам присущ ряд свойств, которые и объясняют, почему исполнитель может получить решение какой-то задачи без особого труда. Если какое-то свойство не выполняется, то такое описание последовательности действий над заданными объектами будет называться не алгоритмом, а *предписанием*.

1. *Дискретность* – одно из первых требований, которое предъявляется к алгоритму. Суть его заключается в том, что описываемый им процесс должен состоять из последовательности законченных действий-шагов. Возникающая в результате такого разбиения запись представляет собой упорядоченную совокупность четко разделенных друг от друга предписаний (команд, операторов, директив), образующих дискретную структуру алгоритма. Переход к следующему шагу возможен лишь после завершения предыдущего. Причем на каждом шаге выполняется простейшая операция, соответствующая одному из определенных простейших актов, на которые расчленяется определяемый алгоритмом процесс. Свойство алгоритма состоять из отдельных шагов и называется дискретностью.

2. *Массовость* – возможность применять алгоритм к различным вариантам исходных данных (например, алгоритм Евклида), т. е. возможность решать любую задачу некоторого класса, описываемого данным алгоритмом. Это свойство значительно увеличивает практическую ценность алгоритма.

3. *Понятность*. Используемые на практике алгоритмы составляются с ориентацией на определенного исполнителя. Чтобы составить для него алгоритм, нужно знать, какие команды этот исполнитель может понять и выполнить, а какие

– нет. У каждого исполнителя имеется своя система команд, даже у отдельно взятых компьютеров системы команд различаются (например, компьютер с процессором *Pentium* без блока команд *MMX* не может выполнять приложения мультимедиа). Очевидно, составляя алгоритм для определенного исполнителя, например для станка с программным управлением, можно использовать только те команды, которые имеются в системе команд этого исполнителя. Это свойство алгоритмов и будем называть *понятностью*.

4. *Результативность*. Под результативностью алгоритма будем понимать существование таких вариантов исходных данных, для которых после конечного числа элементарных операций выполнение алгоритма заканчивается и дает искомый результат, и отсутствие таких вариантов, для которых процесс выполнения алгоритма приводит к неправильному результату. При этом допускается существование вариантов исходных данных, для которых не получается никакого результата, так как процесс выполнения алгоритма обрывается в силу того, что к полученному промежуточному результату дальнейшее применение алгоритма не имеет смысла, либо потому, что этот процесс никогда не заканчивается (зацикливается).

В практике автора был случай, когда после решения задачи в течение четырех лет на очередной порции исходных данных задача зависла. После неоднократной проверки задачи на исходных данных за предыдущие отчетные периоды удалось обнаружить ошибку в алгоритме, которая проявилась только на определенном наборе исходных данных.

Широко известен также факт, свершившийся в конце 1994 г. в связи с выпуском фирмой *Intel* первых процессоров *Pentium*. В более чем 5 млн проданных процессоров была обнаружена ошибка алгоритма выполнения операции деления с плавающей запятой при определенных сочетаниях двух чисел, что незамедлительно сказалось на престиже и финансовом положении фирмы.

При многочисленных тестовых проверках процессора в течение нескольких месяцев летом 1994 г. специалисты фирмы пришли к выводу, что с ошибкой выполняется в среднем одна из 9 млрд. операций деления чисел с плавающей запятой. При самой интенсивной загрузке компьютера вычислительными операциями такая ошибка должна была происходить один раз в 27 лет. Поскольку это время значительно превосходило все мыслимые сроки эксплуатации процессора, то на фирме решили не придавать значения этому факту. Но тут-то и случилась беда. Доктор Томас Нейсли из колледжа небольшого городка Линчбург в штате Вирджиния занимался теорией чисел и проводил много вычислений с точностью до 19 знака после запятой. 13 июня 1994 г. он обнаружил несоответствие между имеющимися таблицами и результатами вычислений. Он долго перепроверял свои программы и, наконец, 19 октября обратился в фирму *Intel*, но в течение недели ответа не получил. 30 октября по электронной почте он разослал серию посланий своим коллегам и всем, у кого есть компьютеры с такими процессорами, с просьбой проверить его результаты. За считанные дни новость распространилась по всему миру. Множество людей бросились перепроверять свои расчеты и также обнаружили ошибки. Эта новость появилась на первых полосах газет и в заголовках телевизионных новостей, акции фирмы стремительно начали падать в цене. Фирма же вместо того, чтобы покаяться и извиниться, сделала вид, что

ничего и не произошло, заявив, что об ошибке она знает, но вероятность появления ее настолько мала, что процессоры менять не надо. Людей же раздражало главное – что фирма знала об ошибке раньше, чем ее обнаружили, и молчала. Фирме пришлось все-таки принести свои извинения и объявить, что она готова заменить процессоры на новые с условием доставки старых.

Все сделали определенные выводы:

- фирма *Intel* поняла, что врать покупателям нельзя;
- свои возможности показала сеть Интернет;
- были обнаружены ошибки в действующих и широко распространенных программах, в частности, в *MS Excel 5.0* – при выполнении одной из функций.

Приведем несколько пар чисел, при делении которых появлялась ошибка: **5505001 : 294911** правильный ответ – 18,66665197, а компьютер давал результат – 18,66600093; **4195835 : 3145727** правильный ответ – 1,33382044, результат компьютера – 1,333739068.

5. Определенность. Каждый алгоритм строится в расчете на некоего исполнителя. Чтобы исполнитель мог решать задачу по заданному алгоритму, необходимо, чтобы каждая команда алгоритма всегда однозначно понималась и точно исполнялась. Запись алгоритма должна быть настолько четкой, полной и продуманной в деталях, чтобы у исполнителя не могло возникнуть потребности в принятии решений, не предусмотренных алгоритмом. Это значит, что любая команда, выполняемая много раз различными исполнителями, имеющими эту команду в своей системе команд, при одних и тех же исходных условиях всегда должна давать один и тот же результат. После выполнения каждой команды исполнитель должен знать, какую команду нужно выполнять следующей. Чтобы исключить возможность неоднозначного понимания команд алгоритма, для описания алгоритмов разработаны специальные формализованные алгоритмические языки, в которых каждое предложение имеет абсолютно точный смысл.

Способы записи и разработка алгоритмов. Для записи алгоритмов существуют различные способы. Одни из них ориентированы на исполнителя – человека, другие – на программно-управляемые электронные устройства, третьи являются вспомогательными для облегчения рассуждений. Алгоритм можно представить с помощью словесного или графического описания, в виде последовательности формул или в табличном виде.

Словесный способ представляет собой текстовую запись последовательного выполнения определенных действий, приводящих к конечному результату. Форма записи команд произвольная, главное, чтобы она не допускала неоднозначных толкований и была понятной. Покажем это на примере нахождения наименьшей из двух величин.

1. Задать числовые значения величин A и B .
2. Если $A < B$, то положить Y равным A , иначе положить Y равным B .
3. Записать в ответе значение Y .
4. Конец.

Словесный способ чаще всего используется на начальных этапах изучения поставленной задачи. При этом задача разбивается на отдельные самостоя-

тельные крупные блоки, имеющие функциональную законченность. Затем эти блоки детализируются и разбиваются на более мелкие, которые потом объединяются в один алгоритм. Этот способ записи в основном рассчитан на исполнение алгоритма человеком.

Графический способ. При этой форме записи алгоритмов отдельные команды изображаются в виде плоских геометрических фигур (см. справа), внутри которых описываются словесно или с помощью формул выполняемые действия. Используемые графические фигуры блоков стандартизированы в соответствии с *ГОСТ 19.003-80. Схемы алгоритмов и программ. Обозначения условные графические.* Сейчас он заменен на *ГОСТ 19.701-90 (ИСО 5807-85).*

Прямоугольник часто называют элементом общей обработки, ромб – элементом принятия решений и используют для организации ветвления или цикла в алгоритме. Последовательность выполнения блоков, т. е. переход от одного блока к другому, обозначается стрелками. Этот способ имеет ряд преимуществ благодаря наглядности. Он имеет четкую и ясную структуру, легко поддается проверке. Такие алгоритмы называют **блок-схемами**.



Блок-схема – это ориентированный граф, вершины которого могут быть одного из типов, представленных на рис. 13.

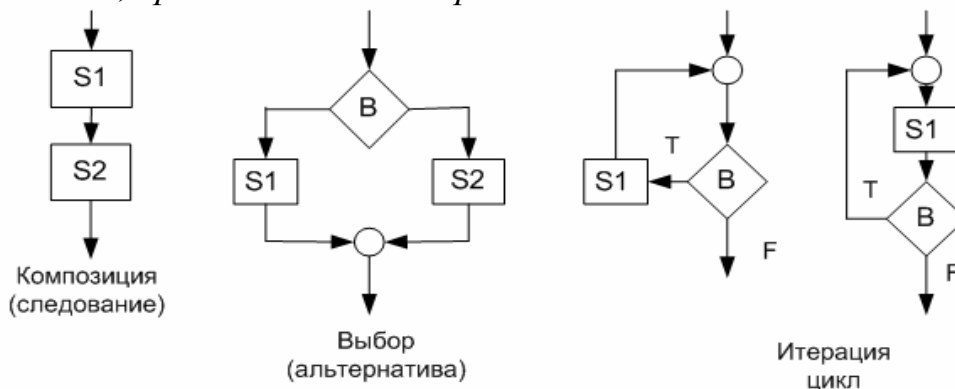
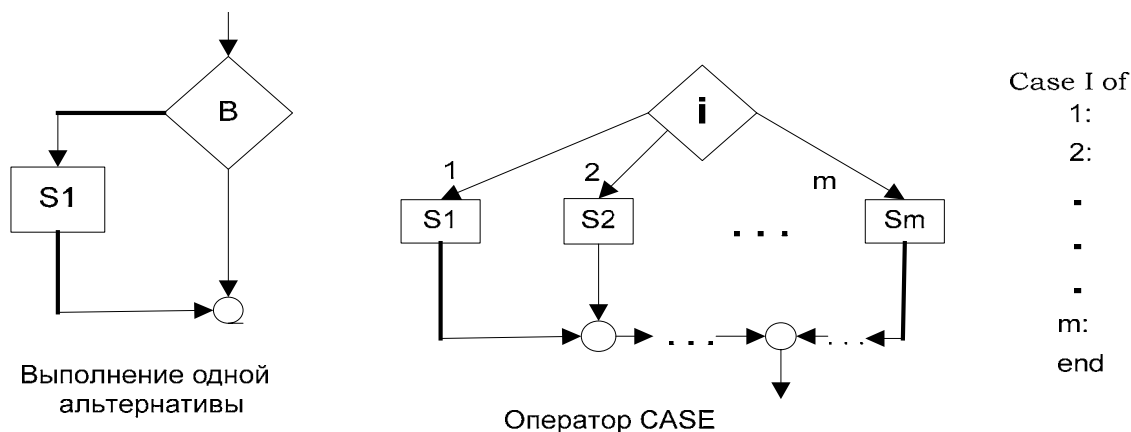


Рис. 13. Примеры блок-схем алгоритмов

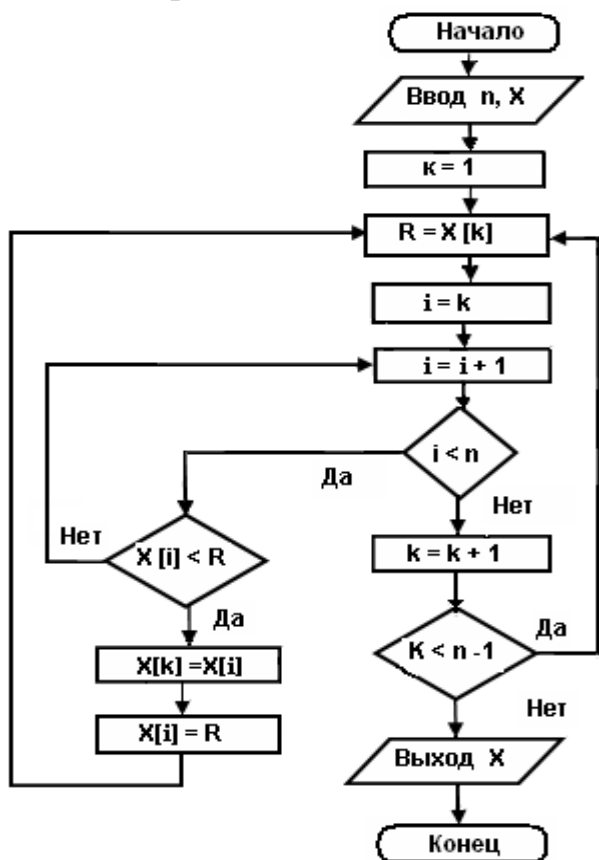
Из этих четырех элементарных блок-схем может быть получена любая *структурная блок-схема*. Важной особенностью приведенных структур является то, что они имеют один вход и один выход. Бывает, что в процессе поиска алгоритм не может быть представлен в структурном виде. Тогда приходится применять специальные приемы, преобразования неструктурного алгоритма в

структурный. Это и дублирование блоков, введение вспомогательных постоянных (метод Ашкородта-Манна), введение логических переменных – признаков (метод логического признака), объединение условий продолжения цикла и т. д.

В более широком плане структурное программирование допускает большее разнообразие элементарных структур управления, чем предложенные четыре. На рисунке ниже представлены еще две элементарные структуры: выполнение одной альтернативы и оператор множественного выбора. Причиной для расширения множества структур является требование удобства и естественности.



Приведем пример составления алгоритма упорядочения по возрастанию значений элементов массива $X[1:n]$. Алгоритм упорядочения (сортировки) элементов довольно часто используется в информационно-справочных системах. Нужное слово в словаре легко найти только потому, что все слова там упорядочены по алфавиту (лексикографически). Так же упорядочена информация в библиотечном каталоге, телефонном справочнике, энциклопедии, в регистратуре поликлиники и т.д. Для решения задачи сортировки придумано множество различных алгоритмов.



Рассмотрим один из них, идея которого заключается в следующем. Сначала запоминают первый элемент массива и сравнивают его последовательно с остальными, пока не попадет элемент, меньший первого. Тогда их меняют местами, и так до тех пор, пока на первое место не попадет наименьший элемент массива. Затем запоминают второй элемент и проделывают с ним то же самое, заменяя его следующим наименьшим элементом. Прделав этот процесс $n-1$ раз, располагают элементы в массиве по возрастанию.

Алгоритмический язык. В алгоритмическом языке используется гра-

ниченное число слов, смысл и способы употребления которых строго определены. Эти слова называются *служебными*. Исходные данные задачи называются *аргументами* и перечисляются после записанного сокращенно служебного слова *арг.* Для выявления структуры алгоритмов обычно при их записи каждую команду пишут в отдельной строке, делают отступы от начала строки, выделяя отдельные блоки или процедуры алгоритма, чтобы была наглядность и легкость его понимания.

Рис. 14. Пример алгоритма сортировки ваются *аргументами* и перечисляются после записанного сокращенно служебного слова *арг.* Для выявления структуры алгоритмов обычно при их записи каждую команду пишут в отдельной строке, делают отступы от начала строки, выделяя отдельные блоки или процедуры алгоритма, чтобы была наглядность и легкость его понимания.

Запишем на русском языке простой алгоритм нахождения наименьшего числа из двух. В алгоритме курсивом выделены служебные слова. Обычно при разработке алгоритмического языка стремятся к тому, чтобы он удовлетворял следующим требованиям:

Алг минимальное из двух

Арг А, В

Рез Y

Нач

Если A<B

То Y:=A

Иначе Y:=B

Все

Кон

- легкость изучения, понимания и применения;
- компактность описания процессов;
- удобство использования в публикациях;
- универсальность.

Некоторые из этих требований противоречат одному другому, например второе и четвертое, так как нельзя сделать компактным универсальное. Однако при создании языков находят компромиссы, удовлетворяющие всем требованиям.

Язык программирования высокого уровня служит для записи алгоритмов в таком виде, в котором они могут быть исполнены компьютером, т. е. по существу в виде компьютерных программ. Компьютерная программа - это алгоритм, доведенный до машинного исполнения. Для примера запишем на языке программирования TurboPascal 7.0 текст алгоритма ввода 20 чисел в массив и упорядочивания их по возрастанию:

```
PROGRAM СОРТИРОВКА(INPUT,OUTPUT);
VAR I,J:INTEGER;
    R:REAL;
    X:ARRAY [1..20] OF REAL;
BEGIN
  FOR I:=1 TO 20 DO READ(X[I]);
  FOR I:=1 TO 19 DO
    FOR J:=I+1 TO 20 DO
      IF X[I]>=X[J] THEN
        BEGIN
          R:=X[I]; X[I]:=X[J]; X[J]:=R;
        END;
  WRITELN ('СОРТИРОВКА ПРОИЗВЕДЕНА');
  FOR I:=1 TO 20 WRITE(X[I])
END
```

Разработка алгоритмов решения задач на ЭВМ требует специальных навыков и знания определенных приемов их записи. Существует весьма большое количество всевозможных приемов и методов разработки алгоритмов. Основным приемом является структурный подход, т. е. разбивка всего алгоритма

на отдельные структуры, которые затем объединяются в единый алгоритм. Структуры могут быть самые различные: циклы, ветвления, процедуры, подпрограммы и т. д. Особо следует обращать внимание на организацию циклов. Практически не бывает алгоритмов, в которых не использовались бы циклические структуры. Они являются основными частями всяких практически используемых алгоритмов. Количество повторений цикла может быть известно заранее (*арифметические* циклы) или заранее неизвестно (*итерационные* циклы). Для тех и других циклов существует множество способов их организации. Нередко взаимосвязь между циклами в алгоритме бывает довольно сложной: циклы переплетаются друг с другом или входят один в состав другого. Организация цикла – наиболее сложная задача, так как от оптимального построения циклов зависит время выполнения алгоритма или быстрота решения задачи.

В практике автора был случай, когда для решения задачи оптимального планирования была приобретена программа для ЭВМ ЕС, которая при матрице примерно 1000x900 выполнялась более 22 часов непрерывно. Дело оказалось в том, что обращение к дискам (медленным устройствам) было организовано в сложном цикле.

Приведем несколько методов построения алгоритма.

Метод частных целей лежит в основе решения многих задач и составляет основу алгоритмизации. Суть его сводится к вопросу: «Можно ли данную задачу разбить на последовательность более простых». Конечно, в конкретной сложной задаче очень часто трудно указать способ ее разбиения на набор более простых задач. Здесь большое значение имеют опыт и искусство специалиста.

Метод подъема. Этот метод, как и предыдущий, можно отнести к одному из общих «рецептов» разработки алгоритмов. Алгоритм начинается с принятия начального предположения или построения начального решения задачи. Затем идет уточнение решения, алгоритм дорабатывается более подробно и так продолжается движение «вверх» от начального уровня по направлению к получению наилучшего решения.

Программирование с отходом назад. Данный метод применяется для разработки программ искусственного интеллекта, независимо от того, к чему он прилагается – программирование игр, распознавание образов, выбор решений и т. д. Здесь основным является *перебор вариантов*. Это сложная задача, так как алгоритмы перебора ищут решение не по заданным правилам вычислений, а путем проб и ошибок. Схема их не укладывается в схемы ветвлений и циклов. Ситуация часто осложняется тем, что прямыми методами перебор всех возможных вариантов осуществить нельзя из-за их огромного количества. Метод алгоритмизации с отходом назад позволяет осуществить организованный исчерпывающий поиск требуемого решения задачи. При этом часто удается избежать перебора всех возможных вариантов.

Наиболее сложным разделом алгоритма перебора вариантов является движение назад (откат) и связанная с этим необходимость восстановления номера последнего варианта предыдущего хода. Этот вопрос автоматически решается при *рекурсивной* организации алгоритма.

Имеется еще ряд методов и способов разработки алгоритмов, используемых программистами высокой квалификации, но из-за их сложности эти методы здесь не рассматриваются.

Лекция 11

Понятие о программировании

Чтобы заставить компьютер выполнять какой-либо алгоритм, нужно ввести в его память инструкции по выполнению этого алгоритма. Такими инструкциями является компьютерная программа – набор машинных команд, записанных в виде двоичных кодов на понятном компьютеру языке. Запись человеком алгоритма на машинном языке в кодах обладает рядом существенных недостатков. Одним из них является то, что логика выполнения алгоритма у машины иная, чем у человека. В соответствии с этим человеку надо приспособить свой алгоритм к машинной логике. Поэтому широкое применение компьютеров для обработки информации в самых различных сферах деятельности людей вызвало появление множества искусственных *машиннонезависимых языков программирования высокого уровня* для создания компьютерных программ. Алгоритмические языки позволяют в большей степени отражать логику человека в программах, реализующих алгоритм. Средства записи алгоритмов на этих языках близки к обычному математическому описанию и более естественны для человека. При построении конструкций алгоритмических языков используются математические выражения, запись которых мало отличается от общепринятой. Все это делает средства написания компьютерных программ на алгоритмических языках более привычными и понятными для человека. Кроме того, преимуществом таких языков по сравнению с машинными кодами является их универсальность: программа, написанная на таком языке, может выполняться на компьютерах различного типа. Привязка программы к конкретному типу машины выполняется при трансляции очень просто и не требует практически каких-либо переделок или требует, но очень незначительных.

*Разработка программ с помощью языков программирования высокого уровня на основе рассмотренных в предыдущей лекции алгоритмов из структурных блок-схем составляет процесс **структурного (процедурного) программирования**.*

К языкам программирования на разных стадиях жизненного цикла программного изделия предъявляются различные, порой противоречивые требования. Например, при эксплуатации программы сложность языка для программирования не столь существенна, как ее ресурсоемкость. На стадии проектирования важно удобство написания программы, а на стадии сопровождения программы – удобство и легкость ее чтения.

Если очевидно, что разнообразие машинных языков обусловлено большим количеством типов машин, то зачем же такое разнообразие языков программирования высокого уровня? Их ведь надо учить и знать особенности и тонкости каждого! Ответ прост. Это обусловлено потребностями людей решать с помо-

стью компьютера широкий спектр проблем. Развитие всех сфер жизни людей ежедневно ставит перед человечеством все новые и новые проблемы, требующие своего решения. Одни компьютеры и языки программирования предназначены для решения узкого круга специфических и однотипных задач, например обработка аэрокосмической информации или моделирование процессов ядерной физики. Другие позволяют охватить очень широкий круг разнообразных задач. Здесь уместна аналогия с автомобилями: одним нужен комфорт, другим – скорость, третьим – малый расход топлива.

Основой каждого языка является определенный набор символов – *алфавит* языка: буквы, цифры, знаки математических операций $+$, $-$, $/$, $*$, $^$, $<$, $>$, $=$, синтаксические знаки $($, $;$, $:$, $[$, $]$, $\{$, $\}$, специальные знаки $\#$, $\$$, $\&$, $\%$ и др. Используются также *ключевые слова* – последовательность символов, имеющая особый, раз и навсегда установленный смысл в данном языке, например: *FOR, INPUT, READ, RETURN, THEN, SUB, CALL, BEGIN*. Имена, или идентификаторы присваиваются *константам, переменным, функциям*, целым наборам переменных – *массивам* и т. д. Правила построения из символов алфавита различных языковых конструкций представляют собой *синтаксис* языка. *Семантика* языка определяет смысл каждой языковой конструкции, ее связь с элементами той предметной области, для описания и решения задач которой создан язык.

Для большинства языков основными синтаксическими элементами являются *операторы* – зарезервированные символы (или слова), которые выполняют действия над переменными и константами для получения результата. Обычные математические знаки (сложить $+$, умножить \cdot (\times), разделить $:$ ($/$), вычесть) являются *арифметическими операторами*. Знаки $>$, $>=$, $<$, $<=$, $<>$, $=$ являются *операторами сравнения* – сравнивают значения двух операндов и возвращают значения *True* (истина) или *False* (ложь) в зависимости от сравниваемых значений. Переменные или константы, объединяемые операторами, называются *операндами*. В примере: $2 \times 3 - 4 = 2$ числа 2, 3 и 4 являются операндами, а знаки \times , $-$, $=$ являются операторами. Оператор может выполнять действия над одной переменной, тогда он имеет *унарную* форму (оператор с одним операндом), например $Var = - Var$. В данном случае унарный минус меняет знак переменной *Var* с «плюс» на «минус» или с «минус» на «плюс», т. е. инвертирует знак переменной. Операторы могут быть и *логическими* *And* (логическое умножение), *Or* (логическое сложение) или *Not* (отрицание). Рассмотрим пример: если в выражении $Y = X \text{ AND } 32$ пятый разряд числа *X* равен 1, то $Y = \text{True}$, иначе $Y = \text{False}$.

Язык должен включать определенный набор средств и правил выполнения различных операций над данными. Последовательности операций вычисления некоторых значений переменных представляют собой *выражения*, например: $X = S + 2.7 \times \text{SIN}(I) - 127.64$. Выражения строятся из констант и переменных с помощью различных операторов и обозначений стандартных функций.

Подчеркнем, что в практике программирования операторами называют инструкции (команды) программного кода, а не синтаксические элементы языка программирования. Правда, в последние годы от этого быстро отходят и все чаще называют команды программного кода инструкциями. Каждый язык про-

граммирования включает в себя определенный набор различных инструкций.

Последовательность команд (инструкций) представляет собой текст программы на данном языке.

Тип значения, полученного при вычислении выражения, зависит от типов входящих в это выражение величин и характера операций. Любая программа оперирует объектами, которые для компьютера принято называть *данными*. Каждый язык программирования имеет свой набор данных. С каждой величиной в определенный момент связывается некоторое *значение*. Если значение величины фиксировано на весь период решения задачи, то эта величина называется *константой* (например, $R=25,8$; $S=-172,63$), в противном случае – *переменной*. Значение переменной может быть изменено во время выполнения программы простым присвоением нового значения. Например, текущая переменная I при повторении цикла меняет свое значение, увеличиваясь на единицу $I = I + 1$. Человек, оперируя какими-либо данными, способен быстро разобраться в них и в тех операциях, которые допустимы для определенных данных. Быстро разобраться в этом ему позволяет отличие в записи разных *типов* данных. Компьютер же любую информацию воспринимает как последовательность двоичных чисел. Поэтому ему *необходимо указать*, какой тип имеет каждое используемое в программе данное. Язык должен иметь средства описания типов величин. В большинстве языков программирования это осуществляется с помощью операторов-описаний. Типы всех величин, как правило, предварительно описываются (объявляются) в разделе переменных в начале программы, чтобы в дальнейшем программа знала, с каким типом выполняется та или иная операция. Некоторые операции сами определяют тип данных.

Например, в макроязыке программирования *Visual Basic for Application (VBA)*, встроенном в программный пакет *Microsoft Office*, ключевое слово *Dim* служит для объявления переменной и позволяет указать тип данной переменной: *Dim m As Integer*. Здесь объявлена переменная m , которой присвоен тип *Целое*. В языке программирования *TURBO PASCAL 7.0* для объявления переменных служит ключевое слово *VAR*, например *VAR I:INTEGER; VAR V:REAL* и т. д.

Перечислим некоторые существующие типы данных.

Тип целый (INTEGER) задает некоторое подмножество целых чисел. Размеры подмножества зависят от особенностей реализации языка. Со значениями целого типа обычно выполняются операции сложения, вычитания и умножения. Допускаются операции деления нацело и определения остатка от деления. Целые величины чаще всего используются в качестве *индексов и параметров циклов*.

Тип вещественный (REAL) задает некоторое подмножество вещественных (действительных) чисел. Конкретная реализация языка определяет диапазон и точность используемых действительных значений. Например, в языке программирования *TURBO PASCAL 7.0* вещественные числа содержат до 20 десятичных значащих цифр в зависимости от типа. Для записи вещественных чисел очень часто используется экспоненциальная форма, например $0,478E+05$, где символ E читается как «умножить на 10 в степени» (см. лекцию 8). С вещественными числами могут выполняться все арифметические операции. Общее для

всех языков правило: если хотя бы один из операндов в операции имеет вещественный тип, то весь результат вещественный.

Тип булевский (BOOLEAN) задает множество из двух констант, называемых *TRUE* (истина) и *FALSE* (ложь). С булевскими (логическими) значениями переменных выполняются логические операции отрицания (*NOT*), конъюнкции или логического умножения (*AND*), дизъюнкции или логического сложения (*OR*), сложения по модулю два и другие, рассмотренные в лекции 9.

Тип символьный (CHARACTER) задает множество наборов (слов, строк) произвольных символов алфавита языка программирования. Длина переменной символьного типа зависит от конкретной реализации языка и обычно составляет 255 символов. Со значениями данного типа чаще всего выполняются операции выделения текста из строки символов и сцепления цепочек символов. Например, с помощью операции сцепления *&* объединяются в единое целое два слова *Москва & Киев*.

Приведем несколько рекомендаций по вводу текста программы.

а Инструкции вводятся по одной в каждой строке. Используется клавиша *Enter* для перехода к следующей строке. Хотя в некоторых языках программирования в одной строке можно ввести несколько инструкций через точку с запятой (;), делать это не рекомендуется. Такие фрагменты программы, как циклы, процедуры, подпрограммы и т. д. выделяются отступами строки. *Чтение программы, написанной сплошным текстом, затруднительно.*

а Если строка кода очень длинная из-за многочисленности аргументов функций и методов или из-за большой длины аргументов, то можно продолжить инструкцию на следующей строке. Нужно ввести специальный знак (например, в VBA – знак подчеркивания *_*) в конце строки и перейти на следующую строку.

а Между выражениями следует оставлять пустые строки, чтобы улучшить читаемость кода. Это позволит разбить длинную программу на разделы связанных инструкций.

а Следует использовать комментарии, поясняющие сложные для понимания участки программы. Если программа будет редактироваться из-за изменившихся условий после длительного промежутка ее использования, то комментарии помогут понять, как она работает. Комментарием в строке считается все, что находится справа от специального знака (например, в VBA – знака апостроф *'*).

Кратко рассмотрим наиболее часто используемые в программах команды.

Оператор присваивания присваивает переменной значение, полученное путем вычисления выражения или значение другой переменной. В качестве знака присваивания в разных языках программирования используются символы «*=*», «*:=*» или «*←*». Необходимо, чтобы тип значения выражения или переменной, присваиваемых другой переменной, соответствовали объявленному для нее типу. В случае несовпадения типов многие языки программирования автоматически преобразуют тип вычисленного значения к типу переменной, которой присваивается значение. В этом случае может происходить искажение результата. Программист должен отслеживать подобные моменты. *Еще раз подчеркнем, что присваивание – это не равенство, а замена одного значения пере-*

менной другим значением: $x = 2 \times x + 5$. Здесь переменной x присваивается новое значение, равное сумме удвоенного текущего ее значения и числа пять.

Организация цикла. Важную роль в программах играют структуры, многократно повторяющиеся при изменении некоторого параметра, о чем мы упоминали при рассмотрении алгоритмов. Циклы являются основными структурами практически всех используемых программ. Количество повторений цикла может быть заранее известно – *арифметические* циклы, или заранее неизвестно – *итерационные* циклы. Для разных видов циклов в языках программирования существует множество способов их организации. Однако каждый способ имеет общее, заключающееся в том, что в конце цикла проверяется условие его повторения, и в зависимости от этого командой перехода управление передается либо на начало цикла, либо на продолжение программы

Рассмотрим пример записанной на языке программирования *TURBO BASIC* простейшей программы звучания встроенного динамика компьютера при изменении частоты сигнала от 80 Гц до 12 КГц в течение трех секунд на каждой частоте (итерационный цикл). Инструкция *WHILE* в третьей строке является началом цикла, организующим цикл повторения до тех пор, пока переменная F

<i>PLAY "MF"</i>	будет принимать значения меньше 12000. Затем идут
<i>F=80</i>	команды звучания и изменения частоты (тело цикла).
<i>WHILE F<12000</i>	За ними идет команда окончания цикла <i>WEND</i> .
<i>SOUND F,3</i>	
<i>F=F*1,05</i>	
<i>WEND</i>	

Приведем еще пример циклической программы, записанной на языке программирования *PASCAL*, вычисляющий сумму 30 чисел по формуле:

$$S = \sum_{I=1}^{30} (2,7 * \sqrt{I} - 1).$$

Текст программы:

PROGRAM SUMMA;	{-заголовок программы}
VAR S:REAL; I:INTEGER;	{-описание типа переменных S и I}
BEGIN	{-начало}
S:=0;	{-присвоение начальных значений}
FOR I:=1 TO 30 DO	{- FOR - оператор цикла с параметром}
S:=S+(2.7*SQRT(I)-1);	{- вычисление суммы}
Writeln ('Сумма S=', S:5:1)	{-вывод на печать значения суммы S}
END.	{- конец программы}

Здесь цикл организован путем применения оператора цикла с параметром, который используется в тех случаях, когда заранее известно, сколько раз должна повторяться циклическая часть программы (арифметический цикл). Тело цикла выполняется для каждого значения параметра цикла I от его начального значения (1) до конечного значения (30) включительно. Шаг по умолчанию равен 1.

Не менее важны в программах команды ветвления или выбора. Они могут быть безусловными и с условием. Инструкция безусловного перехода во многих языках программирования имеет вид: *GOTO* метка. Для выбора выполняемого блока программы из нескольких возможных по условию наиболее распространена

блочная структура: *If условие Then [операторы] ElseIf условие Then [операторы] ... End If.*

Например, нужно определить рейтинг студента на основе экзаменационных оценок по двум предметам. Запишем код программы в виде процедуры на языке *Visual Basic for Applications (VBA)*:

```
Sub Экзамен()           'имя процедуры
  Dim bal_1 As Integer
  Dim bal_2 As Integer   'раздел объявления типа переменных
  Dim Рейтинг As String
ab:                     'метка перехода по оператору GoTo
bal_1 = InputBox («Введите оценку по физике», «Ввод исходных данных») 'окно ввода
  If (bal_1 < 1) Or (bal_1 > 5) Then 'инструкция ветвления или условия
    MsgBox («Неправильно введена оценка») 'окно вывода сообщения
    GoTo ab 'безусловный переход на метку
  End If
ac:
bal_2 = InputBox («Введите оценку по информатике», «Ввод исходных данных»)
  If (bal_2 < 1) Or (bal_2 > 5) Then
    MsgBox («Неправильно введена оценка»)
    GoTo ac
  End If
  If (bal_1 = 5) And (bal_2 = 5) Then
    Рейтинг = «Отлично»
  ElseIf (bal_1 = 4) And (bal_2 = 4) Then
    Рейтинг = «Хорошо»
  ElseIf (bal_1 = 3) And (bal_2 = 3) Then
    Рейтинг = «Удовлетворительно»
  Else
    Рейтинг = «Неудовлетворительно»
  End If
  ...
End Sub
```

В данной процедуре использованы две команды безусловного перехода на метки *GoTo ab* и *GoTo ac*, две простых блочных структуры и одна более сложная структура ветвления или условия типа *If...Then...End If*, в которых три раза делается выбор сочетания оценок и выдается сообщение. Один раз (*Else*) сообщение выдается при любом сочетании оценок, кроме указанных первых трех.

Языки программирования должны позволять оформление повторяющейся неоднократно в различных местах программы последовательности действий в *процедуре, или подпрограмму*, которой присваивается имя. Это имя используется в качестве сокращенной записи данной последовательности в тех местах программы, где встречается данная последовательность, путем вставки *оператора вызова подпрограммы*. В качестве такого оператора чаще всего используется оператор: *CALL имя подпрограммы* (в *BASIC*).

Описанные принципы программирования относятся к **процедурному программированию**.

Программа, записанная на языке высокого уровня, называется *текстом программы* и представляет собой текстовый файл с расширением, определяемым используемым языком, например, *PRG.PAS* или *PRG.BAS*. Она не может быть выполнена компьютером, так как написана без привязки к какому-либо компьютеру. Машина «понимает» только язык машинных команд. Следовательно, исходный текст программы необходимо *транслировать* (перевести) на язык команд конкретного типа ЭВМ. Такой перевод осуществляется автоматически самим компьютером с помощью специальных программ-трансляторов.

Текст программ на языках программирования высокого уровня для программ-трансляторов является *исходными данными*, которые необходимо обработать, а получаемая программа в машинных кодах – *результатом* работы транслятора. В зависимости от того, как обрабатывается текст программы, трансляторы подразделяют на *компиляторы и интерпретаторы*. Программа-компилятор обрабатывает текст программы целиком весь. Результатом ее работы является *объектный модуль* – файл с расширением *OBJ*, который можно считать «полуфабрикатом» машинной программы. Объектный модуль может не включать в себя всех необходимых для работы машинных команд и процедур, поэтому для получения рабочей машинной программы объектный модуль должен быть обработан еще специальной программой – *редактором связей (компоновщиком)*, которая недостающее в программе объектного модуля вставит из специальных библиотек объектных модулей, имеющихся в каждой системе программирования. Таким образом, получается *загрузочный модуль* программы в виде исполнимого файла с расширением *COM* или *EXE* – машинная программа, готовая к выполнению. После этого программу можно запускать на выполнение.

Второй способ трансляции (интерпретацию) часто называют еще *программным моделированием*. Он заключается в том, что программа - интерпретатор как бы моделирует некий компьютер, для которого исходный текст программы на языке высокого уровня является машинным кодом, т. е. каждый оператор является как бы машинной командой. При этом каждый оператор читается, анализируется в контексте уже работающей программы и выполняется. Интерпретатор заменяет его несколькими машинными командами реального компьютера. Таким образом, выполнение программы производится во время ее трансляции. Компиляцию можно сравнить с письменным переводом текста, а интерпретацию с синхронным переводом текста при его чтении.

Естественно, что компиляция и интерпретация имеют свои преимущества и недостатки, на которых мы останавливаться не будем. *Здесь подчеркнем только, что привязка программы, написанной на языке программирования высокого уровня, к конкретному типу машины осуществляется именно через транслятор.* Поэтому для одного и того же языка программирования высокого уровня имеются несколько трансляторов – для каждого типа машины.

По принципу работы многие трансляторы являются многопроходными. При запуске на первый проход исходного текста программы транслятор, например, выявляет синтаксические и некоторые семантические ошибки, при запуске на второй проход - оптимизирует программу и т. д.

По способу реализации языки программирования подразделяются на компилируемые и интерпретируемые. Некоторые из них могут иметь обе реализации. Например, *PASCAL* компилируется, а *TURBOBASIC* является одновременно интерпретирующим и компилирующим. Это очень удобно, так как небольшие программы для разового использования компилировать не имеет смысла. Их лучше сразу после набора текста запускать на выполнение путем интерпретации, а большие программные пакеты для многоразового использования лучше компилировать и хранить в виде загрузочных модулей.

Откомпилированная программа по сравнению с исходным текстом возрастает в размере во много раз за счет замены каждого оператора множеством машинных команд. Загрузочный модуль по размерам может отличаться даже от объектного в 1,5 – 2 раза за счет включения в него редактором связей дополнительной информации, необходимой для работы программы.

Современные **визуальные средства программирования** приложений для Windows базируются на *объектно-ориентированной модели* программирования, реализованной в таких языках, как *4GL*, *5GL*, *Visual Basic 6.0*, *Borland Delphi 7.0*, *Visual C++*, *Visual J++* и др. **Windows по своей природе – объектно-ориентированная среда.** Эта среда позволяет программисту разрабатывать приложение графически, помещая текстовые поля, раскрывающиеся списки, кнопки, переключатели, полосы прокрутки и т. д. в «чистую» форму приложения (как художник помещает отдельные фрагменты рисунка на холст), с последующим написанием относительно несложного кода программ для «оживления» указанных элементов. Суть объектно-ориентированного программирования составляют три основных принципа: наследование, инкапсуляция и полиморфизм. *Наследование* означает возможность использования ранее созданного класса (класс-потомок), который будет наследовать все свойства ранее созданного класса с добавлением новых свойств. *Инкапсуляция* означает недоступность пользователя к внутренней структуре объекта. *Полиморфизм* задает правила взаимодействия со средой разработки и другими компонентами.

В основе программы для Windows лежит *управляемая событиями модель*. Выполнением кода программы управляют *события*, возникающие во время выполнения программы. Щелчок мыши на кнопке, открытие или закрытие формы, нажатие определенной комбинации клавиш на клавиатуре – при каждом из этих событий программе будет передано сообщение, которое вызовет запуск определенной, написанной для этого события процедуры. Таким образом, Windows информирует приложение о событии при помощи *сообщения*, которое описывает изменение, произошедшее в окружающей приложении среде. Каждое сообщение связывается с конкретным окном приложения. Окно, в свою очередь, связано с собственной *оконной процедурой*.

Чтобы было более понятно, рассмотрим пример создания в *Excel* приложения в виде Калькулятора с помощью макроязыка программирования *Visual Basic for Applications (VBA)*, встроенного в пакет *Microsoft Office*. На рис. 15 показано главное окно редактора VBA, в котором для наглядности размещены четыре локальных окна и панель элементов управления. Обычно при открытии

главного окна некоторые локальные окна не видны и их по мере необходимости можно открывать из пункта меню View (Показ).

Кратко опишем назначение и взаимосвязь окон.

Окно проектов (*Project*). В окне проектов отображается содержание открытых проектов VBA. Его можно использовать для получения быстрого доступа к необходимым программным модулям и процедурам, а также для копирования и перемещения модулей одного проекта в другой или для удаления модулей из проекта. Это окно очень похоже на окно программы *Проводник Windows*. Окно проектов позволяет работать с любыми проектами, которые открыты в данное время. Напомним, что в Excel проекты хранятся в книгах, поэтому в окне показаны проекты для каждой из книг. Каждый проект в окне представлен папками, содержащими различные виды объектов. Объектами являются рабочие листы открытой книги, сама книга Excel, модули, формы и др. Для книги и рабочих листов модуль создается автоматически. Эти модули используются, как правило, при написании кода *событий* для каждого рабочего

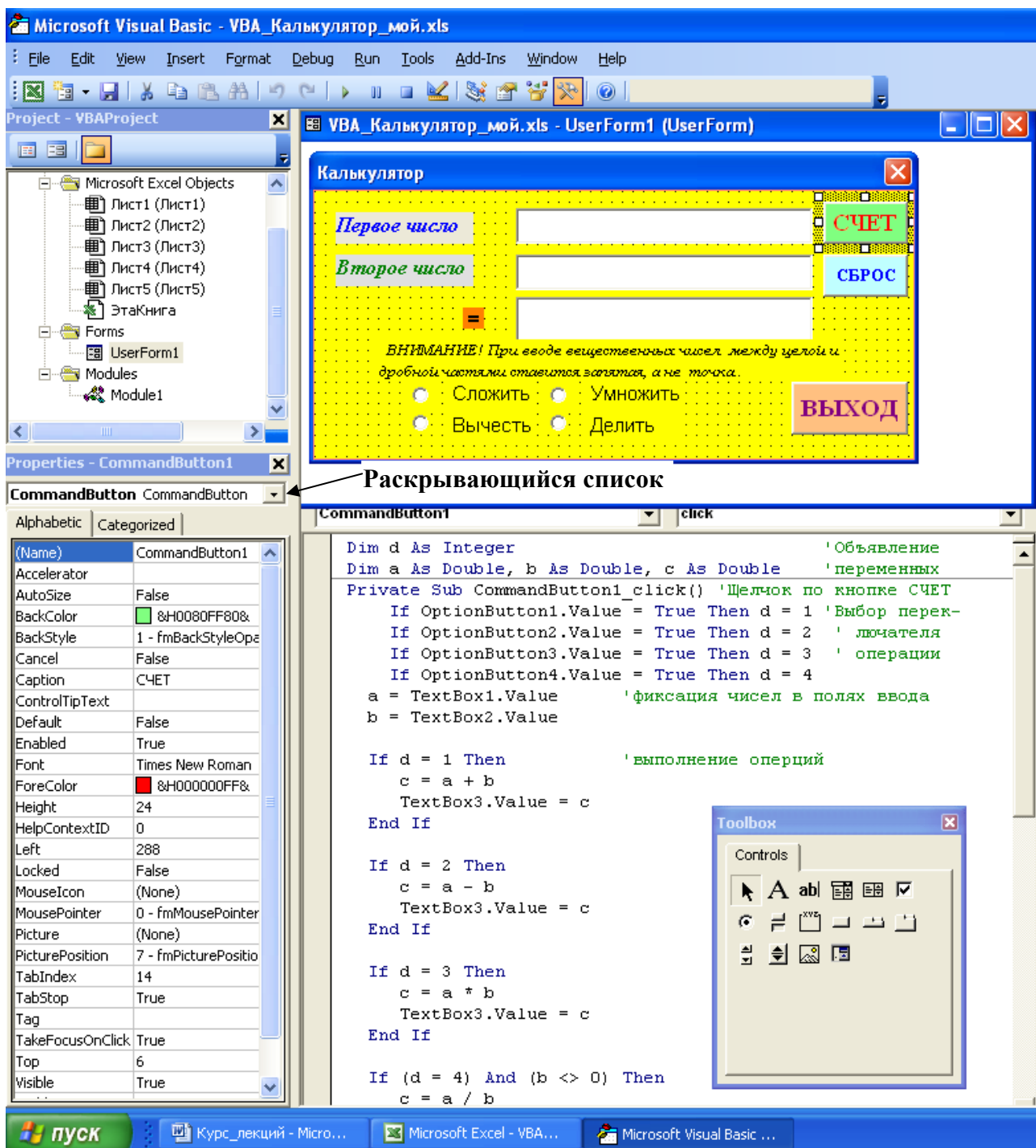


Рис. 15. Главное и локальные окна редактора VBA

листа. С помощью встроенных инструментов VBA для проектирования форм можно создавать пользовательские диалоговые окна (формы) для представления данных пользователю и получения данных от него. Каждая форма содержит модуль VBA с кодом обработки событий для этой формы. К любой рабочей книге можно добавить любое количество модулей. Копирование и перемещение модулей осуществляется теми же способами, что и для папок в окне программы *Проводник*. Для удаления модуля нужно выделить удаляемый модуль и выполнить команду *Удалить имя_модуля* из пункта меню *Файл*. Для

переименования проекта нужно выделить проект и выполнить команду *Свойства имя_проекта* из пункта меню *Сервис*. В появившемся диалоговом окне переименовать проект и закрыть окно, щелкнув по кнопке ОК.

Окно формы и панель элементов управления. Форма – это окно будущего приложения. Для показа окна с формой нужно выполнить команду *UserForm* из пункта меню *Insert* (Вставка). Когда Вы приступаете к новому проекту, VBA создает пустую форму в окне Object и присваивает ей заголовок UserForm1. В процессе разработки приложения форма служит как бы «холстом», на котором вы рисуете различные части программы – *объекты* или *элементы управления*: окна для ввода данных, кнопки, переключатели, линейки прокрутки и т. д. Элементы управления создаются на основе инструментов, перетаскиванием их с панели элементов управления (*Toolbox*). На этапе разработки вы управляете объектами на форме, изменяя их размер, позицию, цвет и другие атрибуты. Однако объекты неактивны, и код не исполняется (его еще требуется написать). Например, поместив объект «линейка прокрутки» на форму, можно изменить его размер и позицию, но прокрутить с его помощью ничего нельзя.

Окно свойств объектов (*Properties*). Окно свойств используется для просмотра и изменения свойств *на этапе разработки*. В левом столбце окна перечислены имена существующих свойств, а в правом – существующие значения свойств. Каждое свойство имеет определенный тип: текстовое значение, числовое значение, логическое значение и т. д. Некоторые свойства ограничены определенными значениями. Например, у объекта, обладающего свойством видимости (*Visible*), только два параметра: логические значения *True* (*Истина* – объект видим) и *False* (*Ложь* – объект невидим). Другие свойства могут иметь неограниченный набор параметров. Например, название окна формы. *Важно подчеркнуть, что имя объекта и название объекта это не одно и то же*. Если вы изменили имя, то название автоматически не изменится, и наоборот. Имя используется в программном коде как идентификатор объекта для ссылки на этот объект, а название отображается на объекте и видимо на экране. На рис. 15 имя кнопки (первая строка) – *CommandButton1*, а название кнопки (7 строка) – *Счет* (отображается в форме). Имя кнопки используется в названии процедуры *Private Sub CommandButton1_Click()*, а название кнопки отображается на ней самой. Хотя многие свойства допускается изменять как на стадии разработки, так и при выполнении программы, окно *Свойства* (*Properties*) активно только на стадии разработки.

Раскрывающийся список в верхней части этого окна, отражающий имена и типы объектов, включенных в приложение, называется *списком объектов*. Изначально здесь содержится лишь информация о форме, но по мере того, как вы размещаете на ней элементы управления, VBA регистрирует их в этом списке. Ниже списка объектов расположено *окно с текущими параметрами свойств выбранного объекта*. Их значения легко изменить: щелкните на значении в правом столбце и введите новое значение, или выберите один из предлагаемых вариантов из раскрывающегося списка, который появляется при щелчке мышью в конце строки существующих значений свойств.

На рис. 15 с помощью описанных элементов в окне редактора VBA подготовлена форма приложения (интерфейс пользователя) с именем **Калькулятор** – все его элементы на своем месте и имеют имена. Приложение можно «запустить» в таком состоянии: из пункта меню *Run* (Выполнить) по команде *Start*, или щелкнуть на одноименной кнопке ▶ панели инструментов. В результате все окна исчезнут, а на листе книги Excel появится окно Калькулятора. Его можно подвигать по экрану, можно пощелкать по кнопкам и увидеть, что они «живые» и реагируют на щелчки, хотя и ничего не происходит, так как программный код для них пока не написан.

Использование окна программы и составление ее кода. При программировании на языке VBA большую часть работы необходимо выполнять в окне программы. В этом окне код программы отображается для просмотра, редактирования и печати. Окно программы вызывается двойным щелчком мыши по форме приложения или по любому ее объекту. На рис. 15 показано окно с частью программного кода для выполнения четырех арифметических действий Калькулятором над двумя числами, введенными в текстовое поле 1 (TextVox1) и текстовое поле 2 (TextVox2), при выборе соответствующего переключателя (OptionButton) для выполнения конкретной операции при щелчке командной кнопки СЧЕТ (CommandButton1). Там же приведены и комментарии к коду.

На первый взгляд, окно программы выглядит и работает так же, как и обычное окно для редактирования текста. Если же ввести всего несколько строчек кода, то можно заметить, что набранный текст внимательно анализируется программой. Редактор автоматически определяет многие ошибки в коде и затем отображает предложения, как их исправить.

Код программы будет храниться в том же файле, что и экранные объекты формы. Как и со свойствами, любой объект можно связать с набором процедур, выполняемых в строго определенные моменты. *Процедура* – это группа операторов языка программирования VBA. Весь составленный *исполняемый код* обязательно помещается в какую-нибудь процедуру. Процедура, присвоенная объекту, связана с определенным *событием* или действием, поэтому называется *процедурой обработки события*. Событиями считаются, например, *Click* (Щелчок кнопкой мыши) или *Resize* (Изменение размера при изменении размера окна формы). *События происходят только в период выполнения программы, но не разработки*. «Вокруг» одного объекта может происходить не одно, а несколько событий.

Следует отметить, что для программирования задач искусственного интеллекта, например, при создании экспертных систем (см. лекцию 15) используются **логическое** программирование на основе языка *Prolog* и **функциональное** программирование на основе языка *Lisp*.

После того как программа написана и может что-то выполнять, это еще не окончание работы. Нужно убедиться, что она работает корректно во всех ситуациях, которые могут возникнуть при ее эксплуатации. Если при опытной эксплуатации возникают какие-то проблемы, то нужно их найти и устранить.

Таким образом, после завершения процесса программирования наступает очень важный этап тестирования и отладки (см. стр. 75 об ошибке фирмы Intel).

Тестирование – проверка работоспособности программного продукта при всех возможных вариантах исходных данных его использования с целью обнаружения ошибок.

Отладкой называется процесс поиска и устранения ошибок.

После отладки необходимо снова повторить тестирование, так как устранение одних ошибок нередко приводит к появлению новых, особенно, если программный продукт большой и сложный по структуре и функциональным возможностям. Для большей уверенности, что программный продукт не подведет пользователя, нужно провести тестирование на большом числе тестовых примеров, охватывающих все возможные ситуации.

Однако для больших программных комплексов число возможных ситуаций столь велико, что полному охвату не поддается, и стопроцентной гарантии, что в программе нет ошибок, дать нельзя. Можно добиться лишь того, что в большинстве часто встречающихся вариантов использования продукт «ведет себя» правильно. Для достижения этого существуют различные методы *тестирования*. Для серьезных разработок пользоваться нужно не одним из них, а всеми в комплексе. Для продуктов, выпускаемых на рынок, используют тестирование широким кругом потенциальных пользователей: выпускают так называемую «*бета-версию*» продукта и распространяют ее (обычно бесплатно) без гарантий надежной работы. Сбор информации об ошибках и отказах дает неоценимый материал для дальнейшей отладки.

Ошибки, которые могут быть в программе, принято делить на три группы:

- синтаксические;
- ошибки времени выполнения;
- алгоритмические.

Синтаксические ошибки наиболее легко устранимы – их обнаруживает компилятор, а программисту остается только внести изменения в текст программы и выполнить повторную компиляцию.

Ошибки времени выполнения возникают не при каждом запуске программы, а лишь при определенном наборе входных данных (см. примеры в Лекции 10). Для их выявления требуются тщательно подготовленные тестовые примеры.

С алгоритмическими ошибками дело обстоит иначе. Компиляция программы, в которой есть алгоритмическая ошибка, завершается успешно. При пробных запусках программа ведет себя нормально, однако при анализе результата выясняется, что он неверный. Для устранения алгоритмических ошибок может понадобиться не один год эксплуатации программы.

В большинстве инструментальных систем, например в *Borland Delphi*, программисту предоставляется мощное средство поиска и устранения ошибок в программе – *отладчик*. Отладчик позволяет выполнять трассировку программы, наблюдать значения переменных, контролировать выводимые программой данные.

Трассировка программы – процесс выполнения программы по шагам. Во время трассировки программист дает команду – выполнить очередную инст-

рукцию программы – и видит реальное выполнение каждой инструкции, т. е. реальный порядок выполнения инструкций разработанному алгоритму.

При отладке программ широко используется метод, который называют методом точек останова. Суть метода заключается в том, что программист помечает некоторые инструкции программы (ставит точки останова), при достижении которых программа приостанавливает свою работу, и программист может начать трассировку или проконтролировать значения переменных. Программисту в этом помогает отладчик – позволяет наблюдать значения переменных.

Заканчивая краткое рассмотрение принципов программирования, следует напомнить, что любая программа является продуктом (программным продуктом) и, следовательно, она должна соответствовать определенным стандартам и сопровождаться программной документацией. Создание программной документации – самый неприятный и тяжелый этап работы программиста. К сожалению, сейчас обычно этому не учат совсем, либо, в лучшем случае, не обращают на качество получаемых документов должного внимания. Дело в том, что в соответствии с Законом Российской Федерации «О стандартизации» стандарты становятся обязательными только на контрактной основе, (если на них есть ссылка в договоре на разработку или поставку программных средств).

Тем не менее, владение этим искусством является очень часто одним из важнейших факторов, определяющим профессионализм программиста. Заказчик не будет вникать в тонкости и особенности даже самой замечательной программы. Он будет сначала читать документацию. Во-вторых, грамотно созданный пакет программной документации может избавить от многих больших неприятностей. В частности, избавиться от необоснованных претензий к программному продукту и назойливых вопросов, можно просто отослав пользователя к документации. Стоит напомнить о многомиллионном иске к компании *IBM*. В 70-х г. прошлого века одно крупное издательство, неудовлетворенное качеством программного обеспечения, разработанного компанией *IBM*, предъявило ей иск. Компания *IBM* иск выиграла только благодаря тому, что предъявила в суде подписанное и согласованное обеими сторонами **Техническое задание** на разработку. Таких случаев, только более мелких, за недолгую историю существования ЭВМ можно привести много.

В СССР, как и в других странах, в 1970-е и 1980-е гг. была разработана «Единая система программной документации» – комплекс государственных стандартов, устанавливающих взаимосвязанные правила разработки, оформления и обращения программ и программной документации. Система ЕСПД включала около 30 ГОСТов, начиная с ГОСТ 19.001–77. «Общие положения» и заканчивая ГОСТ 19.604–77. «Правила внесения изменений в программные документы, выполненные печатным способом». ГОСТы определяли виды программ и программных документов, стадии разработки, общие требования к программным документам, методики испытаний программных продуктов, требования к содержанию и оформлению технического задания, текста программы, описания применения, пояснительной записки, различные руководства:

системного программиста, оператора, программиста, описание языка программирования и многое другое.

Частично эти стандарты морально устарели, в них не отражены некоторые современные тенденции разработки программ и программной документации, тем не менее, до настоящего времени не было официального заявления об отмене ЕСПД. К ней добавляются новые стандарты, например, еще в 1999 г. был принят и введен в действие Постановлением Госстандарта России от 23.12.99 г. № 675-ст дополнительно ГОСТ Р ИСО/МЭК 12207-99. Этот Государственный стандарт Российской Федерации содержит полный аутентичный текст международного стандарта ИСО/МЭК 12207-95 «Информационная технология. Процессы жизненного цикла программных средств». Стандарт предназначен для использования как в случае отдельно поставляемых программных средств, так и для программных средств, встраиваемых или интегрируемых в общую систему. Стандарт определяет процессы, работы и задачи, которые используются при приобретении системы, содержащей программные средства, или отдельно поставляемого программного продукта, при оказании программной услуги, а также при поставке, разработке, эксплуатации и сопровождении программных продуктов. Стандарт также определяет процесс, который может быть использован при определении, контроле и модернизации процессов жизненного цикла программных средств.

Стандарт применяется при приобретении систем, программных продуктов и оказании соответствующих услуг; а также при поставке, разработке, эксплуатации и сопровождении программных продуктов и программных компонентов программно-аппаратных средств как в самой организации, так и вне ее. Стандарт содержит те аспекты описания системы, которые необходимы для обеспечения понимания сути программных продуктов и услуг. Стандарт также применяется при двусторонних отношениях сторон и может в равной степени применяться, даже если обе стороны принадлежат к одной и той же организации. Диапазон применения может простирается от неформального соглашения о сотрудничестве до официально заключаемого контракта (договора). Стандарт может использоваться одной из сторон для самоконтроля.

Лекция 12

Классификация программ

Компьютер является электронным устройством. Элементы его представляют собой быстродействующие переключатели, которые могут под воздействием импульсов тока только переключаться из одного состояния в другое. Поэтому он в принципе не может обладать знаниями ни в одной области своего применения. Основной принцип обработки любой информации в компьютере аналогичен мозгу человека. Как человек хранит в своей памяти исходные данные и последовательность операций по их обработке (при необходимости заносит их с внешних носителей информации, например из справочников), так и компьютер хранит данные и алгоритмы их обработки в виде машинных кодов программ в своей памяти. Все его знания и все его умение сосредоточены в тех программах, которые он выполняет.

Меняя программы, можно выполнять на компьютере самую разнообразную работу:

- набирать и редактировать различные документы не только быстрее и эффективнее, чем на пишущей машинке, но и с использованием недоступных ранее возможностей;
- автоматизировать процесс проектирования сложнейших технических устройств и механизмов с выполнением графических работ, в том числе и трехмерной графики;
- быстро обрабатывать огромные массивы числовых и логических данных при решении самых различных задач, включая задачи оптимизации;
- превратить компьютер в рабочее место секретаря - референта, бухгалтера или агронома, менеджера по сбыту продукции, управления кадрами, по рекламе или сервису.

Поэтому для эффективного использования компьютера необходимо иметь представление о программах, знать назначение, возможности и свойства существующих на данный момент программ, которых в настоящее время разработано и эксплуатируется сотни тысяч, ясно представлять, где и какую программу следует использовать, знать их разновидности.

Напомним, что любая информация, в том числе и команды выполняемой компьютером программы, в его памяти представляется в виде наборов двоичных чисел, т. е. в виде наборов нулей и единиц. В общем виде каждая машинная команда состоит из нескольких двоичных чисел. Одно число – это код операции, которая выполняется по данной команде. Еще одно или два числа являются либо непосредственно операндами, над которыми выполняется данная операция, либо адресами (номераами ячеек памяти), в которых они хранятся. Этими числами могут быть адреса устройств, к которым обращается процессор. Таким образом, машинная команда в памяти ничем не отличается от числовых данных, текста или описания графического изображения и процессор распознает команду только благодаря размещению ее в специальном устройстве (регистре команд). Сколь бы сложную работу не выполнял компьютер (обработка цифровых или текстовых данных, графики или музыки), его процессор и устройства выполняют самые элементарные операции над двоичными числами. Они фактически передают от устройства к устройству, или от одной электронной схемы

к другой серии импульсов тока, которые переключают элементы схемы из одного состояния, хранящего какое-то двоичное число, в другое состояние, соответствующее новому числу. Таким образом, выполняя строго предписанные последовательности элементарных машинных команд, компьютер в итоге производит большую и разнообразную работу по обработке и преобразованию информации из одной формы в другую.

Несмотря на большое разнообразие и количество, все программы можно разделить на три основных класса:

1. *Системные программы* – предназначены для обслуживания компьютерной системы в целом. Они необходимы для пользователей любых категорий. Без этих программ работа на компьютере невозможна. Среди них особое место занимают *операционные системы (ОС)*, которые управляют компьютером, запускают в работу пользовательские программы, распределяют компьютерные ресурсы этим программам и управляют их работой, обеспечивают защиту доступа к данным, обеспечивают простое и легкое общение пользователя с компьютером (пользовательский интерфейс) и т. д. Операционная система является как бы **платформой**, на которой устанавливаются все другие программные продукты. Поэтому часто говорят: «приложение такое-то разработано, например, для платформы *UNIX* или *AS/400*». *Каждая программа, выполняемая на компьютере, пользуется услугами ОС, а потому может работать только на том компьютере, на котором установлена ОС, обеспечивающая ей эти услуги.* Например, программы, разработанные под управлением операционной системы *Windows*, работать на компьютере под управлением операционной системы *MS-DOS* не будут. Таким образом, ОС очень важна, так как определяет, с какими программами вы сможете работать на своем компьютере. От выбора ОС зависят производительность работы (однопрограммная или многопрограммная ОС), необходимые аппаратные средства, степень защиты данных и многое другое.

Для работы компьютеров в сети в качестве системных программ необходимы *сетевые программы*, обеспечивающие взаимодействие компьютеров между собой. В дополнение к своей операционной системе на каждой рабочей станции устанавливается *сетевая программа-клиент*, которая позволяет ей взаимодействовать с серверами. В свою очередь, серверы работают под управлением *сетевой программы-сервера* (например, *Novell NetWare*), которая обеспечивает их взаимодействие с рабочими станциями и позволяет последним обращаться к файлам, хранящимся на сервере.

Важной группой системных программ являются программы управления работой устройств – *драйверы*. Они расширяют возможности операционной системы, позволяя ей работать с теми или иными дополнительными внешними устройствами, обучая ОС новому протоколу обмена данными. Программы управления работой устройств, как правило, пишутся на языке программирования, приближенном к машинному, и привязаны к конкретному устройству. Они учитывают все его особенности и возможности работы. Драйверы для различных устройств часто поставляются вместе с новыми устройствами или платами управления (*адаптерами*) этих устройств. Многие драйверы унифицированы,

большое количество их включается в современные операционные системы. Например, при запуске *Windows* система автоматически определяет установленное оборудование и подключает необходимые драйверы, освобождая пользователя от забот по их установке.

К системным программам относится также большое количество программ-*утилит*, или программ вспомогательного назначения. Среди них следует упомянуть следующие:

- программы для защиты информации от несанкционированного доступа и ограничения доступа к ней;
- программы – *антивирусники*, позволяющие бороться с компьютерными вирусами, например пакет Касперского, полидетектор *VirusScan*, детекторы *AIDSTEST* Д. Лозинского, *Dr. Web* И. Данилова и другие;
- программы упаковки информации, или *архиваторы*, позволяющие объединять несколько файлов с различной информацией в одном «упакованном» файле в сжатом виде, например *WinRar*, *WinZip*, *Arj*, *Zoo* и др.;
- программы *резервирования*, позволяющие быстро, в сжатом виде, копировать информацию с жесткого несъемного диска (*винчестера*) на съемные носители для хранения на случай повреждения ее на винчестере, например *Backup*;
- *диагностические* программы для проверки работоспособности устройств компьютера, выявления и поиска неисправностей;
- программы *форматирования* дисковых накопителей и управления памятью, обеспечивающие более гибкое использование оперативной памяти и еще большое число программ различного назначения.

2. *Инструментальные системы*, или системы программирования, базирующиеся на том или ином языке высокого уровня программирования. Они включают в себя средства ввода и редактирования текста, трансляторы, ассемблеры и макроассемблеры, редакторы компоновки и загрузки программ, отладчики программ и др. С их помощью создаются все другие программы, в том числе и новые инструментальные средства, перечисленные выше системные программы и программы пользователя.

С каждой операционной системой связывается относительно стандартный набор инструментальных систем. Различия имеются лишь в версиях языков высокого уровня и в форматах объектных и загрузочных модулей. Наиболее распространены инструментальные системы *TURBOPASCAL 7.0*, *C++*, *VISUAL BASIC 8.0*, *DELPHI 6.0*.

3. *Прикладные программы или приложения*. Приложение является комплексом программ, который предоставляет пользователю на компьютере удобную, «дружественную» среду для создания документа определенного типа. Документом может быть текст, рисунки, таблицы, музыкальные фрагменты и др. Это проблемно-ориентированные программы, рассчитанные на решение определенных проблем и на определенные категории *пользователей (user)*, которые не обязаны знать тонкости и особенности устройств компьютера и уметь программировать. К ним относятся студенты, учащиеся, различного рода специалисты – врачи, юристы, менеджеры, работники налоговой службы, служащие

банков и т. д. Для них только на персональных компьютерах разработаны и используются сотни тысяч различных прикладных программ для самых различных применений.

Многие приложения разработчики объединяют и создают *пакеты прикладных программ (ППП)*. ППП являются наиболее динамично развивающейся частью программного обеспечения. Ярким примером может служить широко распространенный во всем мире ППП *Microsoft Office*, в состав которого входит порядка семи приложений офисного назначения. Наибольшее количество ППП создано для IBM PC-совместимых компьютеров с операционной системой *WINDOWS*.

Одни из прикладных программ являются программами *общего назначения* для любых пользователей, например, текстовые и графические редакторы, табличные процессоры, системы управления базами данных, системы автоматизированного проектирования и т. д. Другие – *узкоспециализированные*, предназначенные для пользователей определенного класса: программы учета кадров или учета материальных ценностей и др. Перечислим некоторые наиболее широко используемые в повседневной практике прикладные программы:

- *редакторы текстов*, или текстовые процессоры – для подготовки документов различного назначения и их редактирования, в том числе и для подготовки Web-страниц, например, *WinWord* из пакета *Microsoft Office*, а для подготовки Web-страниц – *Microsoft Front Page*, которая сейчас входит в состав пакета *Microsoft Office*, или *Microsoft Internet Studio*;

- *табличные процессоры* – для обработки документов табличной формы, в которых можно производить различные вычисления, строить диаграммы и графики, проводить аналитические исследования, решать оптимальные задачи и т. д. К ним относятся *Quattro Pro*, *1-2-3* и др. Наиболее широко в мире распространен табличный процессор *Excel* из пакета *Microsoft Office*;

- *системы управления базами данных (СУБД)* – для накопления и обработки больших массивов разнообразной структурированной информации. Наиболее общими операциями, которые выполняются средствами СУБД, являются поиск, редактирование, добавление и удаление данных. Из базы данных можно быстро и легко по запросу выбрать любую, имеющуюся в ней информацию для формирования какой-либо справки или отчета. К СУБД относятся, например *DB2*, *ORACLE*, *Paradox*, *Access* из пакета *Microsoft Office*;

- *экономического назначения* – бухгалтерского учета, финансового анализа, банковских операций, программы для торговых фирм и др. Например, *Инфо-бухгалтер*, *1С: Бухгалтерия*, *1С: Предприятие*, *Бухгалтер* фирмы Атлант-Информ, для анализа инвестиционных проектов – *Альт-Инвест*, *Альт-Анализ* и др. К этой группе можно отнести и различные *органайзеры* – программы управления проектами, позволяющие спланировать проект любой величины и сложности, эффективно распределить людские, финансовые и материальные ресурсы, составить оптимальный график работ и проконтролировать его исполнение. К ним можно отнести *Time Line*, *Microsoft Project*, *CA – Super Project* и еще большое количество подобных программ;

- *системы автоматизированного проектирования (САПР)* и конструирования различных механизмов, устройств, деталей и их вычерчивания, напри-

мер *AutoCAD, CadMex, Grafic CAD Professional, Inventor, Mechanical Desktop, DesingCAD, Компас*;

- *системы тестирования электронных и цифровых устройств*, например, *WorkLab 8.0* и *OrCad*;

- *информационно-справочные системы*, такие как «Гарант», «Консультант-плюс», «Юридический справочник»;

- *графические редакторы* – редакторы растровой графики используются для создания рисунков, редактирования и обработки изображений, оцифровки фотоизображений и т. д. К ним относятся *Adobe Photoshop, Corel Photo Paint, Picture Publisher, Aldus Photo Styler* и т. д. Редакторы векторной графики используются для профессиональной работы, связанной с технической и художественной иллюстрацией различных отчетов, печатных изданий. Это, например, *Adobe Illustrator, CorelDraw, Aldus Free Hand, Professional Draw* и др. Редакторы для 3-х мерной графики используются для создания и редактирования объемных изображений, например, *AutoDesk 3D Studio*;

- *распознавания символов* при переводе букв, цифр и графических изображений в их коды. Используются сканерами, например, программы *Fine Reader, Omni Page, Tiger, Cunie Form*;

- *для сочинения и воспроизведения музыки*, например *Score* и *Encore*;

- *издательские системы* – системы подготовки документов типографского качества для профессиональной издательской деятельности. Это, например, для рекламы и газет *PageMaker, Corel Ventura, Page Plus, QuarkXPress*, для подготовки книг и различной технической документации *Microsoft Publisher, Frame Maker* и др.

- *статистического анализа* данных по материалам органов государственной статистики, например, *Статистик-Консультант, StatGraphics*;

- *обучающие и контролирующие системы* для подготовки кадров массовых профессий и повышения квалификации специалистов;

- *пакеты программ мультимедиа* – для обработки и воспроизведения аудио- и видеoinформации, например, *Multimedia Viewer Kit, NEC MultiSpin* и др.

- *системы искусственного интеллекта и экспертные системы* для решения разного рода проблем диагностирования, прогноза и др. (см. лекцию 15);

- *различные словари, справочники, переводчики* с одного языка на другой, например словарь *Лингво*, переводчики *Stylus, Сократ, Jinn* и др.;

- *многочисленные компьютерные игры*, например, *Спайдермен, Сапер, различные Интернет-игры* и бесчисленное множество других.

Важно отметить, что программы 2-й и 3-й групп самостоятельно на компьютере работать не могут, если не инсталлирована операционная система.

Наиболее популярные у пользователей программы постоянно дорабатываются. В них добавляются новые возможности, исправляются обнаруженные ошибки, совершенствуются алгоритмы и т. д. Как правило, новое имя доработанным программам не дают, а сохраняют под старым именем в виде новой *версии*. Дискеты или компакт-диски, на которых изготовители поставляют программные средства, называют *дистрибутивными* (установочными). Как прави-

ло, программы устанавливать на компьютер необходимо с них, так как на них записывается служебная информация. Кроме того, при установке (*инсталляции*) программ с дистрибутива, для оптимизации выполнения они настраиваются на конфигурацию конкретного компьютера – того, на который устанавливаются. Часто дистрибутивы поставляются с защитой от несанкционированного копирования программ. В этом случае скопированная по всем правилам на винчестер программа может отказаться работать – необходимо знать ключ для копирования.

Лекция 13

Понятие о компьютерных вирусах и других вредоносных программах

Виды компьютерных вирусов. *Компьютерный вирус* – это специально написанная, вредоносная программа, способная размножаясь, самостоятельно внедрять свою копию в другие программы, системные области дисков, драйверы, документы, передаваться по сети на другие ресурсы и т. д. Выполняемые вирусом несанкционированные действия могут проявляться не сразу после попадания вируса в компьютер, а *обуславливаться* наступлением определенной даты, или определенного числа размножений, или сочетанием определенных условий, например записи зараженной программы на винчестер. Комбинация этих условий может быть достаточно сложной, чтобы затруднить ее определение. Вирусы могут нанести большой вред, вплоть до уничтожения всей информации, хранимой на диске компьютера, или заблокировать сетевой ресурс, поэтому рассмотрению их мы уделим немного больше внимания, чем уделили другим программам в предыдущей лекции.

Процесс внедрения вирусом своей копии в другую программу называется заражением. Возможными и единственными каналами проникновения вирусов в компьютер являются сменные носители информации и средства межкомпьютерной связи. Компьютерные вирусы не являются программами, использующими ошибки или недостатки конкретной операционной системы. Для обеспечения своего функционирования вирусу достаточно лишь нескольких вполне обычных операций, используемых большинством «нормальных» программ. Поэтому *принципиально* не может существовать универсального метода, защищающего операционную систему от распространения любого вируса.

С появлением первых электронных вычислительных машин возникла идея создания для них таких программ, которые бы самовоспроизводили себе подобные, что позволило бы значительно снизить затраты на создание программного обеспечения. В 1951 г. упоминавшийся уже американский математик Джон фон Нейман предложил метод создания таких программ. В результате многочисленных исследований появились безобидные игровые программы, основанные на технологии самовоспроизведения. На первом этапе подобные разработки носили исследовательский характер, когда авторы выполняли экс-

перименты, заручившись согласием пользователей. Исследования были направлены не на создание теоретической базы для будущего развития компьютерных вирусов. Наоборот, именно эти труды легли в основу многих более поздних работ по робототехнике и искусственному интеллекту.

Второй и наиболее важный этап в развитии вирусов связан с появлением *персональных компьютеров (ПК)*, как наиболее распространенных в мире. Атмосфера второго этапа носит мрачный характер противостояния пользователей безответственным или уголовным элементам. Воспользовавшись накопленными научными и практическими результатами, вандалы от компьютеров стали разрабатывать программы-вирусы, наносящие значительный ущерб пользователям.

Первые случаи массового заражения персональных компьютеров вирусами были отмечены в 1987 г. В настоящее время количество различных вирусов в мире приближается к полумиллиону, причем они появляются «как грибы после дождя». Появились различные антивирусные средства, и теперь ситуация напоминает печальную гонку вооружений и средств защиты от них. Первоначально вирусы были достаточно примитивными и носили скорее развлекательный характер (например, при работе на компьютере периодически сверху вниз по экрану падали символы, не нанося никакого вреда информации в памяти и на дисках). Современные же вирусы являются весьма изощренными и приносят большие неприятности пользователям (вплоть до уничтожения всей информации на дисках компьютера), поскольку пишутся высококлассными «технобандитами» в условиях массированного противодействия со стороны разработчиков антивирусных средств. Авторы вирусов додумались не только до *самошифрующихся вирусов и вирусов-невидимок* (стелс-вирусов), способных выдавать желаемое за действительное, но и до *мутирующих* (полиморфных) вирусов, которые способны изменять свой программный код (свое обличье) с целью предотвращения их обнаружения. Стелс-вирусы очень трудно обнаружить и обезвредить, так как они перехватывают обращения операционной системы к пораженным файлам и секторам дисков и подставляют вместо своего тела незараженные участки диска. Еще труднее обнаружить полиморфные вирусы, которые наиболее опасны и трудно определимы. Эти вирусы содержат код генерации шифровщика и расшифровщика. Таким образом, они модифицируют свой код в зараженных программах так, что два экземпляра одного и того же вируса могут не совпадать ни в одном бите.

Рассмотрим отдельные группы вредоносных программ в зависимости от среды их распространения

Классические компьютерные вирусы. К ним относятся программы, распространяющие свои копии по ресурсам локального компьютера с целью:

- последующего запуска своего кода при каких-либо действиях пользователя для искажения или уничтожения информации;
- дальнейшего внедрения в другие ресурсы компьютера.

Различают следующие типы вирусов:

- загрузочные, заражающие компоненты системной области дисков с программами первоначальной загрузки;

- файловые, инфицирующие программные файлы и документы;
- файлово – загрузочные, включающие черты первых двух типов.

Чтобы вирус мог инфицировать компьютер или начать распространяться, обычно на компьютере необходимо выполнить какое-либо действие, например, открыть инфицированное вложение электронной почты, или запустить зараженную им программу на выполнение.

В зависимости от способа активации различают *нерезидентные и резидентные* (находящиеся в памяти компьютера, пока он включен) вирусы. Загрузочные вирусы всегда являются резидентными, так как в память компьютера они попадают при включении машины, т. е. при загрузке операционной системы и остаются там на весь период работы.

Среди этой группы наибольшее распространение получили *макро-вирусы* для приложений *Microsoft Office* (Word, Excel, PowerPoint). При работе с документами и таблицами *MS Office* выполняет различные действия: открывает, сохраняет, печатает, закрывает документ, при этом Microsoft Word, например, ищет и выполняет соответствующие «встроенные макросы», которых в приложениях *Microsoft Office* большое количество для самых разных случаев. Автоматически (без участия пользователя) выполняются также макросы/функции, ассоциированные с какой-либо клавишей, либо моментом времени или датой. Макро-вирусы, поражающие файлы *Microsoft Office*, как правило, пользуются одним из перечисленных выше приемов – в вирусе либо присутствует авто-макрос (авто-функция), либо переопределен один из стандартных системных макросов, либо макрос вируса вызывается автоматически при нажатии на какую-либо клавишу или комбинацию их. Получив управление, макро-вирус переносит свой код в другие файлы, чаще всего в те, которые редактируются в данный момент.

Следует отметить также скрипт-вирусы, являющиеся подгруппой файловых вирусов. Эти вирусы пишутся на различных скрипт-языках (VBScript, JavaScript, PHP, BAT и др.). Они либо заражают другие скрипт-программы (например, командные и служебные файлы *Windows* или *Linux*), либо являются частями многокомпонентных вирусов. Эти вирусы могут также заражать файлы других форматов (например, HTML), если в них заложено выполнение скриптов.

Файловые вирусы не используют сетевых сервисов для проникновения на другие компьютеры. Копия вируса попадает на удаленные компьютеры только в том случае, если зараженный объект по каким-либо причинам оказывается активизированным на другом компьютере. Например, при заражении доступных дисков вирус проник в файлы, расположенные на сетевом ресурсе, или вирус скопировал себя на съемный носитель или заразил файлы на нем. Может оказаться, что пользователь отослал электронное письмо с зараженным вложением и т. д. В настоящее время эти вирусы практически исчезают, уступив место сетевым червям. *Как оценивают специалисты, опасность появления нового файлового вируса, способного вызвать глобальную эпидемию, сейчас практически близка к нулю.*

С бурным развитием общемировой сети Интернет все больше появляется качественно новых вредоносных программ, назначение и способы распростра-

нения которых совершенно иные, чем у файловых вирусов. Суть этих программ – обман пользователя. Появился даже новый термин, отражающий итоги работы таких программ – *фишинг*.

Фишинг в Интернете – это способ обманным путем получить с компьютеров пользователей личную или финансовую информацию через мошенническое сообщение электронной почты или поддельный веб-узел. Обычно фишинг в Интернете начинается с получения конфиденциального сообщения, похожего на официальное уведомление из надежного источника: от администрации банка, из компании кредитных карточек или от известного получателя интерактивного платежа. В электронном сообщении содержится ссылка на поддельный веб-узел, на который следует перейти и ввести личную информацию, например номер банковского счета или пароль, которые якобы украдены, или искажены. Эта информация впоследствии используется с целью кражи. Появились *шпионские программы*, которые могут выводить на экран рекламу (например, всплывающие объявления), изменять параметры компьютера без согласия пользователя. Некоторые шпионские программы могут не проявлять видимых симптомов. Их предназначение – сбор информации о вас и вашем компьютере. Шпионские программы могут устанавливаться веб-узлами, загруженными в память программами или программами, установленными с компакт-диска, флэш-карты или дискеты. Чаще всего шпионские программы устанавливаются вместе с бесплатным программным обеспечением, таким, как программы общего доступа к файлам, экранные заставки или панели поиска.

Для обмана пользователя широкое распространение получили *троянские программы*. Свое название они получили по имени легендарного Троянского коня. Подобно ему, модуль троянской программы прячется внутри другой вроде бы полезной или интересной программы, например экранной заставки, но осуществляет сбор информации (обычно системные пароли, пароли доступа к Интернету, IP-адреса, номера телефонов и др.) и передачу ее злоумышленнику. Часто троянские программы осуществляют электронный шпионаж и используются для кражи информации пользователей систем онлайн-платежей и банковских систем. Троянские программы обычно не распространяются сами по себе – чаще всего это делается с помощью вирусов, червей или загруженного на компьютер программного обеспечения.

По своей функциональности некоторые из *шпионских* программ во многом напоминают различные системы администрирования, разрабатываемые и распространяемые фирмами-производителями, но в них отсутствует предупреждение об установке и запуске такой программы. При запуске такая программа устанавливает себя в системе и затем следит за ней, при этом пользователю не выдается никаких сообщений о действиях ее в системе. В результате пользователь может и не знать о ее присутствии в системе, в то время как его компьютер открыт для удаленного управления. Такие троянские программы скрытого управления позволяют делать с компьютером все, что в них заложил автор: принимать или отсылать файлы по определенным адресам, запускать и уничтожать

их, выводить на экран различные сообщения, стирать информацию в памяти или на диске, изменять параметры компьютера, перезагружать компьютер и т. д.

Отдельно следует отметить группу троянских программ – *бэкдоров*, способных распространяться по сети и внедряться в другие компьютеры, как это делают сетевые черви. Отличие от вирусов-червей состоит в том, что они распространяются по сети не самопроизвольно (как черви), а только по специальной команде «хозяина», управляющего данной копией троянской программы.

Некоторые троянские программы – *Интернет кликеры* организуют несанкционированные обращения к интернет-ресурсам (обычно к Web-страницам). Достигается это либо посылкой соответствующих команд браузеру, либо заменой системных файлов, в которых указаны «стандартные» адреса интернет-ресурсов. Цели злоумышленников в этом случае могут быть следующие: увеличение посещаемости каких-либо сайтов с целью увеличения показов рекламы; привлечение потенциальных жертв для заражения вирусами или троянскими программами; организация DoS-атаки (Denial of Service) на какой-либо сервер. Сюда же можно отнести и троянские программы, скрытно осуществляющие анонимный доступ к различным интернет-ресурсам. Обычно они используются для рассылки *спама*, или «забиванию мусором» (бесполезными сообщениями) каналов Интернета – IRC-каналов, компьютерных пейджинговых сетей, электронной почты и т. д. К спаму можно отнести:

- индивидуальные сообщения, тематика которых не имеет к адресату прямого отношения;
- массовую рассылку почтовых сообщений пользователям, не желающим получать подобную корреспонденцию, причем разницы нет, коммерческая это реклама или просто полезная, по мнению отправителя, информация;
- помещение в телеконференцию Usenet, дискуссионного листа, гостевой книги сообщений, не имеющих отношения к заданной тематике, или сообщений, являющих собой прямую рекламу, если это разрешено установленными в конференции правилами;
- подписку пользователя на список рассылки теленовостей без его ведома или желания.

Существует еще целая группа различных троянских программ, например, «бомбы» в архивах – особо опасные для файловых и почтовых серверов. Если на сервере используется какая-либо система автоматической обработки информации – «архивная бомба» может просто остановить работу сервера. Программы-*dialers* предназначены для дозвона на различные платные ресурсы. Они непосредственного вреда системе не наносят, но могут привести к значительному финансовому ущербу для владельца телефонного номера, с которого идет дозвон на платный ресурс.

Очень обширная группа вредоносных программ – *сетевые черви*, формально именуемые *репликаторами*. Это компьютерный код, распространяющийся без вмешательства пользователя. Большинство червей содержится во вложениях электронной почты, которые инфицируют компьютер при их открытии. Червь сканирует инфицированный компьютер в поисках файлов с адреса-

ми электронной почты: адресные книги или временные веб-страницы. Червь использует адреса для рассылки инфицированных электронных сообщений и часто воспроизводит (или *подделывает*) адреса в последующих электронных сообщениях, чтобы эти инфицированные сообщения выглядели как настоящие письма от знакомых. Черви распространяются автоматически через электронную почту и сети, зачастую перегружая их, прежде чем удастся выяснить причину.

Если вирусы наносят ущерб посредством лавинообразного саморазмножения и явного разрушения информации, а троянские программы скрытого управления позволяют делать с компьютером все, что в них заложил автор, то основная функция вирусов-червей – *взлом атакуемой информационной системы*, т. е. преодоление защиты с целью нарушения безопасности и целостности системы. Этот вид нарушения безопасности представляет наибольшую угрозу, так как ему в любой момент может подвергнуться любая локальная или глобальная сеть или информационная система, например система разведки государства или крупного банка. Их основная особенность состоит в способности к размножению без внедрения в другие программы. Наибольший всплеск появления различного рода сетевых червей, поразивших компьютерный мир, пришелся на 2000–2001 г., когда вирусописателями были найдены и использованы бреши в таких операционных системах, как Linux и Windows 2000. Жертвами вирусов пали в это время такие графические пакеты, как Corel DRAW, Corel PHOTOPAINT, COREL VENTURA и популярное приложение для создания диаграмм и блок-схем Visio. Репликаторы создаются с целью:

- проникновения на удаленные компьютеры;
- запуска своей копии на удаленном компьютере;
- дальнейшего распространения на другие компьютеры в сети.

Репликаторы могут иметь начинку, состоящую из классических вирусов. Для своего распространения сетевые черви используют разнообразные компьютерные и мобильные сети и услуги: электронную почту, системы обмена мгновенными сообщениями, файлообменные (P2P) и IRC-сети, локальные вычислительные сети, сети обмена данными между мобильными устройствами (телефонами, карманными ПК) и т. д. Большинство известных червей распространяется в виде файлов: вложение в электронное письмо, ссылка на зараженный файл на каком-либо Web – или FTP – ресурсе в ICQ – и IRC – сообщениях, файл в каталоге обмена P2P и т. д. Так называемые «безфайловые» или «пакетные» черви распространяются в виде сетевых пакетов, проникают непосредственно в память компьютера и активизируют свой код.

В этом отношении можно провести аналогию между программой-червем и шариковой бомбой. Для проникновения на удаленные компьютеры и запуска своей копии черви используют различные методы: социальный инжиниринг (например, текст электронного письма, призывающий открыть вложенный файл), недочеты в конфигурации сети (например, копирование на диск, открытый на полный доступ), ошибки в службах безопасности операционных систем и приложений и т. д. Некоторые черви способны заражать выполняемые файлы на локальном диске, т. е. имеют свойства классического компьютерного вируса, или содержат функции троянских программ.

К следующей группе можно отнести *хакерские утилиты* и другие вредоносные программы, например:

- утилиты автоматизации создания вирусов, червей и троянских программ (конструкторы). Эти утилиты позволяют генерировать исходные тексты вирусов, объектные модули, или непосредственно зараженные файлы. Некоторые конструкторы снабжены стандартным оконным интерфейсом, где можно при помощи системы меню выбрать тип вируса, поражаемые объекты, наличие или отсутствие самошифровки, противодействие отладчику, выбрать эффекты, сопровождающие работу вируса и т. д.;

- библиотеки программ, разработанные для создания вредоносного программного обеспечения;

- хакерские утилиты скрытия кода зараженных файлов от антивирусной проверки (шифровальщики файлов);

- «злые шутки», затрудняющие работу с компьютером;

- программы, сообщающие пользователю заведомо ложную информацию о своих действиях в системе и т. д.

И, заканчивая рассмотрение вредоносных программ, следует отметить такой экзотический класс, как вредоносные программы для персональных карманных компьютеров (КПК). Стремительный рост популярности ОС *Windows Mobile 2003* для КПК, широкие возможности сетевой коммутации данных компьютеров и наличие среды разработки приложений (*.NET framework*) неминуемо приведут к появлению не только троянских программ (для системы *PalmOS* они уже существуют), но и их более опасных разновидностей, не исключая и сетевых червей.

Кто и почему создает вредоносные программы. В прошлом вирусы создавали студенты и школьники, изучившие язык программирования, но не нашедшие более достойного применения для самоутверждения. Значительная часть подобных вирусов не распространялась, они гибли вместе с дискетами, на которых хранились. Жизнь подобных вирусописателей стала заметно проще с развитием Интернета и появлением многочисленных Web-сайтов, ориентированных на обучение написанию вирусов. Со временем, набравшись опыта, эти люди стали талантливыми программистами и стали запускать в мир «профессиональные» вирусы. Отдельно можно выделить «исследователей», довольно сообразительных программистов. Эти программисты пишут вирусы не ради собственно вирусов, а скорее ради «исследования» потенциалов «компьютерной фауны». Часто авторы подобных вирусов не распространяют свои творения, однако активно пропагандируют их идеи через многочисленные Интернет-ресурсы, вследствие чего вскоре эти идеи появляются в новых вирусах. Подобные категории вирусописателей в последние годы становятся все менее и менее актуальными. Доля подобных вирусов и троянских программ занимает не более 10% «материала», заносимого в антивирусные базы данных. *90 % появления новых вирусов связано с коммерческими целями. Шантаж, вымогательство и рэкет – вот самый распространенный вид угроз в Интернете в настоящее время.*

Мелкое воровство. С появлением и популяризацией платных услуг Интернет-серверов начинает проявляться повышенный интерес к получению дос-

тупа в сеть за чужой счет, т. е. посредством кражи чьего-либо логина и пароля (или нескольких логинов и паролей с различных пораженных компьютеров) путем применения специально разработанных троянских программ, о чем говорилось выше. «Мелкими воришками» создаются также троянские программы других типов: ворующие регистрационные данные и ключевые файлы различных программных продуктов (для установки не сертифицированных копий их), часто – сетевых игр.

Криминальный бизнес. Наиболее опасную категорию вирусописателей составляют хакеры-одиночки или группы хакеров, которые осознанно создают вредоносные программы с единственной целью – получить чужие деньги (рекламируя что-либо или просто воруя их), получить ресурсы зараженного компьютера (опять таки, ради денег – обслуживание спам-бизнеса или организация DoS-атак с целью дальнейшего шантажа). Обслуживание рекламного и спам-бизнеса – один из основных видов деятельности данных хакеров.

Вторым основным видом деятельности подобных вирусописателей является создание, распространение и обслуживание *троянских программ-шпионов*, направленных на воровство денежных средств с персональных и корпоративных «электронных кошельков» или с обслуживаемых через Интернет банковских счетов. Троянские программы собирают информацию о кодах доступа к счетам и пересылают ее своему хозяину.

Третьим типом криминальной деятельности является *Интернет-рэкет*. Это организация массированной DoS-атаки на один или несколько серверов с последующим требованием денежного вознаграждения за прекращение атаки. Обычно под удар попадают компании, бизнес которых напрямую зависит от работоспособности их Web-сайтов. Это интернет-магазины, букмекерские конторы и др. Труд хакеров используется также для решения таких задач, как системы навязывания электронной рекламы, различные «звонилки» на платные телефонные номера, утилиты, периодически предлагающие пользователю посетить те или иные платные Web-ресурсы и т. д.

Следует особо отметить, что наибольшую опасность таит не только объединение в руках злоумышленников различных технологий создания вредоносных программ, но всевозрастающие криминализация и организованность преступного бизнеса в Интернете. Она опасна тем, что согласна потратить деньги и нанять профессионалов, способных отыскать новую уязвимость в распространенном программном обеспечении и создать вредоносный код, который будет красть конфиденциальные сведения.

Признаки заражения компьютера. В качестве симптомов заражения компьютера вирусом выделяют:

- вывод на экран непредусмотренных сообщений или изображений;
- подача непредусмотренных звуковых сигналов;
- неожиданное открытие и закрытие лотка CD-ROM-устройства;
- появление предупреждений о попытке какой-либо программы вашего компьютера выйти в Интернет, хотя вы это никак не инициировали (при наличии на вашем компьютере межсетевого экрана);

- произвольный, без вашего участия, запуск на компьютере каких-либо программ.

Кроме того, есть некоторые характерные признаки поражения вирусом через электронную почту:

- друзья или знакомые говорят вам о сообщениях от вас, которые вы не отправляли;
- в вашем почтовом ящике находится большое количество сообщений без обратного адреса и заголовка.

Не всегда такие признаки вызываются наличием вируса. Иногда они могут быть следствием других причин. Например, в случае с почтой зараженные сообщения могут рассылаться с вашим обратным адресом, но не с вашего компьютера.

Есть также косвенные признаки заражения вашего компьютера:

- частые зависания и сбои в работе компьютера;
- медленная работа компьютера при запуске программ;
- невозможность загрузки операционной системы;
- разрушение файловой структуры на диске;
- частое обращение к жесткому диску (часто мигает лампочка на системном блоке);
- Интернет-браузер «зависает» или ведет себя неожиданным образом (например, окно программы невозможно закрыть).

Возможно, на компьютере установлено какое-либо шпионское программное обеспечение, если:

- изменилась домашняя страница по умолчанию, указатель мыши или поисковая программа;
- при вводе адреса определенного веб-узла (например, поисковой системы) без предупреждения выполняется переход на другой веб-узел;
- в веб-обозревателе появились нежелательные новые панели инструментов, ссылки или элементы списка избранного, которые не добавлялись пользователем;
- отображается множество всплывающих рекламных объявлений, даже если нет подключения к Интернету.

Защита компьютера от вредоносных программ.

Защита от фишинга. Приведем некоторые простые советы по защите от фишинга в Интернете.

- Никогда не сообщайте личную информацию по запросу в сообщении электронной почты, мгновенном сообщении или всплывающем окне.
- Используйте только веб-узлы, предоставляющие заявление о конфиденциальности или сведения о способах использования личной информации.
- Не щелкайте ссылки в сообщениях электронной почты и мгновенных сообщениях от незнакомых людей и все остальные подозрительные ссылки. Поскольку даже сообщения от друзей и членов семьи вполне могут оказаться поддельными, следует выяснять у отправителей, действительно ли они послали сообщения.
- Регулярно проверяйте финансовые отчеты и кредитную историю и сообщайте о любых подозрительных действиях.

Некоторые шпионские программы могут не проявлять видимых симптомов, поэтому установка антишпионских программ, например *Защитник Microsoft Windows*, позволит осуществлять сканирование компьютера с целью обнаружения и удаления шпионских программ. Для установки этой программы зайдите по адресу: <http://www.microsoft.com/athome/security/spyware/software> на веб-узле *Microsoft* (страница может быть на английском языке).

- Регулярно обновляйте операционную систему *Windows* и *Internet Explorer* через *Центр обновления Windows* из пункта меню *Сервис* обозревателя. Корпорация *Microsoft* периодически выпускает специальные обновления безопасности, которые помогут защитить компьютер. Эти обновления могут предотвратить вирусные и хакерские атаки, закрывая потенциально опасные точки входа. Наилучший способ своевременного обновления *Windows* – включить автоматическое обновление. Для получения дополнительной информации посетите веб-узел (<http://update.microsoft.com>).

Для противодействия компьютерным вирусам могут использоваться меры как организационного, так и технического характера.

К организационным мерам относятся осторожность и разборчивость в связях, как по сети, так и при использовании переносных дисков и других устройств. Многие вирусы содержатся во вложениях сообщений электронной почты и начинают распространяться сразу после открытия вложения электронной почты. Поэтому, прежде чем открывать сообщение или вложение электронной почты, рассмотрите следующие вопросы.

1. Происходил ли обмен электронными письмами с этим пользователем ранее? если получено электронное письмо от кого-то, кто вам известен, но с кем никогда не велась переписка, подумайте, есть ли какая-нибудь причина получения письма именно сейчас, в особенности, если письмо содержит вложение или ссылку на веб-узел.

2. Известен ли вам и насколько вызывает доверие отправитель сообщения электронной почты? Следует соблюдать осторожность при получении писем электронной почты от неизвестных лиц или компаний

3. Соответствует ли тема сообщения его отправителю? Если тема сообщения не имеет смысла, это может указывать на то, что тема была сгенерирована вирусом.

При отрицательных ответах на все перечисленные выше вопросы, возможно, самое лучшее решение – удалить письмо. Если отправитель письма известен, но оно вызывает недоверие, без колебаний отправьте другое сообщение этому лицу с вопросом, действительно ли им было послано это электронное письмо. Намного проще выполнить проверку перед открытием сообщения, чем потом производить удаление вирусов с компьютера.

Не все узлы в Интернете безопасны, поэтому соблюдайте осторожность по отношению к веб-узлу при следующих условиях:

- предлагается бесплатная регистрация на веб-узле, но при этом требуется ввести подробные личные сведения, предоставлять которые вы не хотите или не считаете нужным;
- ссылка на этот узел прислана незнакомым пользователем по электронной почте;

- на узле содержатся неправдоподобно выгодные предложения;
- предлагается ввести номер кредитной карточки, но на веб-узле нет указаний на безопасность транзакции;

К видам противодействия технического характера следует отнести:

- защита компьютера от проникновения вирусов – использование *брандмауэра*. Брандмауэр оповещает о наличии подозрительной активности при попытке вируса или червя подключиться к компьютеру. Он также позволяет запретить вирусам, червям и хакерам загружать потенциально опасные программы на компьютер;

- защита компьютера от деструктивных действий вирусов, в частности, защита данных от разрушений;

- нейтрализация вирусов, т. е. блокирование к способности размножения и уничтожение их.

Защита данных возлагается в основном на общесистемные программные и аппаратные средства, чтобы обеспечить:

- резервирование и архивирование файлов;
- резервирование системных областей дисков;
- дефрагментация файлов (непрерывные файлы проще восстановить);
- разграничение доступа к внешним носителям информации по записи.

Остальные виды противодействия поддерживаются главным образом специальными антивирусными средствами. Разумеется, антивирусные программы надо применять наряду с организационными мероприятиями и профилактическими мерами, позволяющими уменьшить вероятность заражения вирусом. Антивирусные программы можно разделить на следующие виды в соответствии с выполняемыми ими функциями, хотя в современных антивирусниках, как правило, присутствуют все перечисленные ниже компоненты.

Детекторы позволяют обнаруживать файлы, зараженные одним или несколькими вирусами. Некоторые детекторы выполняют эвристический анализ файлов и системных областей дисков, что часто позволяет обнаружить новые, неизвестные ранее детектору, вирусы. Многие детекторы позволяют также «лечить» зараженные файлы или диски, удаляя из них вирусы.

Ревизоры запоминают сведения о состоянии файлов и системных областей дисков, а при последующих запусках сравнивают их состояние с исходным. При выявлении несоответствия выводится на экран сообщение пользователю. Часто ревизоры можно настроить так, чтобы они выдавали сообщения только о подозрительных изменениях, не беспокоя лишним раз пользователя. Большинство ревизоров позволяют также «лечить» зараженные файлы или диски, удаляя из них вирусы.

Сторожа располагаются резидентно в памяти и проверяют на наличие вирусов запускаемые файлы и используемые сменные носители информации. При наличии вируса об этом сообщается пользователю. Кроме того, многие сторожа перехватывают те действия, которые используются вирусом для размножения и нанесения вреда и сообщают о них пользователю. Сторожа позволяют обнаружить многие вирусы на самой ранней стадии, когда в дисковод

вставляется зараженная дискета или в файле, полученном из сети, то есть когда вирус еще не успел размножиться и что-либо испортить.

К сожалению, ни один тип антивирусных программ по отдельности не дает полной защиты от вирусов. Однако совместное использование их дает неплохие результаты, так как они хорошо дополняют друг друга. Как правило, программы-сторожа должны работать на компьютере постоянно, детекторы – для проверки поступающих из внешних источников файлов и переносных запоминающих устройств (например, флэш-память), а ревизоры запускаются раз в день для выявления и анализа изменений на дисках.

Установка антивирусной программы и ее постоянное обновление позволяют защитить компьютер от вирусов. Антивирусные программы проверяют на наличие вирусов сообщения электронной почты, операционную систему и файлы. Новые вирусы появляются ежедневно, поэтому необходимо постоянно проверять обновления на веб-узле изготовителя антивирусной программы. Большинство антивирусных программ продаются с годовой подпиской, которую при необходимости можно продлевать.

В настоящее время в Интернете постоянно находятся российские антивирусники Лаборатории Касперского и Dr.Web Данилова. Они постоянно совершенствуются, почти ежедневно обновляются и антивирусные базы.

Если у вас на компьютере установлен антивирусник, а компьютер подключен к Интернету, то можно воспользоваться рекомендациями, приведенными в конце лекции 27. Для **одноразового лечения** можно воспользоваться бесплатной утилитой, находящейся по адресу <http://www.drweb.com>. лечащей от вирусов, рутиков, троянских и шпионских программ и разного рода вредоносных объектов. На этом сайте найдите ссылку «*Бесплатный сервис*», щелкните ее и на открывшейся странице найдите утилиту Dr. Web CureIt®, в которую «вживлен» антивирусник Dr.Web Данилова.

Dr.Web CureIt не требует установки, не конфликтует ни с одним антивирусником, а значит, на время сканирования компьютера не требуется отключать установленный антивирусник другого производителя. Утилита содержит самый последний набор дополнений к вирусной базе Dr. Web и обновляется один или несколько раз в час. Утилита определяет язык, используемый ОС. В настоящее время она поддерживает интерфейс на 34 языках мира. *Утилита вылечит инфицированную систему однократно, но она не является постоянным средством для защиты компьютера от вирусов.*

Действия пользователя при заражении компьютера вирусом:

1. Прежде всего, при обнаружении вируса и, в особенности, при уничтожении им какой-то информации очень важно не предпринимать поспешных действий, а сначала «остановиться, оглядеться». Поспешное восстановление обычно приводит не только к потере части файлов, но и к повторному заражению.

2. Если есть сомнения о наличии вируса на компьютере, то лучше выключить компьютер, прежде отключив его от локальной сети или Интернета, если он был к ним подключен, чтобы вирус, если он действительно есть, не продолжил своих разрушительных действий.

3. Если симптом заражения состоит в том, что Вы не можете загрузиться с жесткого диска компьютера, попробуйте загрузиться в режиме защиты от сбоев или с защищенного от записи диска аварийной загрузки Windows, который вы создали при инсталляции операционной системы на компьютер. При этом следует использовать только программы, хранящиеся на защищенных от записи носителях, гарантированно чистых от вирусов.

4. Прежде чем предпринимать какие-либо действия, сохраните результаты вашей работы на внешний носитель (дискету, сменный жесткий диск, CD-диск, флэш-карту и др.)

5. Если вы хотите установить на компьютер постоянно действующие антивирусные программы, скопируйте из Интернета (на другом, например у друга или на работе не зараженном компьютере) на флэш-карту антивирусник «Лаборатории Касперского» Personal <http://www/kaspersky.ru/trials>, который будет работоспособным две недели с момента установки. У антивирусника Касперского есть еще адрес: (<http://www.kaspersky.ru/homeuser?chapter=145592748>).

6. Получите последние обновления антивирусных баз. Если Вы будете использовать антивирусник Касперского, то получить обновления антивирусных баз можно на дискете или диске в «Лаборатории Касперского» или у ее дистрибьюторов. В Интернете можно скопировать и антивирусник Dr.Web Данилова, действующий также определенное число дней.

7. Установите рекомендуемый уровень настроек антивирусной программы, имеющейся у вас, и запустите полную проверку компьютера.

8. Хотелось бы предостеречь от поспешных «панических» действий по восстановлению информации или деактивации, без консультации со специалистами. Слишком часто при этом уничтожаются данные, которые вполне можно было бы спасти. Причем, чем больше повреждение, тем более продуманными и осторожными должны быть действия по восстановлению.

9. Любые рекомендации по поводу лечения не надо воспринимать как догму, так как вирусы появляются, чуть ли не ежедневно, и рекомендации из-за этого могут не соответствовать действительности.

Проще всего, если вирус еще не успел активизироваться. Например, при обращении к носителю программа-сторож может вывести сообщение, что диск заражен вирусом и предложит его удалить. Надо просто согласиться. В более сложном случае, когда вирус уже мог активизироваться, заразить или испортить какие-то данные на дисках, следует:

1. Выяснить сведения о вирусе, который обнаружен на компьютере, и оценить возможные последствия заражения. Так, если компьютер оказался заражен неопасным загрузочным вирусом, то надо просто его удалить. А устранение последствий заражения вирусом, изменяющим случайно выбранные участки диска, могут быть гораздо серьезнее. Обычно при этом приходится заново устанавливать все пакеты программ с дистрибутивов, а собственные данные с резервных копий, если, конечно, они были сделаны.

2. Пользоваться только программами - антивирусниками. Только антивирусники лучше всего знают, как вирус портит файловую систему и как ее восстановить.

3. Компьютер может быть заражен несколькими вирусами. Надо использовать все имеющиеся антивирусники поочередно и проверку осуществлять многократно.

4. Лучше всегда файл удалить, а не «лечить», если его можно восстановить с дистрибутива.

5. Проверку на наличие вируса лучше делать во всех файлах, в том числе документах и архивах, а не только в программных, так как вирус может быть в файлах баз данных, а также в документах, созданных, например, в текстовом редакторе Word или табличном процессоре Excel.

После того, как вирус выявили и удалили, надо проверить целостность файловой системы и поверхности дисков с помощью программ диагностики. Если повреждения значительны, то все, что можно спасти, надо скопировать, винчестер заново переформатировать и установить все пакеты программ с дистрибутивов, а данные – с резервных копий.

Лекция 14

Информационные системы, их назначение, структуры и признаки классификации

Слово «система» происходит от греческого *system*, что означает целое, составленное из частей или множества элементов, связанных друг с другом. Для системы характерны следующие основные свойства:

Сложность системы. Зависит от множества входящих в нее компонентов, их структурного взаимодействия, а также от сложности внутренних и внешних связей и динамичности.

Целостность системы. Означает, что функционирование множества элементов системы подчинено единой цели.

Делимость системы. Означает, что она состоит из ряда подсистем или элементов, выделенных по определенному признаку, который отвечает конкретным целям и задачам.

Многообразие элементов системы и различия их природы связаны с их функциональной специфичностью и автономностью. Например, в материальной системе объекта, связанной с преобразованием вещественно-энергетических ресурсов, могут быть выделены такие элементы, как сырье, основные и вспомогательные материалы, топливо, полуфабрикаты, запасные части, готовая продукция, трудовые и денежные ресурсы.

Структурированность системы. Определяет наличие установленных связей и отношений между элементами внутри системы, распределение элементов системы по уровням иерархии.

Термин «система» используется очень широко. Например, система образования, социально-экономическая система, энергетическая система, транспортная система, политическая система, система скорой медицинской помощи.

Объект изучения информатики – информационные системы. В настоящее время существует множество информационных систем (ИС) самого различного назначения. В общем случае *информационная система – это организационно упорядоченная совокупность документов, информационных технологий, в том числе с использованием средств вычислительной техники и связи, реализующих информационные процессы, т. е. процессы сбора, накопления, хранения, обработки и выдачи информации пользователям.* Так трактует это понятие Федеральный закон об информации.

Информационная система может быть создана для конкретного специалиста в виде автоматизированного рабочего места (АРМ), для офиса или крупной компании как, например, железная дорога, для региона (например, для Среднего Урала) или целой страны. Это могут быть прогнозирующие или советующие системы, управляющие системы в экономике или технических устройствах, системы автоматизированного проектирования, организационные или информационно-справочные системы и т. д. Одни системы *не производят качественного изменения информации* (например, учетные системы), другие системы анализируют информацию (аналитические системы) и *вырабатывают качественно новую информацию* – управляющие решения (управляющие системы).

Информационные системы обеспечивают ускорение потоков товара, снижение издержек производства, привлечение дополнительного числа клиентов фирмы, автоматизируют создание управленческих отчетов и т. д. Многие фирмы-поставщики продукции с помощью своих информационных систем регистрируют данные о клиентах, что позволяет определять устойчивые группы постоянных клиентов, их запросы с целью оперативного реагирования на них, посылать потенциальным клиентам различные предложения, рекламу, напоминания, предоставлять клиентам различные услуги: поставки продукции в кредит, со скидками, с отсрочкой платежей и т.д. Потребители становятся заинтересованными во взаимодействии с фирмой, а фирма тем самым привлекает дополнительное число клиентов, увеличивая поставки и соответственно свою прибыль. Кроме того, иметь большие запасы продуктов на складах фирмы очень невыгодно, а не иметь их невозможно. В информационной же системе заказчик прямо с терминала вводит свои пожелания по предоставляемому ему каталогу. Менеджеры поставщика, делая выборки по поступившим заказам, принимают оперативные управленческие решения по доставке заказчику нужного товара за короткий промежуток времени. Таким образом, экономятся огромные деньги на хранение товаров, ускоряется и упрощается поток товаров, отслеживаются потребности покупателей.

Информационные системы оперативного уровня являются как бы связующим звеном фирмы с внешней средой. Если система работает плохо, то фирма не получает информацию извне либо не выдает информацию клиентам, что ведет к потере клиентов и уменьшению количества сделок фирмы.

Информационные системы могут быть оперативной обработки транзакций* (*on-line transaction processing, OLTP*), например, системы заказа и бронирования билетов, справочные системы и т. д., либо оперативного анализа (*on-line analytical processing, OLAP*), в качестве которых могут быть информационные системы анализа и генерации отчетов для руководителя на основе накопленной информации за определенный промежуток времени.

Иногда системы *OLTP* в то же время могут выполнять и функции систем *OLAP*. Например, информационная система по продаже билетов может накапливать информацию о продаваемых билетах, анализировать архивные данные за длительные периоды, оценивая перспективы продажи всех билетов на тот или иной рейс с учетом сезонности, т. е. *эффективность и необходимость этого рейса*, назначать разумные цены на те или иные билеты с целью снижения уровня непроданных билетов и т. д.

В связи с переходом к рыночным отношениям стали уделять большое внимание вопросу стратегии развития и поведения фирмы, что способствовало коренному изменению во взглядах на информационные системы. Они стали рассматриваться как стратегически важные системы. На *стратегические* информационные системы появился большой спрос. Они позволяют принимать решения по выбору целей фирмы, ее задач, методов, продуктов и услуг, что дает возможность опередить конкурентов и наладить более тесное взаимодействие с поставщиками сырья и потребителями продукции. Эти системы, как правило, всегда приводят к изменению структуры управления или всей фирмы в целом, к внедрению новых производственных технологий, к изменению номенклатуры выпускаемой продукции и даже к изменению профиля фирмы, содействуя ее процветанию. Чтобы выжить в условиях рынка, фирмы должны быть постоянно готовыми к частым и непрерывным преобразованиям. Единственным критерием конкурентоспособности фирмы является возможность адаптироваться к изменениям рынка быстрее, чем соперники. Поэтому информационные системы должны обеспечивать сотрудников фирмы, клиентов, поставщиков и партнеров мгновенным доступом к информации фирмы, независимо от ее территориального местоположения.

Информационные системы стратегического уровня помогают высшему звену управленцев решать задачи, не поддающиеся строгому математическому описанию и алгоритмизации, осуществлять долгосрочное планирование. Основная задача управленца – сравнение происходящих во внешнем окружении изменений с существующим потенциалом фирмы.

В качестве примера информационно-организационной системы кратко рассмотрим систему «Экспортная карта Свердловской области» («ЭКСО»), презентация которой состоялась в конце 1998 г. Система вызвана к жизни формированием рыночной экономики в России и призвана помогать правительству

* Транзакция – 1. В диалоговых системах это прием порции данных (сообщения или запроса) от пользователя, ее обработка и выдача ответного сообщения. 2. В базах данных и файловых системах это выполнение элементарной целостной операции над данными (например, удаление, модификация или добавление записи), в течение которой база данных или файловая система находятся в некорректном состоянии.

Свердловской области в решении одной из важнейших внешнеэкономических задач – продвижению продукции предприятий Среднего Урала на внешние рынки сбыта. Уникальность системы состоит в следующем.

Во-первых, многие предприятия и имеют собственные страницы и серверы в сети Интернет, но их зарегистрировано уже десятки миллионов, и вероятность того, что заинтересованный клиент, незнакомый с Уралом, выйдет на сервер конкретного предприятия, очень невелика. Экспортные же возможности предприятий Свердловской области очень широки, они все представлены в системе, поэтому любой предприниматель сможет найти в системе ту продукцию, которая ему нужна. Сделать это он сможет легко и быстро, так как *вся информация сосредоточена в одном месте*, а оперативность – главное в современном бизнесе.

Во-вторых, потенциальному партнеру предоставляется вся необходимая информация о регионе, что позволяет ему действовать «с открытыми глазами», принимая во внимание ресурсный потенциал Среднего Урала, состояние социально-экономической и политической систем Свердловской области, особенности территориальной инфраструктуры, транспортную сеть региона и другие особенности.

В-третьих, повышается доверие зарубежных партнеров к представляемой информации. Им предоставляется возможность получения официальных консультаций от государственных органов управления. По желанию они могут обратиться за консультацией и к негосударственным экспертным организациям, информация о которых также представлена в системе.

В-четвертых, все заинтересованные лица могут получить самую свежую информацию об официальных мероприятиях губернатора и правительства Свердловской области в сфере международной деятельности. Это позволит в полной мере реализовать положения заключаемых международных соглашений и других договоренностей в сфере внешнеэкономической деятельности. Так, если при проведении любой международной выставки информация о предложениях участников будет заранее доведена до всех заинтересованных лиц и при этом им будет предоставлена возможность получения необходимых консультаций непосредственно от руководителей предприятий, а после завершения выставки ее «виртуальная копия» останется доступной для всех пользователей, то результативность выставки возрастет многократно.

Кратко рассмотрим основные разделы системы.

Организационная инфраструктура внешнеэкономической деятельности. В этом разделе приведена краткая характеристика основных субъектов управления и организации внешнеэкономической деятельности на территории Свердловской области с определением их функций и указанием каналов связи.

Нормативно-правовые акты. Приведены документы, определяющие условия осуществления внешнеэкономической, инвестиционной и хозяйственной деятельности на территории Российской Федерации в целом и Свердловской области в частности.

Аналитические материалы. Приведены информационно-аналитические обзоры социально-экономической ситуации в регионе, основных показателей

функционирования внешнеэкономического комплекса Свердловской области в динамике их изменений, научного, промышленного и ресурсного потенциала, политической системы, социально-демографической ситуации в области, территориально-административного деления.

Информация, приведенная в данном разделе, ориентирована на привлечение стратегических инвесторов, крупнейших зарубежных фирм, заинтересованных в долгосрочном и комплексном сотрудничестве и нуждающихся в получении полной, достоверной и комплексной информации о регионе для принятия решений. С другой стороны, этот раздел может быть использован территориальными органами управления внешнеэкономической деятельностью для оперативного получения необходимой информации. В связи с этим часть информации раздела доступна лишь для специального пользования.

Внешнеэкономический потенциал предприятий. Приведена информация обо всех основных предприятиях Свердловской области, участвующих во внешнеэкономической деятельности. Приведены номенклатура экспортных поставок, предложения по экспорту, потребность в импорте товаров и услуг, инвестиционные проекты, потребность в кредитных ресурсах для расширения оборотных средств и т. д. Информация структурирована как по отраслевому признаку, так и по ряду других, обеспечивающих заинтересованному пользователю возможность оперативного выхода на конкретное предприятие.

Внешнеэкономический потенциал региона. Приведена информация о всей продукции, экспортируемой с территории региона, а также о продукции, которая может быть востребована на внешнем рынке. Информация структурирована по кодам, при этом обеспечивается выход конкретного потребителя продукции, которая заинтересовала его, на все предприятия, производящие эту продукцию.

Инвестиционный потенциал региона. В этом разделе дана характеристика инвестиционных программ, реализуемых на территории Свердловской области, а также инвестиционных проектов отдельных предприятий и организаций. Система обеспечивает оперативный выход на инициатора проекта.

Официальные страницы. Приведена информация органов государственного управления внешнеэкономической деятельностью о важнейших мероприятиях в сфере международной и внешнеэкономической деятельности.

Открытые страницы для иностранных фирм. Приведена информация об иностранных фирмах, предоставляемых ими товарах и услугах, перспективных направлениях сотрудничества, конкретные предложения по экспорту и импорту товаров и услуг, предложения иностранных фирм по развитию внешнеэкономического сотрудничества, предложения инвестиционных фондов и других финансовых организаций.

Постоянно действующая виртуальная выставка товаров и услуг предприятий Свердловской области. В данном разделе визуальны представлены образцы товаров и услуг, производимых предприятиями Свердловской области, их технические и эксплуатационные характеристики, данные о конкурентоспособности продукции и т. д. В структуре раздела предусмотрена возможность проведения тематических выставок в соответствии с планом проведения реальных выставок, как на территории Свердловской области, так и за рубежом.

Информация о системе распространяется через торговые представительства Российской Федерации в зарубежных странах, через министерства и ведомства зарубежных стран, через торгово-промышленные палаты России и зарубежных стран, через представительства Свердловской области за рубежом, зарубежные представительства предприятий и организаций Свердловской области. Кроме того, информация о системе «ЭКСО» внесена во все действующие в сети Интернет информационно-поисковые системы и крупнейшие базы данных по внешнеэкономической деятельности. В сети Интернет система размещена по адресу www.Midural.ru, где с ней можно познакомиться более детально.

Структура информационной системы. Структуру информационной системы составляет совокупность отдельных ее частей, называемых *подсистемами*.

Подсистема – это часть системы, выделенная по какому-либо признаку.

Общую структуру информационной системы можно рассматривать как совокупность *обеспечивающих* подсистем независимо от сферы применения.



Рис. 16. Обеспечивающие подсистемы информационной системы

Сфера применения определяет *функциональные* подсистемы информационной системы, например подсистема бухгалтерского учета, подсистема учета выпуска продукции, подсистема учета кадров. Их мы рассматривать не будем.

Информационное обеспечение. Назначение подсистемы информационного обеспечения состоит в своевременном формировании и выдаче достоверной информации для принятия управленческих решений.

Информационное обеспечение – совокупность единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков, циркулирующих в организации (предприятии, фирме), а также методология построения баз данных.

Унифицированные системы документации создаются на государственном, республиканском, отраслевом и региональном уровнях. Главная цель – это обеспечение сопоставимости показателей различных сфер общественного производства. Разработаны стандарты, в которых устанавливаются требования:

- к унифицированным системам документации;
- унифицированным формам документов различных уровней управления;
- составу и структуре реквизитов и показателей;
- порядку внедрения, ведения и регистрации унифицированных форм документов.

Несмотря на существование стандартов, при обследовании большинства организаций и предприятий постоянно выявляется целый комплекс типичных недостатков:

- чрезвычайно большой объем документов для ручной обработки;
- одни и те же показатели часто дублируются в разных документах;
- работа с большим количеством документов отвлекает специалистов от решения его непосредственных задач;
- имеются показатели, которые создаются, но не используются, и многое другое.

Поэтому устранение указанных недостатков является одной из задач, стоящих при создании информационного обеспечения.

Схемы информационных потоков отражают маршруты движения информации и ее объемы, места возникновения первичной информации и места использования результатной информации. За счет анализа структуры подобных схем можно выработать меры по совершенствованию всей системы управления.

Построение схем информационных потоков, позволяющих выявить объемы информации и провести ее детальный анализ, обеспечивает:

- исключение дублирующей и неиспользуемой информации;
- классификацию и рациональное представление информации.

При этом подробно должны рассматриваться вопросы взаимосвязи движения информации по уровням управления. Следует выявить, какие показатели необходимы для принятия решений, а какие нет. К каждому исполнителю должна поступать только та информация, которая используется им в работе.

Методология построения баз данных основывается на теоретических основах их проектирования. Для понимания концепции методологии приведем основные ее идеи в виде двух последовательно реализуемых на практике этапов: *1-й этап* – обследование всех функциональных подразделений фирмы (предприятия) с целью:

- понять специфику и структуру ее деятельности;
- построить схему информационных потоков;
- определить информационные объекты и соответствующий состав реквизитов (параметров, характеристик), описывающих их свойства и назначение;
- проанализировать существующую систему документооборота;

2-й этап – построение концептуальной информационно-логической модели данных для обследованной на первом этапе сферы деятельности. В этой модели должны быть установлены и оптимизированы все связи между объектами и их реквизитами. Информационно-логическая модель является фундаментом, на котором будет создана база данных.

Для создания информационного обеспечения необходимо:

- ясное понимание целей, задач, функций всей системы управления организацией;
- выявление движения информации от момента возникновения и до ее использования на различных уровнях управления, представленной для анализа в виде схем информационных потоков;
- совершенствование системы документооборота;

- наличие и использование системы классификации и кодирования;
- владение методологией создания концептуальных информационно-логических моделей, отражающих взаимосвязь информации;
- создание массивов информации на машинных носителях, что требует наличия современного технического обеспечения.

Техническое обеспечение предназначено для реализации всех информационных процессов в системе.

Техническое обеспечение – комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы.

Комплекс технических средств составляют:

- компьютеры любых моделей;
- устройства сбора, накопления, обработки, передачи и вывода информации потребителю;
- оргтехника и устройства автоматического съема информации (например, определение износа колесной пары по высоте ребра при движении поезда). Оргтехника обеспечивает копирование, хранение и поиск документов, звукозапись их с помощью диктофонов и уничтожение;
- устройства передачи данных и линии связи;
- эксплуатационные материалы и др.

Предварительный выбор технических средств, организация их эксплуатации, технологические процессы обработки данных, технологическое оснащение обязательно оформляются документально. Документацию можно условно разделить на три группы:

- общесистемную, включающую государственные и отраслевые стандарты по техническому обеспечению;
- специализированную, содержащую комплекс методик по всем этапам разработки технического обеспечения;
- нормативно-справочную, используемую при выполнении расчетов по техническому обеспечению.

К настоящему времени сложились две основные формы использования технических средств: централизованная и частично или полностью децентрализованная.

Централизованное техническое обеспечение базируется на использовании в информационной системе больших ЭВМ (мейнфреймов) и вычислительных центров. Централизованная обработка и хранение данных обходятся дешевле, чем обслуживание распределенных систем обработки информации, состоящих из сотен и тысяч персональных компьютеров, которые объединены в локальные вычислительные сети. Кроме того, повышается надежность хранения информации за счет дублирования дисковых подсистем.

Децентрализация технических средств предполагает реализацию функциональных подсистем (учет производства, финансово-кредитная, бухгалтерская и др.) на персональных компьютерах непосредственно на рабочих местах. Децентрализация дает оперативность в выполнении информационных процессов и гибкость при решении задач, но при такой организации обработки информации эксплуатационные затраты выше, чем в первом случае.

Перспективным подходом следует считать, по-видимому, *частично децентрализованный* подход – организацию технического обеспечения на базе распределенных сетей, состоящих из персональных компьютеров и большой ЭВМ для хранения баз данных, общих для любых функциональных подсистем.

Математическое и программное обеспечение. Эта подсистема представляет собой совокупность математических методов, моделей, алгоритмов и программ для реализации целей и задач информационной системы, а также нормального функционирования комплекса технических средств (операционные системы, диагностические и антивирусные программы и др.).

К средствам математического обеспечения относятся:

- средства моделирования процессов управления;
- типовые задачи управления;
- методы математического программирования, математической статистики, теории массового обслуживания и др.

В состав *программного обеспечения* входят общесистемные (на информационную систему, а не на компьютерную) и специальные программные продукты, а также техническая документация.

К *общесистемному* программному обеспечению относятся программные комплексы, которые ориентированы на пользователей и предназначенные для решения типовых задач обработки информации (например, программы корректировки информации, структурирования и форматирования информации, поиска информации, распечатки информации и т. д.). Они служат для расширения функциональных возможностей компьютеров и обеспечивают контроль и управление процессом обработки данных.

Специальное программное обеспечение представляет собой совокупность программ, разработанных при создании конкретной информационной системы. В его состав входят пакеты прикладных программ, реализующие разработанные модели разной степени адекватности, отражающие функционирование реального объекта.

Техническая документация на разработку программных средств должна содержать описание задач, техническое задание на алгоритмизацию, экономико-математические модели на задачи, руководства пользователя, инструкции по эксплуатации, контрольные примеры и т. д. Программная документация, как уже указывалось в лекции 11, должна соответствовать ГОСТам.

Организационное обеспечение. Подсистема представляет собой совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации информационной системы.

Организационное обеспечение реализует следующие функции:

- анализ существующей системы управления предприятием, на котором будет использоваться информационная система, и выявление задач, подлежащих автоматизации;
- подготовку задач к решению на компьютере, включая техническое задание на проектирование ИС и технико-экономическое обоснование ее эффективности;

- разработку управленческих решений по составу и структуре предприятия, методологии решения задач, направленных на повышение эффективности системы управления.

Организационное обеспечение создается по результатам предпроектного обследования на 1-м этапе построения баз данных, с целями которого мы познакомились при рассмотрении информационного обеспечения.

Правовое обеспечение. *Подсистема правового обеспечения – это совокупность правовых норм, определяющих создание, юридический статус и функционирование информационных систем, регламентирующих порядок получения, преобразования и использования информации.*

Главной целью правового обеспечения является укрепление законности. В состав правового обеспечения входят законы государства, указы Президента, постановления государственных органов власти, приказы, инструкции и другие нормативные документы министерств, ведомств, организаций, местных органов власти. В правовом обеспечении можно выделить общую часть, регулирующую функционирование любой ИС, и локальную часть, регулирующую функционирование конкретной системы.

Правовое обеспечение этапов разработки информационной системы включает нормативные акты, связанные с договорными отношениями разработчика и заказчика и правовым регулированием отклонений от договора, если они возникнут. Правовое обеспечение этапов функционирования ИС включает:

- статус информационной системы;
- права, обязанности и ответственность персонала;
- правовые положения отдельных видов процесса управления;
- порядок создания и использования информации и др.

Информационные системы многообразны. Поэтому рассмотренный выше структурный признак классификации является не единственным, могут быть классификации ИС и по ряду других признаков. Рассмотрим некоторые из них.

Классификация по сфере применения или направлению деятельности. Информационные системы можно классифицировать по сфере применения: административные, учебные, производственные, медицинские, экологические, метеорологические и т. д. По направлению деятельности системы подразделяются на следующие две большие группы.

Информационные системы организационного управления предназначены для автоматизации функций управленческого персонала. К этому классу относятся наиболее широко распространенные системы:

- промышленной сферы (ИС объединениями, крупными фирмами, корпорациями, холдингами, технологическими процессами непрерывного, дискретного и периодического характера и др.);
- транспортных компаний всех видов транспорта;
- непромышленной сферы (банковские, фондового рынка, финансовые, гостиничные, торговыми фирмами, страховые, налоговые, таможенной службы и др.);
- научной сферы, образования, культуры (АСНИ, САПР, АИС «Библиотека», экспертные системы и др.).

Основными функциями этих систем являются: оперативный контроль и регулирование, оперативный учет и анализ, оперативное и перспективное планирование, бухгалтерский учет, управление сбытом и снабжением и другие организационные и экономические задачи. Они широко используются при организации поточных линий, изготовлении микросхем, на сборке, для поддержания технологического процесса в химической, металлургической и машиностроительной промышленности, для автоматизации функций инженеров-проектировщиков, конструкторов, архитекторов, дизайнеров при создании новой техники и технологии.

Интегрированные (корпоративные) ИС используются для автоматизации всех функций фирмы и охватывают весь цикл работ от проектирования до сбыта готовой продукции. Создание таких систем весьма затруднительно, поскольку требует системного подхода с позиций главной цели, например, получения максимальной прибыли, завоевания рынка сбыта и т. д. Мы о таких системах упоминали в самом начале рассмотрения ИС как о *стратегических* системах.

Классификация ИС по территориальному признаку:

- геоинформационные системы, например ИС Газпрома (охватывает территорию не только России);
- ИС региона, например рассмотренная в начале лекции система ЭКСО;
- ИС республики, области, округа;
- ИС города или района;
- ИС предприятия.

Классификация ИС по характеру использования информации:

Информационно-поисковые системы осуществляют ввод, систематизацию, хранение, выдачу информации по запросу пользователя без сложных преобразований данных, например, ИПС библиотеки, системы продажи билетов и др.

Информационно-решающие системы осуществляют все операции переработки информации по заданному алгоритму. По степени воздействия выработанной результатной информации на процесс принятия решений можно выделить два класса систем: управляющие и советующие.

Управляющие ИС вырабатывают информацию, на основании которой человек принимает решение. Для этих систем характерны тип задач расчетного характера и обработка больших объемов данных, например, система оперативного планирования выпуска продукции, система бухгалтерского учета и др.

Советующие ИС вырабатывают информацию, которая принимается человеком к сведению и не превращается немедленно в серию конкретных действий. Эти системы обладают более высокой степенью интеллекта, так как для них характерна обработка знаний, а не данных.

Чтобы иметь представление об этих системах, в следующей лекции кратко рассмотрим общие характеристики *экспертных систем*, их виды, типы решаемых задач, структуру и режимы работы.

Понятие об экспертных системах

Общая характеристика ЭС. Экспертные системы (ЭС) – это яркое и быстро прогрессирующее направление в области искусственного интеллекта. Это узкоспециализированные информационные системы. Каждая ЭС может решать задачи только в одной конкретной предметной области. Это не универсальные системы. Система, предназначенная для поиска неисправностей в сложном устройстве управления работой ядерного реактора, не может ставить медицинские диагнозы. Решение задачи в ЭС сопровождается понятными пользователю объяснениями, а качество получаемых решений не хуже, а чаще лучше достигаемого экспертами-профессионалами. Главное достоинство ЭС – возможность накапливать знания, сохранять их длительное время и обновлять. Термин «знания» означает информацию, принимающую форму фактов или правил, которые не всегда истинны или ложны. Иногда существует некоторая степень неуверенности в достоверности факта или точности правила. Многие правила ЭС являются *эвристиками*, т. е. эмпирическими правилами или упрощениями. Использование эвристических правил делает поиск решения намного более легким и более практичным. Экспертные системы используют эвристики, потому что задачи, которые они решают (например, поиск новых месторождений алмазов или планирование экспериментов в молекулярной биологии), как правило, трудны и не до конца понятны.

Экспертные системы не решают расчетных задач с применением систем уравнений по определенным алгоритмам. Задачи, решаемые в них, не поддаются строгому математическому анализу или алгоритмическому решению. В ЭС известен алгоритм обработки знаний, а не алгоритм решения задачи. Более того, алгоритм обработки знаний иногда бывает заранее неизвестным и строится по ходу решения задачи на основании некоторых эвристических правил, хранящихся в базе знаний ЭС. Поэтому применение алгоритма обработки знаний может привести к получению неожиданного, ранее не предусмотренного результата.

ЭС относят к системам искусственного интеллекта общего назначения – системам, которые не только исполняют заданные процедуры, но на основе метапроцедур поиска генерируют и используют процедуры решения новых конкретных задач. Выходные результаты ЭС являются качественными, а не количественными.

ЭС – это набор программ, выполняющий функции эксперта при решении задач из некоторой предметной области (медицины, химии, геологии, радиоэлектроники и т. д.), как и эксперт-профессионал, в процессе своей работы они оперируют со знаниями. Они выдают советы, проводят анализ нечисловых данных и ситуаций, выдвигают и отбрасывают гипотезы, оценивают достоверность фактов, самостоятельно пополняют свои знания, контролируют их непротиворечивость, дают консультации, ставят диагноз и т. д.

Главным достоинством ЭС является возможность накопления знаний и сохранение их в течение длительного времени. В отличие от человека к любой информации ЭС подходят объективно, а при решении задач, требующих обработки большого объема знаний, возможность возникновения ошибки при переборе очень мала.

Первая особенность ЭС состоит в том, что они предназначены для пользователей, незнакомых с искусственным интеллектом, программированием, логикой и т. д. Для таких пользователей ЭС выступает как некий инструментарий, помогающий им в повседневной работе. Общения с системой должны быть так же просты, как просты, например, работа с телевизором, стиральной машиной или музыкальным центром.

Вторая особенность экспертных систем заключается в следующем. Во многих видах человеческой деятельности используемые знания далеко не всегда могут быть четко формализованы. Точнее, наряду со знаниями, полученными из учебников, инструкций, учебных фильмов и т. д., существуют *навыки и умения*, овладеть которыми можно, только работая вместе с теми, кто ими уже овладел. Любой профессионал высокого уровня (врач, геолог, инженер, экономист, юрист) – отличается от новичка, овладевшего знаниями в объеме учебных программ учебного заведения, именно этими навыками, опытом и умением.

Повышение профессионального мастерства новичков (и вообще повышение квалификации и переподготовка кадров) – огромная социальная задача, на решение которой тратятся большие финансовые и материальные средства, особенно в условиях рынка, когда фирмы часто перепрофилируют свою деятельность.

Чтобы решить эту задачу, надо научиться извлекать из профессионалов-экспертов те знания, которые не зафиксированы в форме, пригодной для массового распространения – в книгах, наглядных пособиях, инструкциях и др. Само название экспертных систем указывает на то, что они должны хранить в себе знания профессионалов-экспертов в некоторой предметной области. И не просто хранить, но и передавать их тем, у кого таких знаний в данной предметной области нет. Для этого в ЭС предусмотрены не только простые средства общения между системой и специалистом, но и средства доведения хранимых знаний до специалиста вместе с необходимыми пояснениями.

В любой момент времени в ЭС существуют три типа знаний:

- структурированные знания – статические знания о предметной области. После того как эти знания выявлены и проверены, они уже не изменяются;
- структурированные динамические знания – изменяемые знания о предметной области, они обновляются по мере выявления новых знаний;
- рабочие знания – знания, применяемые для решения конкретной задачи или проведения консультации. Это те знания, которые в данный момент известны о предметной области и вводятся в систему для решения задачи. Все знания хранятся в базе знаний ЭС.

На практике ЭС используются, прежде всего, как системы-советчики в тех ситуациях, где специалист сомневается в выборе правильного решения. Экспертные знания, хранящиеся в памяти системы, более глубокие и полные, чем соответствующие знания пользователя. Иногда бывает наоборот. Например,

при работе в системах оперативного управления знания, хранящиеся в базе знаний экспертной системы, могут быть и меньшими, чем у специалиста, работающего в паре с системой. Но быстрота и точность реакции системы значительно выше, чем у человека.

Часто к экспертным системам относят АСНИ (автоматизированные системы научных исследований). В отличие от классических ЭС, они рассчитаны на экспертов-профессионалов. Для таких специалистов нужна не консультирующая или советующая система, а система, способная помочь им в научной работе, в получении новых знаний. На основании частных знаний эксперта эти системы способны обнаружить в большом эмпирическом материале опытов и исследований скрытые связи и закономерности, которые эксперт может и не заметить.

Структура и режимы использования ЭС. Типичная экспертная система состоит из следующих основных компонентов: механизма вывода или решателя (интерпретатора), рабочей памяти или базы данных, базы знаний, компонентов приобретения знаний, объяснительного и диалогового (рис. 17).

База данных предназначена для хранения исходных и промежуточных данных (рабочее множество) решаемой в текущий момент задачи. Этот термин совпадает по названию, но не по смыслу с термином, используемым в системах управления базами данных для обозначения всех данных, хранимых в системе.

База знаний предназначена для хранения долгосрочных данных, описывающих рассматриваемую предметную область (а не текущих данных), и правил, манипулирующих этими данными, описывающих целесообразные преоб-



Рис. 17. Типовой состав экспертной системы

разования данных этой области. База знаний – наиболее важная компонента ЭС, на которой основаны ее «интеллектуальные способности».

Знания в ЭС организованы таким образом, чтобы знания о предметной области отделить от других типов знаний системы, таких, как общие знания о том, как решать задачи, или знания о том, как взаимодействовать с пользователем, например как печатать текст на терминале пользователя или как изменить текст в соответствии с командами пользователя. Выделенные знания о пред-

метной области составляют *базу знаний*, а общие знания о нахождении решений задач называются *механизмом вывода*. База знаний ЭС содержит факты (данные) и правила (или другие представления знаний), использующие эти факты как основу для принятия решений. Механизм вывода содержит *интерпретатор*, определяющий, каким образом применять правила для вывода новых знаний, и *диспетчер*, устанавливающий порядок применения этих правил.

Существует несколько способов представления знаний в ЭС, однако общим для всех них является то, что знания представлены в *символьной форме* – наборами символов, соответствующих понятиям предметной области. На жаргоне искусственного интеллекта *символ* – это строка знаков, соответствующая содержанию некоторого понятия реального мира. В ЭС процесс рассуждения представляется как последовательность символьных преобразований. Наиболее распространен способ представления знаний в виде конкретных *фактов и правил*, по которым из имеющихся фактов могут быть выведены новые. *Правила* в базе знаний имеют обычно вид: *ЕСЛИ А ТО S*, где *A* – *условие*, а *S* – *действие*. Действие *S* выполняется, если *A* истинно. Наиболее часто действие *S* так же, как и условие, представляет собой утверждение, которое может быть выведено самой системой (становится ей известным), если истинно условие правила *A*.

Например, ЕСЛИ: 1) окраска бактерий грамположительная; 2) морфология бактерий характерна для кокков; 3) форма колоний – цепочки ТО: есть основания считать (70 %), что вид бактерий стрептококк.

Когда текущая ситуация (факты) в задаче удовлетворяет или согласуется с частью правила *ЕСЛИ*, то выполняется действие, определяемое частью *ТО*. Это действие может оказаться воздействием на окружающий мир, или же повлиять на управление программой (например, вызвать проверку и запуск некоторого набора правил), или может сводиться к указанию системе о получении определенного заключения (например, добавить новый факт или гипотезу в базу данных).

Компонент приобретения знаний предназначен для добавления в базу знаний новых правил и модификации имеющихся. В его задачу входит приведение правила к виду, позволяющему механизму вывода применять это правило в процессе работы. В более сложных системах предусмотрены еще и средства для проверки вводимых или модифицируемых правил на непротиворечивость с имеющимися правилами.

Механизм логического вывода или Решатель, используя исходные данные из рабочей памяти и знания из базы знаний, формирует такую последовательность правил, которые, будучи примененными к исходным данным, позволяют делать те или иные выводы, что и приводят к решению задачи. Это программная компонента ЭС, реализующая процесс ее рассуждений на основе базы знаний и рабочего множества, поступающего в систему (в базу данных). Она выполняет две функции: во-первых, просмотр существующих фактов из рабочего множества и правил из базы знаний и добавление (по мере возможности) в рабочее множество новых фактов и, во-вторых, определение порядка просмотра и применения правил. Эта подсистема управляет процессом консультации, со-

храняет для пользователя информацию о полученных заключениях, и запрашивает у него информацию, когда для срабатывания очередного правила в рабочем множестве оказывается недостаточно данных.

Работа системы представляет собой последовательность шагов, на каждом из которых из базы выбирается некоторое правило, которое применяется к текущему содержимому рабочего множества. Цикл заканчивается, когда выведено либо опровергнуто целевое утверждение. Цикл работы ЭС иначе называется *логическим выводом*. Логический вывод может происходить многими способами, которые мы рассматривать подробно не будем.

Объяснительный компонент объясняет, как система получила решение задачи (или почему она не смогла получить решения) и какие знания она при этом использовала, что облегчает эксперту тестирование системы и повышает доверие пользователя к полученному результату. Большинство этих объяснений включают демонстрацию цепочек выводов и доводов, объясняющих, на каком основании было применено каждое правило в цепочке. Возможность проверять собственные процессы рассуждения и объяснять свои действия – это одно из самых новаторских и важных свойств ЭС.

В ЭС, база знаний которых насчитывает сотни и тысячи правил, весьма желательным является использование какой-либо стратегии управления выводом, позволяющей минимизировать время поиска решения и тем самым повысить эффективность вывода. Таких стратегий в настоящее время существует несколько, например поиск в глубину, поиск в ширину, разбиение на подзадачи, методы перебора и альфа-бета алгоритм.

Диалоговый компонент ориентирован на организацию дружелюбного общения со всеми категориями пользователей, как в ходе решения задач, так и в ходе приобретения знаний, а также объяснения результатов работы (вроде пользовательского интерфейса в операционных системах).

В разработке ЭС участвуют представители следующих специальностей:

- эксперт в той проблемной области, задачи которой будет решать ЭС;
- инженер по знаниям – специалист по разработке ЭС;
- программист – специалист по разработке инструментальных средств (программ).

Необходимо отметить, что отсутствие среди участников разработки инженера по знаниям или замена его программистом приводит к неудаче процесс создания ЭС, либо значительно удлиняет его. Эксперт определяет знания (факты, данные и правила манипулирования ими), характеризующие проблемную область, обеспечивает полноту и правильность введения в ЭС знаний.

Инженер по знаниям помогает эксперту выявлять и структурировать знания, необходимые для работы ЭС, осуществляет выбор того инструментального средства, которое наиболее подходит для данной предметной области, и определяет способ представления знаний в этом инструментальном средстве. Он выделяет и программирует (традиционными средствами) стандартные функции, типичные для данной предметной области, которые будут использоваться в правилах, вводимых экспертом.

Программист разрабатывает инструментальное средство, содержащее в пределе все основные компоненты ЭС, осуществляет сопряжение инструментального средства с той средой, в которой оно будет использовано.

Режимы работы ЭС. Экспертная система работает в двух основных режимах: приобретения знаний и решения задач (называемом также режимом консультации, или режимом использования ЭС).

В режиме приобретения знаний общение с ЭС осуществляется через посредничество инженера по знаниям – специалиста по разработке ЭС. Эксперт описывает проблемную область в виде совокупности данных и правил. Данные определяют объекты, их характеристики и значения, существующие в области экспертизы. Правила определяют способы манипулирования данными, характерные для рассматриваемой проблемной области. Эксперт, используя компонент приобретения знаний, наполняет систему знаниями, которые позволяют ЭС в режиме решения самостоятельно (без эксперта) решать задачи из указанной проблемной области.

Следует заметить, что режиму приобретения знаний при традиционном подходе к разработке программ соответствуют этапы алгоритмизации, программирования и отладки программ, выполняемые программистом. В отличие от традиционного подхода, разработку программ в экспертной системе осуществляет эксперт (с помощью ЭС), не владеющий программированием.

Важную роль в режиме приобретения знаний играет объяснительный компонент. Именно благодаря ему эксперт на этапе тестирования локализует причины неудачной работы ЭС, что позволяет эксперту целенаправленно модифицировать старые или вводить новые знания. Обычно объяснительный компонент сообщает следующее:

- насколько правильно используют информацию пользователя;
- почему использовались или не использовались определенные данные или правила;
- какие были сделаны выводы и т. д.

Все объяснения делаются, как правило, на ограниченном естественном языке или языке графики.

В режиме консультации общение с ЭС осуществляет конечный пользователь, которого интересует результат и (или) способ получения решения. Пользователь в зависимости от назначения ЭС может не быть специалистом в данной предметной области, в этом случае он обращается к ЭС за советом, или быть специалистом, в этом случае он обращается к ЭС, чтобы либо ускорить процесс получения результата, либо возложить на систему рутинную работу.

В режиме консультации данные о задаче пользователя обрабатываются диалоговым компонентом, который выполняет следующие действия:

- распределяет роли участников (пользователя и системы) и организует их взаимодействие в процессе кооперативного решения задачи;
- преобразует данные пользователя о задаче, представленные на привычном для пользователя языке, на внутренний язык системы;

- преобразует сообщения системы, представленные на внутреннем языке, в сообщения на языке, привычном для пользователя (ограниченный естественный язык или язык графики).

После обработки данные поступают в рабочую память. На основе входных данных в рабочей памяти, общих данных о проблемной области и правил из базы знаний решатель (интерпретатор) формирует решение задачи.

В отличие от традиционных программ ЭС в режиме решения задачи не только исполняет предписанную последовательность операций, но и предварительно формирует ее. Если ответ ЭС не понятен пользователю, то он может потребовать объяснения, как ответ получен.

Исследования и разработки ЭС составляют основу программ информатизации развитых государств. Несмотря на значительную стоимость разработки, ЭС приносят существенный доход. Например, ежегодный доход от использования ЭС *DEFT* (проверка гигантских дисководов для ЭВМ фирмы *IBM*) составляет \$12 млн. (начальная стоимость системы \$100 тыс.), а от ЭС *XCON* (помогает продавцам компьютерных систем *VAX* фирмы *DEC* определять конфигурации систем на заказ) – \$25 млн. при стоимости разработки ЭС около \$3 млн.

Области применения ЭС и типы задач, решаемых ими. Перечислим некоторые из предметных областей человеческой деятельности, в которых ЭС применяются в настоящее время: военное дело, геология, информатика, космическая техника, инженерное дело, компьютерные системы, медицина, метеорология, управление процессами, промышленность и сельское хозяйство, математика, химия, физика, электроника, системы управления ядерными реакторами и многие другие.

Из них *медицина* представляется наиболее популярной. Именно в медицине было разработано ЭС больше, чем во всякой другой области, хотя химия не намного отстает от нее, и разрыв быстро сокращается. В настоящее время ведутся разработки медицинских ЭС, осуществляющих интерпретацию лабораторных анализов, диагностику заболеваний, рекомендуя способы лечения и обучающих диагностике и лечению заболеваний, а также методам анестезии.

В области *компьютерных систем* современные разработки ЭС связаны с диагностикой неисправностей, проектированием компьютерных конфигураций и управлением процессом производства компьютеров.

В *электронике* преобладают исследования и разработки, связанные с диагностикой неисправностей и проектированием сверхбольших интегральных схем (на кристалле микропроцессора величиной чуть больше конторской скрепки располагается более 40 млн. транзисторов и других радиоэлементов, объединенных в сложнейшие функциональные схемы). В настоящее время работы по ЭС в области электроники связаны с обучающими системами, помогающими находить отказы в электрических цепях и проектировать электронно-цифровые схемы, вести диагностику неисправностей в радиоэлектронной и бытовой технике и т. д. Применение ЭС позволяет получать поразительные результаты в росте производительности труда и скорости выполнения отдельных операций, например:

- по данным фирмы *NEC* при проектировании интегральных микросхем удалось повысить производительность труда в 3–6 раз, при этом выполнение некоторых операций ускоряется в 10–15 раз;
- ускорить поиск неисправностей в сложных устройствах в 5–10 раз;
- повысить производительность труда программистов по данным фирмы *Toshiba* в 5 раз;
- при профессиональной подготовке без потерь качества сократить в 8–12 раз затраты на индивидуальную работу с обучаемым и т. д.

Экспертные системы создаются для решения разного рода проблем, но основные типы их деятельности можно сгруппировать в следующие категории: *прогноз, диагностика, проектирование, интерпретация, планирование, наблюдение, отладка, ремонт, обучение, управление.*

Экспертные системы, выполняющие *интерпретацию*, как правило, используют информацию от датчиков для описания ситуации. Они могут обрабатывать разнообразные виды информации. Интерпретирующие ЭС имеют дело не с четкими символьными представлениями проблемной ситуации, а непосредственно с реальными данными, например, показаниями измерительных приборов на химическом заводе для определения состояния технологического процесса, или используют данные дифракции рентгеновских лучей, спектрального анализа или ядерно-магнитного резонанса для вывода химической структуры веществ. Они сталкиваются с затруднениями, которых нет у систем других типов, потому что им приходится обрабатывать информацию зашумленную, недостаточную, неполную, ненадежную или ошибочную. Им необходимы специальные методы регистрации характеристик непрерывных потоков данных, сигналов или изображений и методы их символьного представления.

Экспертные системы, осуществляющие *прогноз*, определяют вероятные последствия заданных ситуаций. Примерами служат оценка спроса на нефть на мировом рынке в зависимости от складывающейся геополитической ситуации, прогнозирование места возникновения следующего военного конфликта на основании данных разведки, количество продаж автомобилей определенного класса, или определение процента инфляции в государстве за год и т. д.

Экспертные системы выполняют *диагностирование*, используя описания ситуаций, характеристики поведения или знания о конструкции компонент, чтобы установить вероятные причины неправильного функционирования диагностируемой системы, например, определение причин заболевания по симптомам, наблюдаемым у пациентов, или определение неисправных компонент в системе охлаждения ядерных реакторов.

Очень кратко рассмотренные основные виды деятельности ЭС и области, в которых они решают проблемы, взятые в совокупности, позволяют увидеть широту диапазона применения, многообразие информации и разнообразие форм представления знаний, присущие существующим экспертным системам.

Проектирование и создание ЭС. На проектирование и создание одной ЭС раньше требовалось 25 – 30 человеко-лет. В настоящее время имеется инструментарий, позволяющий сократить это время в 3 – 5 раз. В широком понимании в инструментарий входят аппаратные средства, языки программирования и

программное обеспечение. По степени отработанности инструментальных средств выделяют три стадии существования: экспериментальная, исследовательская и коммерческая. Высшей стадией существования инструментального средства является коммерческая. Эти инструментальные средства всесторонне и тщательно проверены, хорошо документированы, сопровождаются разработчиком, являются быстрыми и обладают удобным интерфейсом с пользователем. Экспертные системы создаются на ЭВМ общего назначения, персональных компьютерах, интеллектуальных рабочих станциях типа San, Apollo и других, снабженных эффективными инструментальными средствами для создания ЭС. Используются последовательные символьные ЭВМ типа ЛИСП-машин (*Symbolik-3670, Alpha, Xerox 1100* и др.) и ПРОЛОГ-машин, а также параллельные символьные ЭВМ (*Connection, Dado, Faun, Hyper Cube* и др.). По типу инструментальные средства подразделяются:

- на языки инженерии знаний, т. е. языки высокого уровня, ориентированные на построение экспертных систем, например, *OPS-5, LOOPS, KES, ПРОЛОГ* и др.;
- на символьные языки программирования, ориентированные на построение экспертных систем и систем искусственного интеллекта, например, *LISP, INTERLISP, SMALLTALK*;
- системы, автоматизирующие разработку (проектирование) экспертных систем, например, *KEE, ART, TEIRESIAS, AGE, TIMM* и др. Их часто называют окружением (environment) для разработки систем искусственного интеллекта, ориентированных на знания;
- оболочки ЭС (или пустые ЭС) – экспертные системы, не содержащие знаний ни о какой проблемной области, например, *ЭКСПЕРТИЗА, ЭКО, ЕМУСИН, ЭКСПЕРТ* и др.

В приведенной классификации инструментальные средства перечислены в порядке убывания трудозатрат на создание с их помощью конкретной ЭС. Развитие систем, автоматизирующих разработку ЭС, приводит к появлению инструментальных средств, которые можно назвать настраиваемыми оболочками. Они позволяют разработчику использовать оболочку не как нечто неизменное, а генерировать ее из множества механизмов, имеющихся в инструментальных средствах. Типичными таймерами таких инструментальных средств являются *KEE, ART, ЭКСПЕРТИЗА, ГЛОБ*.

В настоящее время, как правило, инструментальные средства для создания простых ЭС разрабатываются на персональных компьютерах, а инструментальные средства для создания сложных ЭС – на символьных ЭВМ, ЭВМ общего назначения и интеллектуальных рабочих станциях.

Преимущества ЭС перед человеком-экспертом.

Экспертные системы имеют следующие преимущества перед экспертом-профессионалом:

1. У них нет предубеждений.
2. Они не делают поспешных выводов.
3. Системы работают систематизировано, рассматривая все ситуации и детали, часто выбирая наилучшую альтернативу из всех возможных.

4. База знаний может быть очень и очень большой. Будучи введены в машину один раз, знания сохраняются навсегда. Человек же имеет ограниченную базу знаний, и если данные долгое время не используются, то они забываются и теряются навсегда.

5. Системы, основанные на знаниях, устойчивы к «помехам». Человек-эксперт легко поддается влиянию внешних факторов и пользуется побочными знаниями, часто не связанными с решаемой задачей. ЭС, не обремененные знаниями из других областей, по своей природе менее подвержены «шумам».

6. Эти системы не заменяют специалиста-эксперта, а являются инструментом в его руках.

Ограничения в применении экспертных систем. Даже лучшие из существующих ЭС, эффективно функционирующих как на больших, так и на мини-ЭВМ, имеют определенные ограничения по сравнению с человеком-экспертом.

1. Навыки системы не возрастают после сеанса экспертизы по сравнению с человеком.

2. Если пользователь не имеет некоторого опыта работы с такими системами, то у него могут возникнуть серьезные трудности. Многие системы оказываются доступными только тем экспертам, которые создавали их базы знаний.

3. ЭС не способны обучаться, не обладают здравым смыслом. Даже ребенок в состоянии легко уяснить, что он станет мокрым, если опрокинет на себя стакан с водой, однако если начать выливать из чашки кофе на клавиатуру, у системы не хватит «ума» отодвинуть ее.

4. ЭС мало применимы в больших предметных областях из-за большой задержки с выдачей экспертизы. Их использование ограничивается предметными областями, в которых решение может быть выдано за приемлемые сроки.

5. В тех областях, где отсутствуют эксперты-профессионалы, применение ЭС оказывается невозможным (например, в астрологии) – их просто некому наполнять знаниями и правилами манипулирования этими знаниями.

Классификация компьютеров

Люди с древнейших времен пытались изобрести себе инструменты, облегчающие производить различные вычисления.[♦] Однако только благодаря трем основным техническим достижениям стало возможным появление компьютера:

- изобретение в 1918 г. электронного реле (триггера) – простейшей быстродействующей схемы, замыкающей и размыкающей электрическую цепь;
- разработка цифрового кодирования любой информации;
- создание устройств искусственной памяти, позволяющей хранить программы и данные не на внешних носителях, а в самом компьютере, и выполнять эти программы автоматически.

Мы уже упоминали о том, что любая информация в компьютере кодируется в виде двоичных чисел. Двоичные данные по своей сути характеризуются двумя состояниями – «включено» и «выключено», поэтому главным требованием к устройствам их обработки является качество переключателей.

С технической точки зрения компьютер и представляет собой, в сущности, набор переключателей для представления информации в виде битов и управления процессами ее обработки. Как только включается питание компьютера и начинают вырабатываться импульсы тока, его процессор начинает выполнение непрерывающейся последовательности машинных команд различных программ (служебных, системных, пользовательских и т. д.), заключающееся в бесконечном переключении схем из одного состояния в другое. Выполняемые программы чередуются одна за другой, но процессор этого не замечает. Он не видит разницы между командами одной программы и командами другой программы, так как и те и другие представляются в виде серий импульсов. Что при этом выполняет компьютер, – задают программы и пользователь.

С другой стороны, это сложное электронное устройство, которое как бы интегрировало (объединило) в себе практически все современные достижения в области хранения, обработки и передачи информации (телевизор, пишущая машинка, магнитофон, проигрыватель, телефон и т. д.) и самые новейшие технологии в области микроэлектроники. Именно компьютер «подгоняет» микроэлектронику развиваться такими бурными темпами.

С точки зрения кибернетики (науки об общих законах управления), компьютер – это программно-управляемый электронный автомат. Мы под термином компьютер будем понимать *управляемое программами электронное устройство, предназначенное для автоматизации сбора, хранения, обработки и передачи информации.*

[♦] Особую роль в развитии вычислительной техники сыграли работы в начале XIX в. выдающегося английского ученого **Чарльза Беббиджа**, но идеи Беббиджа намного опередили свое время. Уровень развития науки и техники того времени не позволил ему воплотить в жизнь свои замыслы.

Компьютер включает в себя два основных компонента:

- программное обеспечение (*Software*);
- аппаратная часть (*Hardware*).

Оба они равнозначны по важности, так как самостоятельно, без программ, компьютер ничего делать не может – это просто набор электронных устройств. В то же время без аппаратуры, реализующей программы пользователя, эти программы бесполезны и никому не нужны.

Компьютеры, как и информационные системы, классифицируют по различным признакам: по мощности, конструктивным и архитектурным особенностям, по организации и т. д. Рассмотрим классификацию по мощности.

1. *Большие ЭВМ, или мэйнфреймы (mainframe)*. Эти компьютеры создаются для обработки большого количества транзакций, т.е. больших объемов информации. Они легко масштабируются, надежны и к тому же поддерживают стандартные внешние интерфейсы. Мэйнфреймы представляются в виде мультипроцессорной структуры, построенной на базе проблемно-ориентированных и функционально-ориентированных специализированных процессоров.

К *проблемно-ориентированным* процессорам относятся различного типа обрабатывающие процессоры (матричные, символьные, графические и др.) и процессоры проблемно-ориентированных языков программирования, обеспечивающие эффективное выполнение программ, написанных на этих языках.

К *функционально-ориентированным* процессорам относятся процессоры ввода-вывода, телекоммуникационные, файловые процессоры, оптимизирующие обработку файлов, хранящихся во внешней памяти компьютера.

Все процессоры объединяются общей шиной, имеющей большую пропускную способность, работают с общей памятью и управляются общей операционной системой. Мэйнфреймы отличаются исключительной надежностью (за счет дублирования оборудования), высоким быстродействием, очень большой пропускной способностью каналов ввода-вывода. К ним могут подсоединяться тысячи терминалов или персональных компьютеров для работы пользователей. Например, 16-процессорный мэйнфрейм *IBM System z990* может обрабатывать до 12 тысяч транзакций (транзакция см. с.114) в секунду.

Большинство крупных корпораций, банков, правительственных учреждений во всем мире обрабатывают свои данные именно на больших ЭВМ. Например, все 25 ведущих мировых банков используют мэйнфреймы. Практически все вычислительные центры РЖД оснащены также мэйнфреймами. И хотя, благодаря фантастически возросшим возможностям, персональные компьютеры активно завоевывают себе «место под солнцем» и способны решать многие задачи, спрос на мэйнфреймы падает медленно, так как обеспечиваемые ими централизованные хранение и обработка данных обходятся дешевле, чем обслуживание распределенных систем обработки информации, состоящих из сотен и тысяч персональных компьютеров, объединенных в локальные вычислительные сети. Думается, что мэйнфреймы уж слишком долго существуют, чтобы могли быстро исчезнуть. Это подтверждают и цифры, например, в 2006 г. поставки мэйнфреймов фирмой *IBM* увеличились на 25 %.

К выпускавшимся ранее отечественным машинам такого типа относились ЭВМ *Единой Системы* – *ЕС ЭВМ* различной вычислительной мощности, начиная от *ЕС 1020* и кончая *ЕС 1067*. Было выпущено свыше 35 тысяч *ЕС ЭВМ*. После развала СССР до настоящего времени отечественные мэйнфреймы не выпускаются, приходится довольствоваться импортными машинами. К ним можно отнести такие, как комплексы японской фирмы *Hitachi* – *HDS/GX6215*, или мэйнфреймы фирмы *IBM* – *IBM S/390*, которые с начала этого столетия носят название *System z9*, в частности *z9BC* и *z9EC*.

Семейство *S/390*[®] фирмы *IBM*, – классическое семейство мэйнфреймов, появившееся более трех десятилетий назад. Архитектура *S/390* занимает доминирующее положение в мире, ей принадлежит более 70 % всей имеющейся мощности мэйнфреймов. Они построены по принципу открытой архитектуры, включают в себя расширенные сетевые возможности и операционные системы, более мощные и дешевые микропроцессоры. Их физические размеры уменьшились на 90 % по сравнению с предыдущими моделями, так как они построены по *CMOS*-технологии. *CMOS*-микросхемы охлаждаются воздушным потоком (водяное охлаждение не требуется), благодаря чему габариты «больших систем» уменьшились до размеров, казавшихся раньше просто фантастическими. Сегодня в одном мощном сервере *S/390 G5* может быть более 10 центральных процессоров, а занимает он менее 5 м². при массе около 900 кг. При этом он обрабатывает данные в 10 с лишним раз быстрее самого мощного мэйнфрейма предыдущего семейства и потребляет всего 4 % электроэнергии от той, которая требовалась предыдущим системам. Как правило, вместе с ним поставляется дисковая подсистема для хранения данных. В начале октября 2006 г. корпорация *IBM* вместе с очередным выпуском операционной системы *z/OS* для машин семейства *z9* объявила о новой инициативе, которая должна упростить работу на этих компьютерах. Дело в том, что большинство системных администраторов, обслуживающих сегодня *z9*, и программистов, создающих для них приложения, начинали работать в эпоху расцвета мэйнфреймов (до середины 1980-х г.), поэтому к середине следующего десятилетия эти специалисты уйдут на пенсию. Компаниям, где по-прежнему используются мэйнфреймы, пока трудно найти достойную замену ветеранам. Таким образом, хотя мэйнфреймы устояли под натиском открытых систем и персональных компьютеров, уже в следующем десятилетии они могут начать исчезать только из-за того, что их некому будет обслуживать.

Рассчитанная на пять лет инициатива, в которую *IBM* собирается вложить 100 млн. долларов, предусматривает выпуск инструментальных средств, облегчающих использование систем *z9*, модернизацию пользовательского интерфейса, а также более удобные утилиты для контроля расходов на лицензирование ПО и разработки приложений. Это позволит сократить сроки подготовки администраторов мэйнфреймов в несколько раз и привлечь их для работы с машинами *z9*. В России корпорация *IBM* с середины 2006 г. создала свое подразделение, вложив около 40 млн. долларов, которое уже занимается вопросами подготовки кадров и поддержкой мэйнфреймов с возможностью их выпуска в дальнейшем.

Произошедшие кардинальные изменения в архитектуре мэйнфреймов позволили существенно снизить эксплуатационные затраты на обработку информации. Так, например, в конце 90-х г. прошлого века в ИВЦ Свердловской железной дороги были установлены два комплекса на базе серверов *IBM 9672.R31* (трехпроцессорные серверы первого уровня), имеющих 512 Мбайт оперативной памяти, 9 каналов *BUS&TAG* (скорость в канале до 4,5 Мб/с.), 3 канала *ESCON* (скорость в канале до 19,7 Мб/с.) и быстродействием 45 млн. операций в секунду. Кроме того, были установлены еще два комплекса на базе машин *HDS/GX6215* фирмы *Hitachi*, имеющих 256 Мбайт оперативной памяти, 24 канала *BUS&TAG* и быстродействием 38 млн. операций в секунду. Это позволило заменить 12 мэйнфреймов типа *ЕС 1045* (*ЕС 1046*), высвободить около 800 м² площади машинных залов, сократить расход электроэнергии почти в 10 раз и высвободить десятки человек обслуживающего персонала. Изменения никак не повлияли на работу ИВЦ – все ранее решавшиеся централизованно и вновь разрабатываемые задачи (порядка 1300 в сутки) стали решаться на этих комплексах. Появилась также возможность организации высокоскоростного обмена комплексов с телекоммуникационной сетью дороги.

2. *СуперЭВМ. Эти компьютеры предназначены для решения задач, требующих громадных объемов вычислений.* Например, задача цифровой обработки аэрокосмических фотографий требует быстрого действия не менее чем 10¹⁴ опера-

ций в секунду. Колоссальных расчетов требуют моделирование ядерного взрыва в реальном времени, решение многомерных задач математической физики, моделирование физико-химических процессов, параллельные алгоритмы вычислительной математики, моделирование столкновений транспортных средств с неподвижными препятствиями и многие другие задачи.

Такие компьютеры, как правило, – многопроцессорные вычислительные комплексы[♦] с макроконвейерной организацией вычислений. Так, при сложении чисел с плавающей запятой один процессор выравнивает порядки чисел, другой – сдвигает мантиссы, третий – производит их сложение и т. д. – и все это одновременно, за счет чего быстродействие вычислений очень велико. Основными потребителями этих машин являются предприятия нефте- и газодобывающей промышленности, крупные транспортные и фармацевтические компании, службы прогнозирования погоды и контроля окружающей среды, силовые структуры, а также многие научные организации. К этим компьютерам, как и к предыдущим, могут быть подсоединены сотни терминалов и персональных компьютеров в качестве рабочих станций. Среди выпускавшихся ранее отечественных машин такого типа можно выделить, например, вычислительный комплекс «Эльбрус-1К2» с быстродействием 120 миллионов операций в секунду. Среди импортных суперЭВМ можно назвать, например, американский комплекс *GrayT3E-1200*, установленный в 2000 г. в одном из исследовательских центров Министерства обороны США. Комплекс представляет собой 1088 процессоров, общая производительность комплекса около 13 триллионов операций в секунду, объем оперативной памяти более 557 Гбайт.

Длительное время в России были прекращены выпуски суперЭВМ. Но вот в газете "Век" № 31 за 2001 г. появилось сообщение о создании и производстве суперЭВМ *MBC 1000M*. В последние годы появился так называемый кластерный подход к разработке многопроцессорных высокопроизводительных вычислительных систем. СуперЭВМ стали собирать из большого числа компьютеров, соединяя их «быстрыми схемами». *MBC 1000M* представляет собой 768 относительно небольших компьютеров, которые определенным образом соединены между собой. Прежде Россия была за пределами знаменитого списка TOP500, содержащего 500 самых мощных машин, установленных в мире. С появлением своих супер-ЭВМ мы поднялись на 25–30 место в мире и на 8–9 место в Европе. А если учесть, что европейские суперЭВМ завезены из США, то мы в Европе находимся на первом месте и входим в четверку сильнейших – вместе с США, Японией и Китаем.

Комплекс расположен на телекоммуникационном узле, что дает возможность ученым из многих городов России производить на нем крупномасштабные вычисления, которые не выполнить ни на каких персональных компьютерах.

На основании постановления Исполкома Союза России и Белоруссии № 43 от 22 ноября 1999 г. разработана совместная программа «СКИФ» по разработке и созданию суперЭВМ. В декабре 2004 г. Объединенный институт проблем информатики Национальной академии наук Белоруссии (ОИПИ НАН), Институт программных систем Российской академии наук (ИПС РАН) и компания «Т-Платформы» г. Москва объявили о создании суперкомпьютера СКИФ К-1000. Академик РАН, директор РНЦ «Курчатовский институт», академик-секретарь отделения информационных технологий и вычислительных систем РАН, член научного совета при Совете Безопасности РФ Евгений Велихов отмечает, что СКИФ К-1000 является наиболее мощной из всех вычислительных систем, установленных на территории Восточной Европы в соответствии с TOP500 за 2004 г.

Некоторые основные технические характеристики СКИФ К-1000:

пиковая производительность – 2,534 триллиона оп/с.;

[♦] По данным официального рейтинга Top500 выпущенная в ноябре 2007 г. суперЭВМ *eServer Blue Gene Solution* фирмы *IBM* имеет **212 992** процессора.

тип используемых процессоров – 2,2 ГГц FMD Opteron;
конструктив компьютера – 8 стоек по 36 узлов в стойке (288 2-х процессорных узлов);
оперативная память – 1152 Гб.;
дисковая память – 23040 Гб.;
суммарная длина кабельных соединений – свыше 2000 м.;
потребляемая мощность в режиме максимальной нагрузки – 89 квт.;
производительность системы охлаждения – 16 000 м³ в час.

Рынок супер-ЭВМ развивается наиболее динамично. Об этом можно судить, например, по такому факту: в 1995 – 96 годах японская супер-ЭВМ CP PACS с быстродействием 614 миллиардов операций в секунду заняла первое место в списке TOP500, а уже в 2003 г. выбыла из списка как самая медленная система. США поставили себе задачу утраивать производительность суперкомплексов каждые 18 месяцев. Комплекс работ, выполненных у нас при достаточном финансировании, позволяет нам ежегодно увеличивать производительность своих супер – ЭВМ в 10 раз. Но вся проблема в финансировании. Федеральный министр по связи и информатизации Леонид Рейман спустя три года после выпуска суперЭВМ *MBC 1000M* по телевидению заявил, что из-за отсутствия финансирования работы по совершенствованию этого комплекса приостановлены. В 2004 г. намечена новая программа «СКИФ-Грид» до 2008 г. За новую программу «проголосовали» Национальная академия наук Белоруссии, Министерство образования и науки России, Министерство экономического развития России. Однако Министерства финансов обоих государств деньги на ее реализацию пока не выделили. В то время как военное агентство передовых исследований США *DARPA* запланировало 494 млн. долларов корпорациям *IBM* и *Cray* для создания уже в следующем десятилетии суперкомпьютера с производительностью 10^{16} операций в секунду.

Иметь свои суперЭВМ крайне необходимо, так как США наложило эмбарго на вывоз компьютеров мощностью свыше 30 млрд. операций в секунду. Более того, по решению конгресса США за использованием тех компьютеров, которые вывозятся из страны, установлен жесточайший контроль. В свое время в США разразился даже скандал, когда кое-кому там показалось, что Минатом России приобрел технику для того, чтобы считать бомбу.

3. *Мини-ЭВМ*. Эти компьютеры занимают промежуточное положение между персональными компьютерами и мэйнфреймами. Используются они в учебных заведениях, центрах обработки информации, в достаточно крупных фирмах и других организациях. Очень часто эти ЭВМ используют в качестве серверов в компьютерных сетях. Обычно к мини-ЭВМ подключаются десятки и сотни терминалов или персональных компьютеров для работы пользователей. В свое время выпуск фирмой *DEC* большого количества мини-ЭВМ семейства *VAX* и распространение операционной системы *Unix* послужило одной из основополагающих причин возникновения Интернет. Основные производители этих машин - фирмы *IBM*, *DEC*, *Hewlett Packard*, *Silicon Graphics* и другие. Компьютеры фирмы *Silicon Graphics* снабжаются специальными аппаратными средствами для ускорения процессов трехмерного моделирования и анимации, поэтому на них создается большинство спецэффектов в выпускаемых сейчас кинофильмах и рекламных роликах. Отечественной промышленностью ранее выпускался ряд машин такого типа *Система микро-ЭВМ – СМ ЭВМ*, начиная с *СМ 1800* и заканчивая *СМ 1814*.

К сожалению, в России производство ЭВМ этих типов в настоящее время прекращено.

4. *Персональные компьютеры*. Появление специальных микросхем для обработки информации (микропроцессоров) обусловило создание совершенно нового типа компьютеров настольного или переносного (*Laptop* и *NoteBook*) исполнения. Эти компьютеры рассчитаны на использование в качестве индивидуального инструмента по обработке информации пользователями самого различ-

ного профиля. Их подразделяют на два больших класса: а) компьютеры фирмы IBM и совместимые с ними компьютеры других фирм-производителей, б) компьютеры типа *Macintosh* фирмы *Apple*, которые не получили такого широкого распространения, как IBM-совместимые. Так в 2006 г. было реализовано всего 4800 ПК. Фирма *Apple* со второго полугодия 2006 г. перешла на процессоры *Intel*, так что большой разницы между этими ПК, видимо, не будет. Переносные компьютеры оформлены в удобном для транспортировки виде (в виде небольших чемоданчиков – дипломатов) с автономным питанием и весом не более 4–5 кг. Обычно они состоят из системного блока, клавиатуры, плоского жидкокристаллического или плазменного монитора, энергонезависимого оперативного запоминающего устройства и средства сопряжения с внешними устройствами.

Персональные компьютеры являются наиболее широко используемым видом компьютеров, их возможности необычайно возрастают, стоимость постоянно снижается и, как следствие, области применения расширяются, охватывая практически все сферы человеческой деятельности. В настоящее время индустрия производства персональных компьютеров и программного обеспечения для них является одной из наиболее важных сфер экономики развитых стран, в том числе и России.

5. *Карманные персональные компьютеры (HANDHELD)*. Эти миниатюрные персональные цифровые ассистенты (Personal Digital Assistants-PDA) размером с блокнот (например, *HP95LX* размером 160x86x25 мм.) начинают теснить настольные персональные компьютеры. Как по размерам памяти, так и по быстродействию грань между ними с каждым днем уменьшается. Многие КПК позволяют работать в Интернете, смотреть видео или играть в игры и обмениваться данными с другими подобными или настольными ПК. В основном КПК поставляются с одной или двумя операционными системами, например с *Palm OS* или *Microsoft Pocket PC*. *Palm OS* имеет привлекательный интерфейс и работает с множеством разнообразных программ, которые можно купить или переписать бесплатно. *Microsoft Pocket PC* – это «калька» с *Windows*, что позволяет миллионам пользователей *Windows* чувствовать себя уверенно с первых мгновений работы с КПК. Многие программы для этой ОС представляют собой адаптированные версии программ для *Windows*, например *Word*, *Excel*, мультимедийные приложения и др. Основная часть процессоров работает на частотах 300 МГц и выше. Например, *HP ipaq HX4700* имеет процессор *Intel* с частотой 624 МГц. У большинства есть стереозвук, можно прослушивать аудиофайлы и использовать как цифровой диктофон. Можно выбрать КПК работающий еще и как мобильный телефон. Многие КПК, как и мобильные телефоны, имеют встроенные цифровые камеры, позволяют подсоединять и использовать любую цифровую аппаратуру.

Питаются КПК, как и мобильные телефоны, от литий-ионных аккумуляторов. Программы вводятся с твердой карточки *ROMCARD* 32, 64 или 128 Кб. КПК поставляются с каким-то стандартным объемом памяти для конкретных приложений, но его можно увеличивать, так как многие из них имеют разъемы для подключения карт памяти разных типов: *Compact Flash*, *SD*, *MMC*, *Memory Stick* и др. Стоимость карт памяти постоянно уменьшается. При выборе КПК

стоит обратить внимание на тип поддерживаемых им карт памяти, чтобы они были такими же, как у цифровой видеокамеры или других устройств, что позволило бы использовать одну и ту же карту памяти с разными устройствами. Некоторые КПК позволяют работать с несколькими типами карт памяти. Например, КПК *HP ipag zz1715* имеет разъемы *SD/MMC*, а КПК *Dell AXIMX5* позволяют работать с картами *Compact Flash, SD* и *MMC*.

В последнее время в связи с фантастически быстрым развитием микроэлектроники грани деления компьютеров на рассмотренные типы быстро стираются. Например, персональные компьютеры по своим вычислительным возможностям уже превосходят возможности малых машин, а мейнфреймы по физическим размерам, потребляемой мощности и другим параметрам стали просто миниатюрными. Отличаются они только по функциональным возможностям. Особенно это сближение проявляется в компьютерах, которые используются в качестве узлов в компьютерных сетях.

Из всех перечисленных типов компьютеров мы будем знакомиться только с IBM-совместимыми персональными компьютерами, которые наиболее широко распространены во всех сферах жизни общества.

Лекция 17

Основные принципы работы компьютера

В основу построения и работы практически всех современных компьютеров заложены общие фундаментальные принципы, сформулированные выдающимся американским математиком Джоном фон Нейманом, который взял за основу идеи Чарльза Беббиджа и развил их применительно к новым условиям. Формальная логическая организация компьютера была впервые представлена им в 1945 г. в «Предварительном докладе о машине EDVAC», одной из первых универсальных электронных вычислительных машин с хранимой программой.

Вот основные из принципов общего устройства компьютера:

1. Компьютер должен состоять из следующих основных устройств:
 - арифметико-логического устройства (АЛУ), выполняющего арифметические и логические операции над данными;
 - центрального устройства управления (ЦУУ), которое управляет логическими схемами и организует процесс автоматического выполнения программ;
 - оперативного запоминающего устройства (ОЗУ) для хранения данных и программ при выполнении их (пока включен компьютер);
 - внешних запоминающих устройств для хранения данных (ВЗУ);
 - устройств подготовки данных (занесения на машинные носители);
 - внешних устройств ввода – вывода информации (УВВ).
2. Компьютер должен быть не механическим, а электронным.
3. Компьютер должен оперировать с двоичными числами при параллельной организации вычислений.

4. Алгоритм представляется в форме последовательности *управляющих слов*, которые определяют смысл операции. Управляющие слова названы командами. Совокупность команд, представляющая алгоритм решения задачи, называется *программой*.

5. Программы и данные хранятся в одной и той же памяти. Команды от данных отличаются месторасположением в памяти, но не способом кодирования.

Такая организация получила название *классической архитектуры фон Неймана* и приведена на рис. 18.

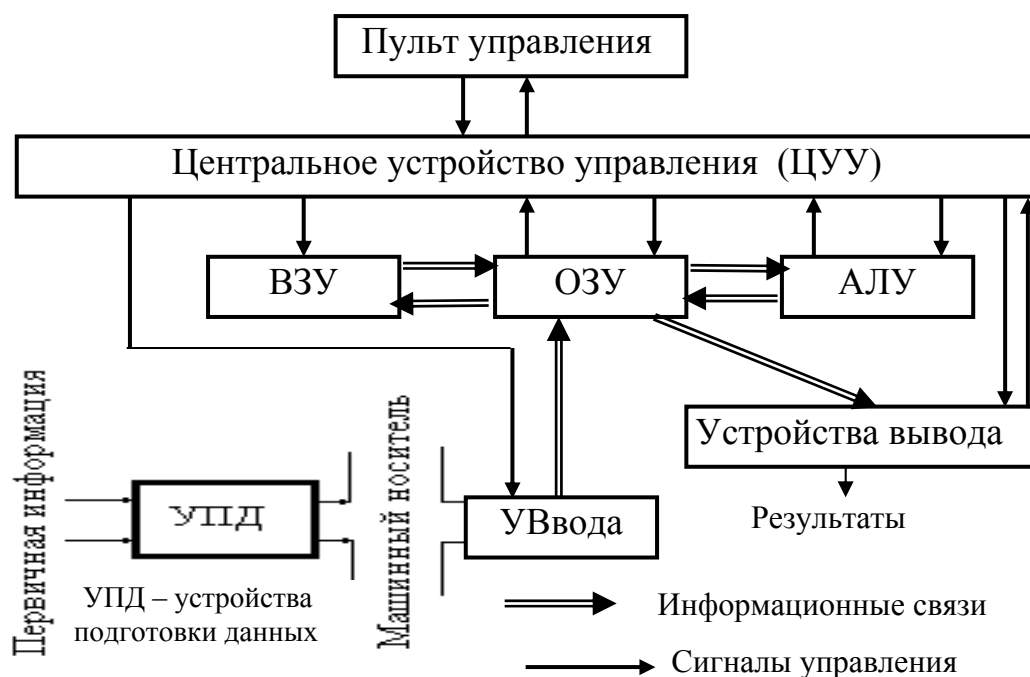


Рис. 18. Классическая архитектура ЭВМ по фон Нейману

Кратко рассмотрим некоторые важные принципы работы компьютера. В отношении ОЗУ заложен **принцип произвольного или прямого доступа**. Это значит, что ОЗУ состоит из множества пронумерованных ячеек, в которых может храниться закодированная в двоичном коде любая информация (цифровые данные, текст, графические изображения и т. д.). Любому устройству компьютера в любой произвольный момент времени доступна любая ячейка по ее адресу (номеру) для записи или чтения. Время доступа для всех ячеек одинаково, независимо от того, обращаемся ли мы к первой от начала или последней ячейке памяти. Оперативная память энергозависима. При отключении питания информация в ней не сохраняется.

Поскольку каждая команда в компьютере представляется в двоичном коде, программа при выполнении может храниться в его памяти, как любая другая информация. В этом состоит **принцип хранения программы** компьютера в самом компьютере (аналогично мозгу человека).

Принцип программного управления компьютером реализуется следующим образом. При запуске программы на исполнение операционная система за-

гружает ее в оперативную память и передает в *программный счетчик* устройства управления адрес первой машинной команды программы. Команда может быть любой (выполнение арифметической или логической операции, чтение из памяти данных, ввод данных из внешнего устройства в память или вывод данных из памяти на внешнее устройство и т. д.). По заданному адресу устройство управления считывает команду *в регистр** команд (там она хранится во время выполнения) и организует ее выполнение. *Код команды* расшифровывается и обрабатывается АЛУ путем выработки устройством управления последовательности управляющих импульсов, воздействующих на АЛУ и другие устройства. Схемы АЛУ переключаются в соответствии с ними и выполняют то, что определено командой. *Адресная часть команды* заносится в регистры адреса, откуда передается на шину адресов для пересылки данных в соответствии с выполняемой командой. Устройство управления по коду команды определяет ее длину, и после выполнения первой команды к заданному первоначальному адресу прибавляет длину в байтах выполненной команды, получая новый адрес, после чего считывает из памяти по этому адресу следующую команду программы. Поэтому в ячейках памяти, следующих за расположением первой команды, должна находиться именно вторая команда, а не что-то другое. Вот почему, в отличие от расположения данных, которые адресуются в командах, **программа должна располагаться в оперативной памяти непрерывным участком**. Иначе устройство управления не найдет следующую команду для выполнения. Затем устройство управления организует выполнение второй команды, определяет адрес следующей и т. д. Таким образом, устройство управления выполняет программу автоматически без вмешательства человека, заставляя бесконечно переключаться схемы АЛУ из одного состояния в другое.

Устройство управления вырабатывает серии управляющих импульсов, реализуя еще один важнейший принцип, вытекающий из принципа программного управления фон Неймана и являющийся его развитием – **принцип микропрограммного управления**. Выполнение любой машинной операции, например, сложения, выливается в довольно емкую последовательность элементарных действий: сдвигов влево или вправо при выравнивании порядков чисел, проверка условий выравнивания порядков, анализа цепочек переносов из одного разряда в другой при сложении мантисс и т. д. Такая последовательность действий управляется сериями импульсов (*микрооперациями*), вырабатываемых устройством управления и воздействующих на схемы АЛУ. Для выполнения каждой такой микрооперации оказывается возможным задать простейший набор импульсов (*микрокоманду*). **Совокупности таких микрокоманд образуют своего рода программы для выполнения той или иной команды процессора**. Эти программы хранятся в постоянном запоминающем устройстве (ПЗУ) процессора. Устройство управления, функции которого заключаются в том, чтобы считать очередную команду программы из памяти, расшифровать ее и подклю-

* Регистрами называются устройства, выполняющие функции приема, хранения и передачи информации. Каждому двоичному разряду числа, записанного в регистр, соответствует свой разряд регистра, выполненный в виде триггера определенного типа.

читать необходимые электронные схемы и цепи АЛУ для ее выполнения, после считывания команды, обращается к ПЗУ и вызывает необходимую программу из микроопераций, которая задает распределение серий импульсов по элементам АЛУ и другим устройствам, и выполняет команду процессора.

Таким образом, практически все команды машины реализуются путем выполнения определенного набора микропрограмм. Такой подход позволяет, во-первых, сравнительно легко расширять системы команд компьютера. Команды компьютеров становятся все более сложными, и схемными методами распределения импульсов их уже не реализовать. Во-вторых, сравнительно просто решается задача совместимости компьютеров путем программирования команд других машин из имеющегося набора микрокоманд данной машины. Выполняется все это на фирме-изготовителе при создании процессора.

Порядок выполнения процессором последовательности команд может быть изменен, если в программе встречается команда условного или безусловного перехода на другой ее участок. В этом случае для устройства управления в самой команде перехода указывается адрес следующей команды, которую оно должно считывать для выполнения. Поскольку внешние устройства работают значительно медленнее, чем АЛУ и память, устройство управления может приостанавливать выполнение программы до завершения операции ввода–вывода с внешним устройством, которая реализуется специальной машинной программой, состоящей из команд устройства.

Устройство управления совместно с операционной системой обеспечивает еще один из принципов работы компьютера – *многозадачность* – способ организации работы компьютера, при котором в его памяти одновременно находятся программы и данные для решения нескольких задач. Для этого имеются специальные аппаратно-программные средства *диспетчеризации*, в том числе системы *прерываний* и *защиты памяти*.

Суть *прерываний* заключается в следующем. Процессор выполняет программу, пока не возникает событие, требующее ее приостановления и выполнения других действий (например, прием вводимых с клавиатуры или с диска данных, требование от системы диспетчеризации на выполнение другой программы, запрос от внутреннего таймера на обновление текущего времени и т.д.). Процессору *выставляется запрос на прерывание*, т. е. подается специальная команда *INT n*, где *n* – номер прерывания. Процессор на некоторое время прекращает выполнение текущей программы и занимается обслуживанием прерывания. Так как любые действия процессор может выполнять только по программе, то и обслуживание прерывания он тоже осуществляет путем выполнения специальной программы – *обработчика прерывания*. Каждое событие или устройство, требующие внимания процессора, имеют свой номер прерывания и соответственно программу-обработчик. Данные из регистров процессора (адрес выполняемой команды, ее код и операнды, значения флагов и др.) перемещаются в специально организованный участок памяти – стек командами процессора *PUSH регистр*. В освободившиеся регистры процессора заносится первая команда программы-обработчика прерывания и процессор, не замечая, что это новая программа, выполняет уже ее. Заканчивается выполнение программы-

обработчика прерывания несколькими командами процессора *POP регистр*, извлекающими из стека в его регистры запомненную там информацию, после чего по команде *IRET* процессор возвращается из прерывания и снова продолжает выполнять ранее прерванную программу.

Сама программа-обработчик тоже может быть прервана, как и любая другая программа, новым прерыванием, т. е. прерывания могут многократно «накладываться» одно на другое. В этом случае чаще всего прерывания «маскируются» (на них накладывается маска – ставится специальный признак). Они не обслуживаются немедленно, а становятся в очередь и обрабатываются поочередно, одно за другим. Есть прерывания, которые не маскируются и не имеют на это права. Например, прерывание от источника питания должно немедленно обрабатываться вне всякой очереди.

Каждый номер прерывания имеет указатель адреса начала его программы-обработчика. Указатели называются *векторами прерываний*. Они состоят из 4-х байт каждый и хранятся в начале оперативной памяти. В байтах вектора прерывания и указывается адрес первой команды программы-обработчика этого прерывания.

Следует подчеркнуть, что большинство векторов прерываний содержат именно адреса программ-обработчиков. Вместе с тем некоторые адреса указывают не на программы-обработчики, а на специальные таблицы с информацией. Так, вектор *1Fh* указывает на таблицу – знакогенератор, содержащую битовые комбинации (шаблоны символов), используемые *ROM BIOS* при выводе на экран монитора символов текста, а прерывания *41h* и *46h* указывают на таблицу с параметрами жесткого диска (винчестера). Такие вектора прерываний используются для удобства доступа к нужной информации, а не для выполнения программы-обработчика.

Прерывания программируются на любом языке программирования высокого уровня. Управление прерываниями осуществляет специальная микросхема – контроллер прерываний.

Идея прерываний была выдвинута и реализована для взаимодействия процессора с внешним миром. Однако оказалось, что прерывания весьма полезны и для внутренней работы компьютера. Это привело к разработке трех типов прерываний, которые получили названия *аппаратных, логических и программных*. Между ними принципиальной разницы нет, но источники у них разные.

Источники *аппаратных* прерываний – аппаратные средства. Например, запрос на прерывание выставляется при падении напряжения источника питания, нажатии клавиши на клавиатуре, приходе очередного импульса от счетчика времени, возникновении специального сигнала от накопителей на гибких и жестких магнитных дисках и др. Каждому устройству на системной шине выделена специальная отдельная линия запроса прерывания (не путать с номером прерывания). Например, с номером 06 – прерывание от контроллера гибких дисков, с номером 1 – прерывание от контроллера клавиатуры, с номером 4 – прерывание от последовательного порта *COM1* и т. д. Всего, например, на системной шине *PCI* имеется 15 линий прерываний. Когда устройство хочет «пообщаться» с

процессором, оно по своей линии запроса прерывания выдает сигнал контроллеру прерываний, а он формирует команду *INT n* и посылает ее процессору.

Логические прерывания формируются самим процессором, когда он встречает какое-либо необычное для него условие, например, деление на нуль, шаговый режим работы, переполнение регистров при выполнении арифметических операций и т. д. В многозадачных операционных системах типа Windows процессор может генерировать прерывание при обращении к странице памяти, которая в данный момент отсутствует в основной памяти (прерывание отсутствия).

Наиболее интересная и самая обширная категория – *программные* прерывания. Они вырабатываются, когда одна программа хочет получить определенный сервис со стороны другой программы. Обычно это запросы программ к операционной системе для получения какого-либо сервиса, например ввода данных с диска или вывода данных на экран. Прерывания с номерами 0–31 обслуживаются программами базовой системы ввода-вывода *BIOS*. Они называются прерываниями *нижнего уровня*. Эти прерывания управляют непосредственно аппаратными средствами компьютера и недоступны пользователю. Обслуживание прерываний с номерами свыше 32 возлагается на операционную систему.

Почти каждая процедура, обслуживающая прерывание с заданным номером, выполняет несколько различных операций. Поэтому программа-обработчик содержит несколько ссылок на вторичные программы прерывания (как бы подпрограммы). Иногда их называют функциями (службами) и внутри каждого прерывания различают по номеру, т. е. каждой функции, как и прерыванию, присваивается свой номер. Так, например, прерывание с номером 11 (управление видеосистемой) включает в себя более 60 различных функций и подфункций для поддержания того или иного режима работы экрана (текстовый, графический, с определенным разрешением и количеством цветов и т. д.). Выполнение функций осуществляется так: в один из регистров (чаще всего *AH*) процессора заносится номер функции и вызывается содержащее ее прерывание.

Аппаратные и логические прерывания не зависят от типа операционной системы и существуют во всех компьютерах. В операционных системах Windows программные прерывания как таковые отсутствуют, хотя сам принцип работы процессора остается неизменным. Процессор в силу мультизадачности этих систем постоянно работает в режиме прерываний (одновременно работает несколько программ, но процессор-то только один!). Такой режим работы обеспечивает метод диспетчеризации, предоставляющий операционной системе возможность захвата *CPU* в любой момент времени, независимо от состояния работающих в данный момент приложений. Это позволяет Windows время работы процессора предоставлять приложениям поочередно, с перерывами, которые из-за большого быстродействия процессора пользователь не замечает.

Приложения же получают любой сервис от Windows при помощи интерфейса прикладного программирования API Win32, который содержит более 2000 различных функций и макросов. Доступ ко всем функциям API Win 32 осуществляется по имени, а не по номерам, как к прерываниям. Например, для

удаления файла вместо функций *41h* или *7141h* прерывания *INT 21h* MS-DOS используется функция *API WIN32* с именем *DeleteFile()*. Для поддержания всеми приложениями длинных имен файлов, введенных в Windows, фирма *Microsoft* расширила набор функций прерывания MS-DOS *INT 21h*, которые полностью эквивалентны функциям *API WIN32*. Для того чтобы обратиться к любой функции, программист в тексте попросту использует ее имя. Так, для вызова функции *MessageBox()* в выполняемый файл при компоновке программы будет включена ссылка на точку входа номер 1 модуля *User*. Вызов приложением функции *LineTo()* будет заменен ссылкой на точку входа номер 19 модуля *GDI* (оба модуля являются ядром Windows). Во время загрузки программы в память Windows заменит ссылки адресами, по которым находятся указанные программы-функции. Любой модуль, предоставляющий подобные ссылки при компоновке, называется динамически компокуемой библиотекой (*DLL*). Каждая *DLL* декларирует набор точек входа, которые соответствуют внешним ссылкам. Большая часть Windows представляет собой совокупность *DLL*, которые используются для взаимодействия между различными компонентами.

Приложения MS-DOS, работающие под управлением Windows, попрежнему используют прерывания, в частности *21h*, включающее всевозможные функции, обеспечивающие услуги MS-DOS. Специальный модуль Windows преобразует прерывания в функции интерфейса *API WIN32*, которые фактически и выполняются. Работа современных компьютеров без системы прерываний немислима.

Защита памяти. Память адресуется сегментами и смещениями внутри сегмента. Возникновение сегментов связано с внутренней структурой процессоров фирмы *Intel*, в которых под адресацию памяти выделяются два регистра: в одном хранится адрес сегмента, а в другом – смещение внутри сегмента от его начала, например, *0040:0017*. Сегментом является определенный участок памяти. При 32-х разрядных регистрах процессора сегмент составляет около 4 Гбайт. Встроенный менеджер памяти за счет использования виртуальной памяти



Рис. 19. Адресация памяти

позволяет работать программам так, как будто имеется до 16536 сегментов доступных каждой задаче. Сегментный адрес является *селектором* (индексом-указателем) на *дескриптор* истинного адреса сегмента (рис. 19). Дескрипторы группируются либо в частную для *процесса* локальную (*LDT*) таблицу описателей, либо в глобальную (*GDT*) таблицу, которая используется совместно различными *процессами*.

Все таблицы дескрипторов хранятся в ОЗУ, а информация об их местонахождении хранится в регистрах *CPU* специального назначения. Каждый генерируемый *CPU* адрес включает селектор, который указывает, какую таблицу дескрипторов следует использовать и порядковый номер дескриптора в этой таблице. Позиции таблицы имеют длину в

8 байт. В каждой позиции хранится *дескриптор* – истинный адрес сегмента и его описание: пустой сегмент, сегмент хранит программу, хранит данные, используется только для чтения, а также указывается уровень доступа к нему или привилегия.

От несанкционированного доступа к данным в многозадачном режиме защита осуществляется как раз с помощью системы привилегий, регламентирующих доступ к тому или иному сегменту памяти в зависимости от уровня его защищенности и степени важности. Уровни привилегии задаются номерами от 0 до 3. Наиболее защищенная область памяти, отводимая под ядро операционной системы, имеет уровень 0. При обращении программы к сегментам программ или данных в защищенном режиме работы процессора происходит проверка уровня привилегии и в случае, если этот уровень недостаточен, происходит прерывание.

Следует заметить, что обмен данными центральный процессор может осуществлять только с оперативной памятью. Любые другие устройства по отношению к процессору будут внешними, поэтому данные для обработки с внешних устройств должны быть перемещены в оперативную память. Процесс перемещения с внешних устройств в ОЗУ данных и программ для выполнения называется *загрузкой*. Многие внешние устройства современных компьютеров имеют канал прямого доступа к памяти (*DMA*) – аппаратный интерфейс, позволяющий устройству передавать информацию в оперативную память и получать ее из памяти без участия центрального процессора, но не все устройства могут его использовать. Поэтому процесс загрузки данных в память может отнимать время у процессора, который вынужден приостанавливать свою работу и осуществлять операции обмена данными.

Многие быстродействующие компьютеры выполняют параллельную обработку данных на нескольких процессорах. Тем не менее, большинство из них в основном соответствуют принципам, изложенным Джоном фон Нейманом.

Заканчивая рассмотрение основных принципов работы компьютеров, дадим некоторые понятия и термины, связанные с ЭВМ.

Архитектура компьютера. *Архитектурой компьютера называют логическую организацию, описание структуры и ресурсов, которые может использовать программист, описание системы машинных команд, системы адресации к памяти и периферийным устройствам, организации памяти и др.* Архитектура не определяет особенности реализации аппаратной части ЭВМ, времени выполнения команд, ширины шин и других аналогичных характеристик. Архитектура отображает аспекты структуры ЭВМ, которые являются видимыми для пользователя: режимы адресации, форматы данных, набор программно-доступных регистров, систему команд и другие принципы действия, информационные связи, соединение основных логических узлов компьютера, их взаимодействие и т. д. Другими словами это стандарты на устройства, их подключение, на программы управления устройствами, *BIOS*, операционные системы и так далее.

IBM-совместимые компьютеры строятся по открытой архитектуре. Это значит, что все спецификации и стандарты фирмой IBM опубликованы.

В результате чего по этим стандартам сотни и тысячи компаний во всем мире выпускают электронные компоненты, разнообразное периферийное оборудование и программное обеспечение, в результате чего стоимость их постоянно снижается.

Классическая архитектура Джона фон Неймана – одно АЛУ, через которое проходит один поток данных, и одно ЦУУ, через которое проходит один поток команд – программа, была представлена на рис. 18.

Многопроцессорная архитектура приведена на рис. 20.

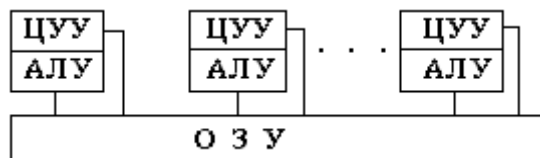


Рис. 20. Многопроцессорная архитектура ЭВМ

В этой архитектуре несколько процессоров работают с одним ОЗУ так, что параллельно может быть организована обработка нескольких потоков данных и нескольких потоков команд. Таким образом, параллельно могут выполняться несколько фрагментов одной и той же задачи.

Архитектура с параллельными процессорами приведена на рис. 21. Здесь несколько АЛУ работают под управлением одного центрального устройства управления и используют одно ОЗУ. В современных ЭВМ часто присутствуют элементы различных типов архитектурных решений.

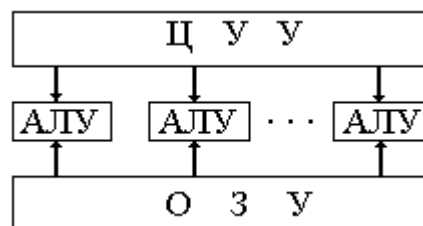


Рис. 21. Архитектура с параллельными процессорами

Многомашинная вычислительная система (МВС) – это несколько процессоров, входящих в вычислительную систему, которые не имеют общего ОЗУ, а имеют каждый свое. Каждый компьютер системы имеет классическую архитектуру. Такие системы применяются для решения специфических задач, когда задача разбивается на столько подзадач, сколько компьютеров в системе.

Часто употребляется термин **конфигурация ЭВМ**, под которым понимают компоновку вычислительного устройства с четким определением характера, количества, взаимосвязей и основных характеристик его функциональных элементов.

Термин **организация ЭВМ** определяет, как реализованы возможности ЭВМ. Принципы фон Неймана можно реализовать множеством способов. Рассмотрим два из них: ЭВМ с шинной организацией и канальной организацией обмена информацией.

Принципы организации ЭВМ

Принципы фон Неймана можно реализовать множеством способов. Рассмотрим два из них: ЭВМ с шинной и канальной организацией.

Шинная организация ЭВМ. Шинная организация является наиболее простой формой организации ЭВМ. В соответствии с приведенными принципами фон Неймана подобная ЭВМ имеет в своем составе следующие функциональные блоки, изображенные на рис. 22.

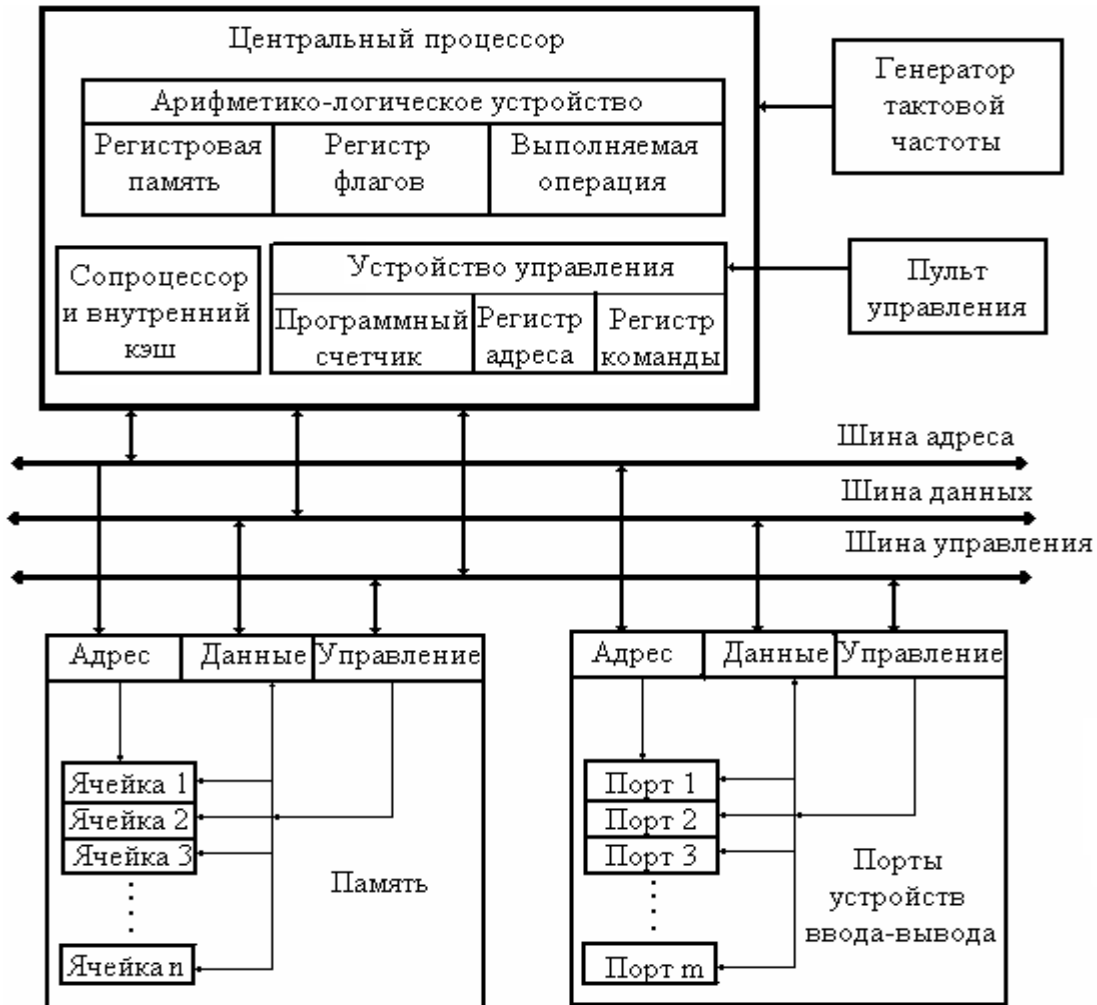


Рис. 22. Шинная организация ЭВМ

Центральный процессор – функциональная часть ЭВМ, выполняющая основные операции по обработке данных и управлению работой других блоков. Это наиболее сложный компонент ЭВМ как с точки зрения электроники, так и с точки зрения функциональных возможностей. Центральный процессор состоит из взаимосвязанных составных элементов: АЛУ, устройства управления и регистров, а также арифметического сопроцессора и небольшого объема быстродействующей памяти – кэш первого уровня. АЛУ содержит в своем составе устройство, хранящее характеристику результата выполнения операции над

данными, называемое *регистром флагов*. Отдельные разряды регистра указывают на равенство результата операции нулю, переполнение разрядной сетки (недопустимые операции), знак результата и др. Программный анализ флагов позволяет производить не только контроль за выполнением операций, но и операции ветвления программы в зависимости от конкретных значений данных. Кроме того, в АЛУ имеется набор программно-доступных быстродействующих регистров процессора.

Регистры составляют основу архитектуры процессора. Среди обязательного набора регистров можно отметить следующие. *Регистр данных (DX)* служит для временного хранения промежуточных результатов при выполнении операций. *Регистр аккумулятор (AX)* – регистр временного хранения, используемый в процессе вычислений (в нем формируется, например, результат выполнения команды умножения). *Регистр указатель стека (SP)* используется при операциях со стеком. Стек представляет собой формируемую в оперативной памяти область (структуру данных), которая работает по принципу: последним пришел – первым вышел. Стеки используются прерываниями и для организации работы подпрограмм. *Индексные, указательные и базовые регистры (IP, BP, SP, SI, DI)* используются для хранения и вычисления адресов (номеров ячеек памяти) операндов, участвующих в операции. *Регистры-счетчики (CX)* используются для организации циклических участков в программах. Каждый из регистров делится как бы на две части, например, *DX (DH и DL)*, *AX (AH и AL)* и т. д. *Регистры общего назначения*, имеющиеся во многих ЭВМ, могут использоваться для любых целей. Наличие большого количества регистров общего назначения с богатым набором возможностей упрощает программирование и повышает гибкость программного обеспечения. Кроме перечисленных, в состав АЛУ могут входить *внутренние системные регистры*, не доступные программно и используемые во время внутренних пересылок информации при выполнении команд.

Работа *устройства управления* и *ОЗУ* были описаны выше при рассмотрении принципов работы компьютера.

Периферийные устройства – это устройства внешней памяти, устройства ввода данных и выдачи результатов и др. Обмен данными с внешними устройствами осуществляется через *порты (адреса ввода-вывода)*. Иногда ошибочно портами называют разъемы, расположенные на задней стенке системного блока компьютера, служащие для подключения мыши, модема, сканера или принтера и т. д. Это неверно – разъемы предназначаются для подключения устройств к параллельным и последовательным портам. На самом деле под портами подразумеваются некие абстрактные каналы связи, организуемые в рамках системной шины и предназначенные для приема или передачи данных по команде процессора (аналогично морским портам – они служат для приема или отправления грузов). Каждому порту присваивается свой уникальный номер. Номера портов задаются 16-ю разрядами, так что потенциально можно иметь 65536 различных портов. В действительности используется гораздо меньше портов и, следовательно, имеются большие возможности для их наращивания.

Порт обеспечивает единственную возможность приема или передачи данных любым компонентам компьютера, кроме памяти. Для процессора любое, подключенное к системной шине, устройство является безличным «черным ящиком», с которым можно обмениваться данными. Он их распознавать по внешнему виду, по имени, выполняемым функциям и т. д. не может. Различать их процессор может только по номерам. Общение процессора с устройством сводится к посылке байтов по заданному адресу или считыванию их оттуда же. Куда конкретно были посланы данные или откуда они были считаны, процессор это не интересует. С равным успехом партнером процессора может быть контроллер жесткого диска, принтер или сетевая плата, звуковая плата или видеоадаптер. С этой точки зрения становится понятным, почему все каналы связи, как внутренние, так и внешние, называют портами – для процессора все они эквивалентны. Просто внешние оконечные устройства последовательных и параллельных портов оформлены в виде разъемов, вынесенных на заднюю панель системного блока для удобства подключения устройств, а внутренние порты пользователю невидимы. Внешне порты напоминают ячейки памяти – также по адресу (номеру порта) можно что-то записать или считать. Но порт с адресом $2F8$ и ячейка ОЗУ с этим же адресом – это не одно и то же. Для них и команды пересылки разные: в ОЗУ – команда *MOV*, а в порт – *OUT*. Считываются данные из порта по команде *IN*. Языки программирования высокого уровня имеют средства программировать порты. Операторы Бейсика *INP* и *OUT* предоставляют те же возможности, что и машинные команды.

```
10 OUT &H378,FF
20 X=INP(&H378)
30 PRINT X
40 END
```

Например, запишем в порт с номером 378 байт с единицами, а потом считаем их оттуда и распечатаем для проверки. Это параллельный порт *LPT1*, к которому обычно подключается принтер. Данный прием можно использовать

для определения устройства при ошибках печати, когда неизвестно, виноват ли принтер или системный блок. Отсоединив кабель от принтера, надо посмотреть осциллографом или логическим пробником на каждом контакте данного разъема сигнал – есть ли все единицы, посланные в порт, или нет. Если все единицы есть, то системный блок исправен, а сбой происходит по вине принтера, или – наоборот.

Порты используются схемами управления внешними устройствами – *адаптерами и контроллерами*. С каждым внешним устройством обычно связано несколько портов, выполняющих определенные функции на различных стадиях обмена. В одни порты заносятся управляющие признаки и адреса внутренних регистров (их номера) обслуживаемой микросхемы адаптера. Другие порты используются для обмена данными между *CPU* и выбранными внутренними регистрами. По содержимому третьих портов операционная система получает информацию о состоянии внешнего устройства, анализирует признаки завершения очередной процедуры обмена, контролирует правильность функционирования оборудования. Например, звуковая плата типа *Sound Blaster 16* имеет следующие порты: с 220 по 233 – звуковой тракт *WAV*, 330 и 331 – *MIDI* порты, с 388 по 38В – частотный синтезатор, с 200 по 207 – игровые порты для подключения манипуляторов типа джойстик.

Программы, обслуживающие обмен информацией с внешними устройствами на уровне портов и оформленные в соответствии с требованиями операционной системы, называются драйверами. Драйверы, как правило, пишутся на языке программирования, максимально приближенном к аппаратуре, чтобы учесть все особенности работы того или иного устройства. В Windows множество универсальных драйверов иерархической структуры включено в саму систему. При подключении нового устройства, Windows сама подключает необходимый драйвер, снимая эту заботу с пользователя. Драйвер Windows, управляющий конкретным адаптером, занимается взаимодействием с устройством на самом низком уровне, включая инициализацию адаптера и обслуживание аппаратных прерываний.

Все периферийные устройства через свои адаптеры и контроллеры подключаются к общему каналу передачи данных – *системной шине*, в том числе к ней подключаются системная память и процессор. В этом состоит еще один из принципов работы компьютера – *принцип взаимодействия устройств компьютера*. Концепция общей шины представляет собой один из наиболее совершенных методов унификации схем компьютеров. Вместо того чтобы соединять все компоненты компьютера между собой специальными соединениями, разработчики ограничили пересылку любых данных одним общим каналом – системной шиной (*BUS*). *Принцип чрезвычайно важен для подключения различных устройств к компьютеру, так как позволяет удобное подключение к нему нестандартных устройств, разрабатываемых самим пользователем, и дает возможность просто использовать компьютер в различных информационно – управляющих системах, тренажерах, обучающих системах, в схемах автоматики и т. д.*

Это результат открытой архитектуры компьютеров, которую с самого начала их выпуска провозгласила фирма IBM. Суть ее заключается в том, что все стандарты на устройства, их подключение, на программы управления ими – драйверы, BIOS и на операционные системы были открыты для любой фирмы. В результате по этим стандартам сотни компаний начали выпускать IBM-совместимые компьютеры, к ним добавились тысячи изготовителей электронных компонентов, разнообразного компьютерного оборудования и программного обеспечения, что положительно повлияло на пользовательский рынок, так как цены на эти изделия начали резко падать, а потребительские качества IBM-совместимых компьютеров улучшаться.

По линиям системной шины передаются данные (шина данных), отдельные линии передают адреса ячеек памяти или устройств (шина адреса), по другим линиям шины подводится питание ко всем устройствам, специальные сигналы управления ими и т. д. Данные пересылаются по шине в сопровождении специальных сигналов, обозначающих их назначение.

Каждое устройство, подключенное к системной шине, ожидает сигнала ввода/вывода. Если по шине выдана команда обмена с ОЗУ, то все устройства будут игнорировать ее. При запросе операции обмена данными на шину выдается адрес (номер устройства), который проверяется адаптером каждого устройства. Если адрес на шине совпадает с адресом устройства, то оно начинает выполнять операцию по вводу/выводу. Эта идея чрезвычайно упростила конструкцию компьютеров и существенно увеличила ее гибкость.

В современных компьютерах используется не одна системная шина, а несколько, в зависимости от скорости передачи данных устройствами. *Скоростные устройства (видеоадаптеры, жесткие диски, платы обработки видеоизображений и др.) подключаются к быстрой шине, а медленнодействующие устройства – к низкоскоростной шине.* Шины расположены на системной плате, к каждой из них присоединены параллельно между собой несколько разъемов (*слотов расширения*). Шины позволяют легко и просто подключать к компьютеру дополнительные устройства для расширения его возможностей. В слоты расширения устанавливаются платы схем управления устройствами – адаптеров и контроллеров, а к ним подключаются сами устройства. Количество слотов расширения может быть разным, но не превышает восьми. Этого достаточно для подключения самого разнообразного оборудования. Следует подчеркнуть, что часть адаптеров размещается непосредственно на системной плате или в самом устройстве и не требует слотов расширения. *Каждый адаптер или контроллер может быть подключен только к той шине, на работу с которой он рассчитан.* В некоторых адаптерах на заднюю стенку системного блока выведены разъемы для подключения соответствующих устройств, например клавиатуры, монитора, мышки, печатающего устройства и др.

За последние 15 лет было разработано довольно много вариантов шин. Объясняется это стремлением получить наивысшее быстродействие работы компьютера. Как это ни удивительно, но до сих пор почти все компьютеры выпускаются с теми же шинами, что и первые *IBM PC/AT*. В новые компьютеры встраиваются быстродействующие *локальные шины*, обслуживающие специально разработанные под них адаптеры.

Одной из главных причин, сдерживающих появление новых структур шины, является их несовместимость со старым стандартом *PC*. В свое время успех *PC* предопределила стандартизация – множество фирм разработали и выбросили на рынок десятки и сотни тысяч плат в этом стандарте. Поэтому новая, более быстродействующая шина должна быть совместимой с прежним стандартом, иначе все старые платы можно будет выбросить на свалку. Поэтому технология шин развивается медленно, без резких скачков. Различия между шинами связаны, в основном, с разрядностью и скоростью передачи данных. Каждая шина строится на основе специализированного комплекта микросхем, которые подключаются к шине процессора. Управляются шины своими адаптерами или контроллерами.

Одной из первых шин в 1982 г. была использована 8-разрядная шина стандарта *ISA (Industrial Standard Architecture)*, в 1984 г., в связи с появлением 16-разрядного процессора *Intel 286* она была модернизирована в 16-разрядную. На сегодняшний день это основной тип шины, используемой в большинстве выпускаемых компьютеров. Разъемы шины были удлинены так, что в них можно вставлять как 8-разрядные, так и 16-разрядные платы адаптеров. На первый взгляд покажется странным, что в современных высокопроизводительных компьютерах применяется шина с такой древней архитектурой, но это объясняется ее надежностью, достаточно широкими возможностями и совместимостью. Эта старая шина работает значительно быстрее, чем многие подключаемые к ней

устройства, например, клавиатура, мышь, модем, дисковод для дискет и другие. Шина содержит 98 линий, из которых 16 используются для передачи данных, 15 – для передачи сигналов аппаратных прерываний, 8 – для каналов прямого доступа (*DMA*) к системной памяти, 24 адресных линии и т. д. Для передачи данных по шине *ISA* требуется от двух до восьми тактов, а максимальная скорость передачи составляет 8,33 Мбайт/с.

К середине 1992 г. появилась шина *PCI* (*Peripheral Component Interconnect*), в апреле 1993 г. она была модернизирована, а к середине 1993 г. появились первые компьютеры с этой шиной. Шину иногда называют *антресольной*, так как она добавляет к традиционной конфигурации шин еще один уровень. Вместо того чтобы подключаться непосредственно к шине процессора, весьма чувствительной к подобным вмешательствам, разработчики создали новый комплект интегральных схем контроллеров для расширения системной шины. По 198-ми линиям шины так же, как и по шине *ISA*, передаются самые разнообразные данные.

Преимущества шины *PCI* в компьютерах с процессором *Pentium* очевидны. Первоначальная тактовая частота шины была равна 33 МГц, а разрядность соответствует разрядности данных процессора. В настоящее время тактовые частоты этой шины в разных компьютерах составляют 400–500 МГц и более. Высокая пропускная способность шины объясняется тем, что она может работать параллельно с шиной процессора, не обращаясь к ней со своими запросами. Например, процессор может работать с данными из внешнего кэша, а шина *PCI* одновременно осуществлять обмен информацией между быстродействующими жестким диском и видеоадаптером. В этом заключается одно из достоинств шины. Шина *PCI* легла в основу разработанного впоследствии стандарта *PnP* (*plug-and-play – вставляй и работай*) для внешних устройств, который не требует настройки плат адаптеров. С конца 1995 г. в компьютерах устанавливаются *PnP BIOS*, выполняющие автоматическую настройку системы.

Тенденция последних лет – новые периферийные устройства стремятся делать последовательными, что как бы противоречит принципу повышения пропускной способности каналов. Известно, что по параллельному каналу можно передавать одновременно несколько байт, а по последовательному – только бит за битом. Но частоты процессоров растут и достигают уже 10 ГГц, а специалисты знают, что повышать тактовую частоту параллельного канала можно только до определенной величины, выше которой начинаются искажения передаваемой информации. Кроме того, необходимость приведения периферийных устройств в соответствие с принципом *plug-and-play* и избавления от клубка проводов и кабелей на задней стенке системного блока вызвало появление на свет стандартов *USB* и *IEEE 1394*, а также несколько разновидностей интерфейса *SCSI*.

Шина *USB* разработана как универсальное средство для подключения к компьютеру самых разнообразных периферийных устройств. Фирма *Intel* практически во всех выпускаемых ею комплектах ИС для системных плат, начиная с *Triton II (82430HX и VX)*, предусмотрела поддержку этой шины. Шина представляет собой четырехпроводную линию связи с быстродействием 12 Мбит/с. К ней можно подключать до 127 устройств по многоярусной (древовидной)

схеме с использованием одного или нескольких распределительных устройств, которые могут располагаться в самом компьютере, в любых периферийных компонентах или быть выполненными в виде автономных блоков. Для медленнореагирующих устройств имеется отдельный низкоскоростной канал с быстродействием 1,5 Мбит/с.

В шине *USB* реализована технология *plug-and-play*, в том числе и возможность «горячего» подключения. Любое устройство может быть подключено к работающей шине без выключения или перезагрузки системы. Контроллер шины обнаружит факт подключения нового устройства, самостоятельно выделит ему необходимые системные ресурсы и загрузит соответствующие драйверы. Драйверы для *USB*-устройств включены в операционные системы Windows последних лет. Предусмотрена также поддержка шины системной *BIOS*. Во всех новых компьютерах на системной плате установлен порт шины *USB* для подключения устройств этого стандарта.

Интерфейс *SCSI*. Само название (*Small Computer System Interface*) подчеркивает, что интерфейс является *системным*. Это не очередная разновидность контроллера, а шина, способная обеспечить работу нескольких устройств, каждому из которых присваивается свой идентификационный номер – *SCSI ID*. Одно из этих устройств называется *основным (host) адаптером* и является связующим звеном между системной шиной и шиной *SCSI*. Сама же шина *SCSI* взаимодействует не с устройствами (например, с дисковыми накопителями), а с встроенными в них контроллерами. Взаимодействие с *SCSI*-устройствами осуществляется через основной адаптер, устанавливаемый в слот системной шины. К одному основному адаптеру можно подключать до 7 устройств, а некоторые *SCSI*-адаптеры позволяют подключать 15 и даже 31 устройство. Среди этих устройств могут быть жесткие магнитные диски, магнитооптические и лазерные диски, стримеры, принтеры, сканеры и даже другие компьютеры. В большинстве компьютеров можно устанавливать до 4-х основных адаптеров, что позволяет подключить от 28 до 124 устройств. Операции ввода/вывода с различными *SCSI*-устройствами могут осуществляться параллельно, т. е. не надо ожидать завершения операции обмена с одним устройством, чтобы начать обмен с другим.

Состав и назначение шин, а также правила их использования, виды передаваемых по шине сигналов и другие характеристики шины могут существенно различаться у разных ЭВМ. Однако есть принципиально общие закономерности в организации шин. Шина состоит из отдельных проводников (линий) и оконечных логических схем с разной функциональностью. Сигналы по линиям могут передаваться потенциальные (разные уровни напряжения) или импульсные. *Шириной* шины называется количество линий, входящих в состав шины.

Обычно на шине в любой момент времени можно выделить два активных устройства. Одно из них называется *задатчиком* и инициирует операцию обмена данными (формирует адреса и управляющие сигналы), другое называется *исполнителем* и выполняет операцию (дешифрирует адреса и управляющие сигналы и принимает или передает данные). В большинстве случаев задатчиком является ЦП. Память всегда выступает только в качестве исполнителя.

Из управляющих линий выделим следующие.

Линии занятости. Если они в состоянии «шина свободна», любой задатчик может начать операцию обмена данными на шине, иначе задатчику придется ожидать, пока шина не освободится.

Линии выполняемой операции. Указывают, какая именно операция будет выполняться: чтение (задатчик принимает данные от исполнителя) или запись (задатчик передает данные исполнителю).

Линии синхронизации. Синхронизация задатчиков и исполнителей может осуществляться по-разному в зависимости от выполняемых операций.

Большинство мини – и микро – ЭВМ, в том числе персональные компьютеры, имеют шинную организацию.

В различных ЭВМ реализация алгоритма выполнения программы может несколько отличаться. Так, например, по-разному может осуществляться синхронизация задатчиков и исполнителей, процессор может считывать из памяти не одну команду, а сразу несколько и хранить их в специальной очереди команд. Часто используемые программой команды и данные могут храниться не в оперативной памяти, а в быстродействующем кэш и т. д.

Канальная организация ЭВМ. В основе этого типа организации ЭВМ лежит множественность каналов связи между устройствами и функциональная специализация узлов. Упрощенная схема организации ЭВМ с каналами приведена на рис. 23.

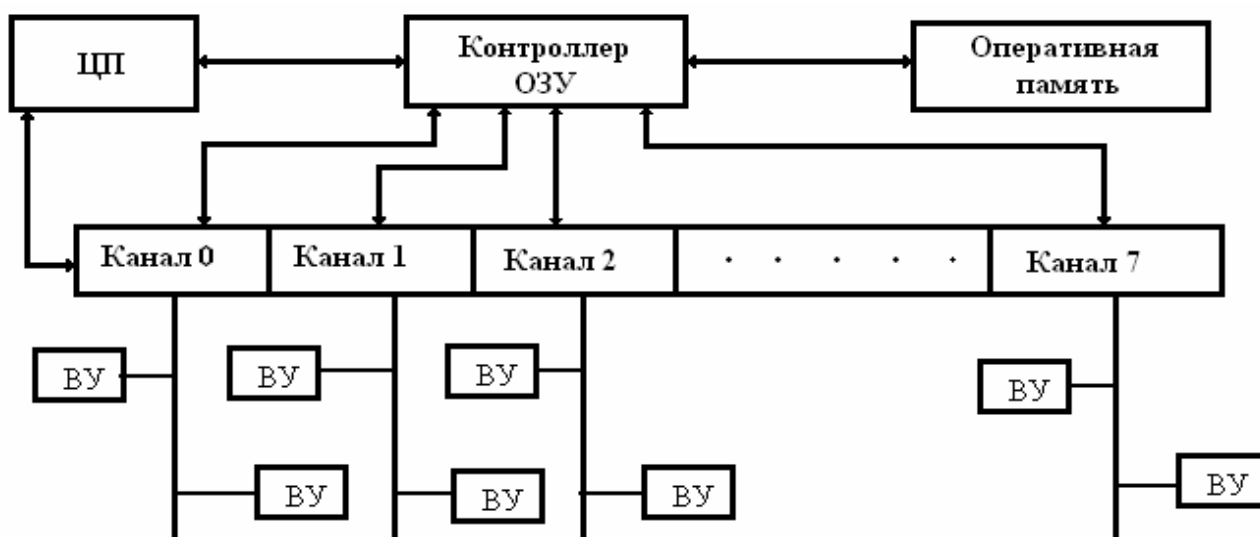


Рис. 23. Канальная организация ЭВМ

Все фон – неймановские ЭВМ очень похожи друг на друга и алгоритм функционирования центрального процессора практически ничем не отличается от рассмотренного выше. Помимо уже знакомого набора устройств в состав ЭВМ с канальной организацией входят устройства, называемые каналами. *Канал* – это специализированный процессор, осуществляющий всю работу по управлению контроллерами внешних устройств и обмену данными между основной памятью и внешними устройствами. Каналы имеют с различной скоростью передачи и подразделяются на *селекторные* и *мультиплексные*. «Быстрые» устройства (например, накопители на жестких магнитных дисках) подсоединяются к селекторным кана-

лам. Такое устройство получает селекторный канал в монопольное пользование на все время выполнения операции обмена данными.

«Медленные» устройства подключаются к мультиплексным каналам. Мультиплексный канал разделяется (мультиплексируется) между несколькими устройствами, при этом возможен одновременный обмен данными с несколькими устройствами. Доступ к оперативной памяти может получить и центральный процессор и один из каналов. Для управления очередностью доступа имеется *контроллер оперативной памяти*. Наименьший приоритет имеет центральный процессор. Среди каналов больший приоритет имеют медленные каналы. Таким образом, приоритет обратно пропорционален частоте обращения устройств к памяти.

Аналогично подсоединяются устройства и в персональных компьютерах с шинной организацией. «Быстрые» устройства подсоединяются к быстрой шине (например, к *PCI* подключаются сетевые карты, *HDD* и др.), а «медленные» устройства подсоединяются к медленной шине (например, флоппи диски, клавиатура и др. подсоединяются к медленной шине *ISA*).

За счет существенного усложнения организации ЭВМ упрощается архитектура ввода-вывода. Операции обмена данными становятся более простыми. Канал, по сути, представляет собой специализированный «интеллектуальный» контроллер прямого доступа к памяти. (Такие контроллеры *DMA-Direct Memory Access* есть и у некоторых устройств в персональных компьютерах с шинной организацией). Для ускорения обмена данными реализованы несколько трактов обмена: процессор – основная память и каналы – основная память. О своем состоянии канал может информировать процессор с помощью прерываний. Свобода подключения внешних устройств к каналу обеспечивается благодаря стандартному протоколу интерфейса, при этом появляется возможность группировать устройства по характеристикам.

Канал, являясь хотя и специализированным, но процессором, выполняет свою *канальную программу*. Она состоит из канальных команд и хранится в оперативной памяти. Подготовку канальной программы и загрузку ее в оперативную память выполняет операционная система. После того как канальная программа подготовлена, адрес ее начала размещается в фиксированной ячейке памяти, называемой словом адреса канала (*CAW-Channel Address Word*).

Для управления каналами процессор имеет всего несколько команд. Операция обмена данными инициируется центральным процессором с помощью команды *SIO M,N* (*Start Input-Output*, где *M*-номер канала, а *N*-номер устройства в канале). Выдав команду запуска обмена, процессор, не обращая внимания на обменный процесс, продолжает выполнять свою программу. Он может проверить состояние канала с помощью команды *TIO (Test Input-Output)* – опросить ввод-вывод. Команда *SIO M,N* передается во все каналы, но воспринимает ее только канал *M*. Если канал занят, то он устанавливает соответствующее состояние своих регистров, и процессор по команде тестирования может выяснить, что запуск канальной программы не состоялся. Если канал свободен, то, во-первых, он выбирает из оперативной памяти *CAW* в свой регистр, во-вторых, передает подключенным к нему устройствам команду *SIO*, которую читают все устройства, но воспринимает только устройство с номером *N*. Если устройство занято или не готово, то в регистрах канала устанавливаются соответствующие

состояния и процессор по команде тестирования снова может узнать о том, что операция обмена данными не состоялась. Если же устройство готово и свободно, оно устанавливает в интерфейсе сигнал ожидания. Вся дальнейшая обменная операция протекает по инициативе внешнего устройства. Получив сигнал ожидания, канал выбирает по адресу *CAW* адрес начала канальной программы и передает ее контроллеру внешнего устройства, где она и выполняется.

Канальные программы могут быть подготовительными или командами обмена данными. Подготовительные команды устанавливают режимы работы внешних устройств, осуществляют операции поиска и т. д. Обменные команды содержат коды операций и адреса оперативной памяти. Обмен происходит по асинхронной схеме по инициативе внешнего устройства. Данные извлекаются из памяти и помещаются в память напрямую, без посредников.

После выполнения команды канал проверяет в выполненной команде признак конца. Если это не последняя команда, меняется адрес *CAW* и выбирается следующая команда. Если команда последняя, канал «привлекает к себе внимание» процессора с помощью сигнала прерывания, по которому обработчик, являющийся частью операционной системы, выполняет операции, завершающие обмен данными.

Канал может сгенерировать сигнал прерывания до окончания выполнения канальной программы при возникновении исключительных ситуаций. В этом случае операционная система запрашивает состояние регистров канала и выясняет, что именно произошло, определяет, какие действия необходимо предпринять в возникшей ситуации.

Отметим некоторые особенности канальных машин. Несколько подряд идущих канальных команд могут образовать цепочку данных. В этом случае имеется одна команда обмена, например чтения физической записи из нескольких адресов оперативной памяти со счетчиков. Одна физическая запись распределяется в несколько адресов оперативной памяти.

В ЭВМ с канальной организацией центральный процессор практически полностью освобождается от рутинной работы по организации ввода-вывода данных. Наличие нескольких трактов передачи данных снимает трудности, связанные с блокировкой единственного канала передачи данных (системной шины), что повышает скорость обмена. В результате общая производительность системы существенно возрастает, за счет чего удорожание схемы окупается. Ярким примером ЭВМ с канальной организацией являются большинство мен-фреймов, начиная еще с машин семейства IBM-360/370, появление которых произвело переворот в вычислительной технике. Все машины семейства ЕС ЭВМ строились по такой схеме. В настоящее время накоплено богатое наследие в виде интересных архитектурных решений, программных и алгоритмических разработок, поэтому канальная организация ЭВМ встречается довольно часто.

Многие быстродействующие компьютеры выполняют параллельную обработку данных на нескольких процессорах. Тем не менее, большинство из них в основных чертах соответствуют принципам, изложенным Джоном фон Нейманом.

Принципы хранения информации в компьютере

Вся информация в компьютере хранится на **внешних** несъемных или сменных носителях. Обычно это накопители на магнитных, оптических и магнитооптических дисках. Как можно найти любого человека по адресу его проживания, так можно найти любую информацию на дисках компьютера, используя для этого специально созданную адресную структуру хранения – *файловую систему ОС*. Каждая операционная система имеет свою файловую систему.

Внешние накопители на дисках, как и любые устройства компьютера, имеют свои *физические номера*. Так, активный жесткий магнитный диск, с которого загружается операционная система, всегда имеет номер *80h*. На логическом же уровне пользователю удобнее работать с именами, поэтому накопителям на дисках присваиваются *имена* в виде букв латинского алфавита с двоеточием. Принято дисководы съемных магнитных дисков (*дискет*) обозначать именами *A:* и *B:*. Жестким несъемным дискам (*винчестерам*) присваивают имена, начиная с имени *C:*. Часто общее физическое пространство винчестера с помощью специальных программ делят на отдельные области – разделы, это аналогично строительству складов на площади, до того лишь *отведенной* под них. В зависимости от назначения разделы могут быть основными (*primary*) и дополнительными (*extended*), а также активными (*active*), системными (*system*) и загрузочными (*boot*). Деление жесткого диска на разделы может быть вызвано одной из следующих причин:

- объемы жестких дисков достигают нескольких сотен Гбайт и не все операционные системы могут работать с дисками большой емкости;
- желанием обеспечить защиту части информации, предоставляемую, например, операционной системой *Windows XP*. Эту информацию необходимо поместить в один из разделов с файловой структурой *NTFS*;
- пользователю удобнее и проще работать, если разместить отдельные группы однотипной информации в разных разделах, чем всю ее хранить вместе.

Каждую такую область физического диска оформляют в виде *логического диска* (диска физически не существующего, а фиктивно созданного программным путем). Этим логическим дискам присваивают имена в виде букв, следующих за *C:*, то есть *D:*, *E:*, *F:*, *G:*, *H:* и т. д. Для появившихся недавно съемных устройств флэш памяти при установке устройства в разъем *USB*, присваивается имя, свободное за всеми именами имеющихся разделов жестких дисков. Чтобы обратиться к тому или иному диску для записи или чтения информации, необходимо указать его имя (как при обращении к человеку). Диск, с которым в данный момент работает пользователь, называется *текущим, или активным. Имя диска – высший уровень в системе адресации.*

Информация на дисках записывается в *файлы*. *Файл – это выделенная на диске или другом носителе область, имеющая имя.* Имя файлу присваивает

пользователь произвольно – по своему усмотрению. Как люди общаются между собой по именам, так и программы в компьютере обращаются к тому или иному файлу по имени. В файле может храниться различная однотипная информация. Это может быть графическое изображение или текст какого-либо документа, массив числовых данных или программа в машинных кодах, таблица расчета рентабельности предприятия или музыкальный сюжет. В последние годы появились файлы типа *мультимедиа*. В таком файле одновременно хранится разнородная информация, например, видеокадры, музыкальные сюжеты и текст. *Файл является еще одним уровнем системы адресации.*

Файлы, относящиеся к какой-то проблеме или предназначенные для выполнения какой-то определенной работы, группируются, их имена и характеристики регистрируются в специальных таблицах или других структурах на диске – *каталогах, или директориях*. Создаваемые пользователями каталоги являются своеобразными файлами, поэтому им также присваивают имена и регистрируют в других каталогах. Имена им задает пользователь по своему усмотрению, но так, чтобы можно было по имени определить, какие файлы содержит данный каталог. *Это будет третий из уровней системы адресации.*

Таким образом, для записи (чтения) какого-то документа или программы на диск, необходимо, как почтовый адрес, указать *путь*: имя диска – имя каталога – имя файла, в который будет записываться информация.

Файлы.

Присвоение имен файлам и каталогам хотя и производится пользователем произвольно, однако зависит от используемой операционной системы.

Некоторые сочетания символов нельзя использовать в качестве имен файлов, так как операционная система резервирует эти имена для обозначения *системных устройств*. К ним относятся:

PRN	–	имя принтера;
LPT1-LPT4	–	устройства, подключаемые к параллельным портам 1–4;
CON	–	при вводе информации – это клавиатура, а при выводе – экран;
COM1-COM4	–	устройства, подключаемые к последовательным портам;
AUX	–	синоним порта COM1;
NUL	–	«пустое» устройство. Для него все операции ввода–вывода игнорируются, но программе сообщается, что ввод – вывод произошел успешно.

Файлы с такими именами операционная система просто не регистрирует, но эти имена допустимы в качестве расширений файлов, например: *TEST.PRN, IA.CON*. При создании файла или изменении его содержимого автоматически регистрируются *дата и время* по показаниям календаря и часов системы. *Имя, тип, размер в байтах, дата и время создания файла фиксируются в каталоге и являются его характеристиками.*

Во многих командах (найти, копировать, удалить, распечатать и др.) для указания сразу нескольких различных файлов из одного каталога используются *шаблоны, или обобщающие символы * и ?*. Знак * обозначает любые допустимые для имен и расширений файлов символы, а знак ? обозначает любой одиночный символ. Примеры шаблонов:

I.XLS* – все файлы типа *XLS*, имеющие имена с буквы *I*;

*.*EXE* – все файлы с расширением *EXE*;

.???* – все файлы;

*P.???.** – все файлы, имеющие имена с буквы *P* не более 4-х символов.

Группу файлов из разных каталогов выделить нельзя.

В большинстве случаев расширения имен файлов стандартизированы и присваиваются им автоматически при сохранении.

Например, все файлы исполняемых программ имеют тип *.COM* или *.EXE*. Все инструментальные системы, а также и многие программы при создании текстов присваивают свои расширения файлам, хранящим эти тексты: *.BAS* – язык программирования Бейсик, *.C* – язык программирования Си, *.PRG* – многие системы управления базами данных, *.PAS* – язык программирования Паскаль, *.DOC* – текстовый редактор *Word*, *.XLS* – табличный процессор *Excel* и т.д.

Файлам могут присваиваться *атрибуты* (признаки) «только для чтения» (*R/O-read only*), «скрытый» (*Hid-hidden*), индексирования и архивации, сжатия и шифрования. Атрибуты устанавливаются в диалоговом окне, открываемом при щелчке правой клавишей мыши на имени файла и выполнении команды *Свойства*. Файлу может быть установлен один или несколько атрибутов одновременно. Назначение их можно посмотреть в этом же окне, щелкнув в строке заголовка окна знак вопроса (?), а затем щелкнув на соответствующем признаке. Флажки атрибутов *файл готов для архивирования* и *индексировать содержимое для быстрого поиска* устанавливаются автоматически при первом сохранении файла.

В современных операционных системах *Windows* разных версий, *OS/2* и других файлам и каталогам разрешается присваивать имена длиной до 254 символов. Можно использовать символы русского алфавита, пробелы, символы + _ - , ; = . []. Это позволяет файлам присваивать наглядные и понятные имена. Прописные и строчные буквы в таких именах файлов должны различаться, что обеспечивает удобочитаемость. Однако в одном и том же каталоге файлы с одинаковыми именами, отличающимися только регистром букв, недопустимы – они просто не будут зарегистрированы. Примеры:

Реферат на тему: Реформы Петра I .doc

Отчет по лабораторной работе.xls

Не рекомендуется длинные имена задавать более 60–70 символов, так как в них неудобно ориентироваться. Имя файла вместе с указанным путем доступа к нему (*спецификацией*) не должно составлять более 260 символов, иначе файл, помещенный в глубоко вложенный каталог, некоторыми программами не будет найден. Кроме того, имя такого файла зарегистрированного в *корневом* каталоге, резко сокращает объем каталога.

Каталоги.

При интенсивной работе на компьютере число файлов быстро растет, и «следить за порядком» на диске становится все сложнее. Структурировать и упорядочить дисковое пространство позволяют каталоги файлов. Каталоги представляют собой простые таблицы или более сложные структуры, например, в виде деревьев, то есть по-существу это тоже поименованные участки на дисках или своеобразные файлы. Структура их зависит от установленной опе-

рационной системы. В MS-DOS каталог представляет таблицу, состоящую из строк по 32 байта каждая (файловая система типа FAT). Такая же структура каталогов создается на дисках и картах флэш-памяти, независимо от операционной системы. В одной строке (позиции) может быть зарегистрирован один файл или каталог пользователя, который регистрируется как обыкновенный файл. Каталог напоминает районный паспортный стол, где каждый из нас (файлов) имеет регистрационную карточку (позицию каталога) с данными о проживании. При подготовке диска к работе (форматировании) создается *главный, или корневой* каталог – основа адресной структуры системы. Имя этого каталога состоит из одного символа \ и присваивается программой форматирования автоматически. Принципиальное отличие корневого каталога от пользовательских состоит в том, что данные о нем нигде не зарегистрированы, как об остальных каталогах (просто нет над ним структуры, где бы он мог зарегистрироваться). А раз нет о нем информации, то изменить имя корневого каталога или удалить каталог с диска невозможно. В отличие от каталога, создаваемого пользователем, который может занимать все дисковое пространство, количество позиций в корневом каталоге ограничено и зависит от объема диска. После форматирования все позиции корневого каталога пустые, так как при форматировании вся информация на диске уничтожается. По мере записи информации на диск позиции занимают информацией о файлах или пользовательских каталогах. На рис. 24 приведен фрагмент корневого каталога небольшого системного диска с ОС MS DOS, выведенный на экран в виде, формируемом утилитой Нортонa *NU.EXE*.

Sector 171 in root directory

Filename	Ext	Size	Date	Time	Cluster	Arc	Attributes				Vol
							R/O	Sys	Hid	Dir	
VICTOR			6/01/99	18:54							
IO	SYS	40774	19/08/94	12:00	281	Arc	R/O	Sys	Hid		
MSDOS	SYS	38138	19/08/94	12:00	318	Arc	R/O	Sys	Hid		
DRV			6/01/99	22:21	8361						Dir
DOS			17/09/98	18:22	149						Dir
ARC			6/01/99	22:26	905						Dir
AUTOEXEC	BAT	496	28/01/99	15:19	92	Arc					
PASCAL			17/01/99	6:33	9844						Dir
COMMAND	COM	54869	19/08/94	12:00	1525	Arc	R/O		Hid		
NC			6/01/99	20:40	254						Dir
CONFIG	SYS	143	28/01/99	15:20	94	Arc					
xATR	PAS	47	24/12/98	11:37	2680	Arc					

Рис. 24. Фрагмент каталога в системе FAT

Из рисунка видно, что при регистрации файла в каталоге записываются его имя, тип, размер в байтах, дата и время создания, номер кластера (участок диска), начиная с которого записывается файл и атрибуты файла. Каталоги, созданные пользователем, имеют метку *Dir*. Они аналогичны файлам. По количеству позиций ограничены только объемом диска. Пользовательские каталоги являются подчиненными тому каталогу, в котором они зарегистрированы, поэтому называются *подкаталогами*. Каталог, в котором зарегистрирован подка-

талог, является для него *родительским*. Каждый пользовательский каталог может включать в себя еще подкаталоги. При записи пути к файлу каталоги отделяются друг от друга обратной косой чертой (\). Так образуется *иерархическая, древовидная файловая система диска*. При удалении файла или каталога с диска первый символ его имени в каталоге заменяется строчной русской буквой х (с кодом 229), сообщающей всем программам, что к файлу доступа нет – он удален. Физически же он сохраняется до тех пор, пока на это место не будет что-то записано вновь (см. последний файл на рис.24). Это позволяет специальным программам восстановить файл или каталог, если они были удалены случайно. В Windows при удалении файлов нужно использовать специальный контейнер (каталог с именем *Корзина*), в который переписываются удаляемые файлы и который позже можно очистить. При необходимости папку *Корзина* можно открыть и восстановить удаленные случайно файлы.

Каталог, с которым в настоящий момент работает пользователь, называется *текущим, или активным*. Для доступа к файлу, зарегистрированному в текущем каталоге, указывать путь не требуется. Для того чтобы посмотреть, какие файлы и подкаталоги зарегистрированы в каталоге, необходимо его *раскрыть – вывести на экран оглавление*.

Если при форматировании диска ему была присвоена *метка тома (Vol)*, то она также регистрируется в каталоге. При этом в позиции каталога записывается: в поле имени файла – имя (метка), дата и время создания метки и признак метки *Vol* (см. позиции каталога на рис. 24). Фактическое расположение информации в строке каталога показано на рис. 25.

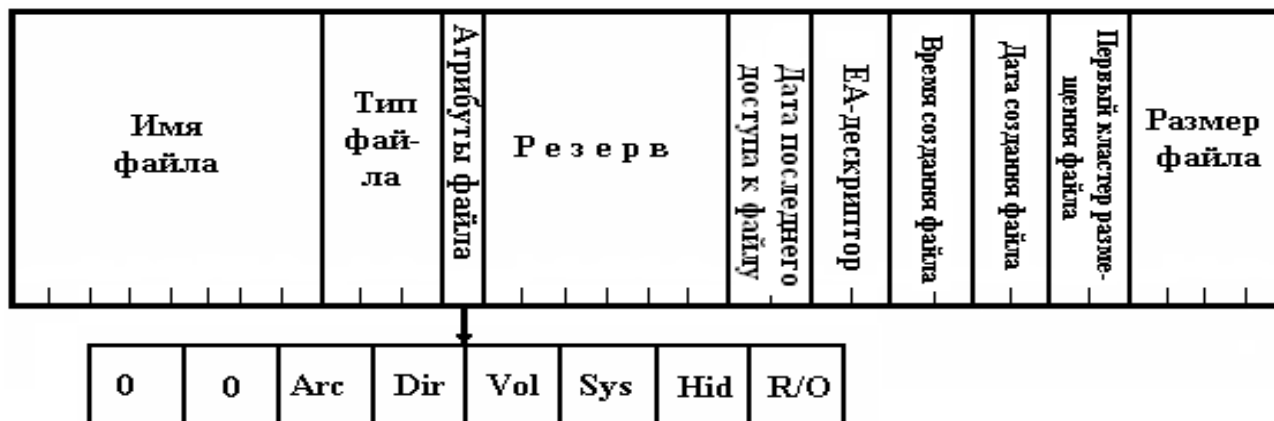


Рис. 25. Расположение информации в строке каталога

Именно признаки *Vol, Hid, Sys u R/O (0Fh* – см. рис. 25) байта атрибутов в позиции каталога используются в Windows для задания длинных имен файлов и каталогов. Оказывается, что практически все прикладные программы «не видят» позиции каталогов с совокупностью этих меток. Поэтому, чтобы не создавать новую файловую систему, было решено записывать длинные имена файлов и каталогов и информацию, связанную с ними, в элементы каталога рядом с коротким именем, ставя в них признаки *Vol, Hid, Sys u R/O*.

Таким образом, на диске совершенно четко и однозначно указывается местоположение любой информации, как почтовый адрес для людей. Например, программа текстового редактора, с помощью которого формировался текст лекций этого сборника, расположена по следующему адресу: *C:\Program Files\Microsoft Office\Office11\WinWord.exe*. В новых операционных системах Windows и OS/2 каталоги называются *папками*. Более подробно файловая структура на диске изучается на практических занятиях при работе на компьютере в лаборатории.

Лекция 20

Физический и логический уровни работы с компьютером

Как уже упоминалось ранее, каждое устройство компьютера работает под воздействием своей схемы управления – *адаптера или контроллера*. Адаптер или контроллер выполняют специальную программу управления устройством – *драйвер*. В большинстве адаптеров и контроллеров используются специализированные или простые процессоры общего назначения – устройства по обработке информации. В центральном процессоре компьютера также встроено устройство управления арифметико-логическим устройством, обрабатывающим информацию. В *лекции 17* подчеркивалось, что и центральный процессор и устройства могут выполнять ограниченное количество самых простейших действий, или *микроопераций*. Например, адаптер накопителя на флоппи-дисках может выполнять только 18 простейших микроопераций, таких, как включить/выключить двигатель, позиционировать головки чтения/записи на заданный номер дорожки, читать/писать информацию из заданного сектора (в сектор) и т. д. Арифметико-логическое устройство тоже умеет выполнять определенный набор простейших операций, таких, как логические (сравнения, проверки двоичных чисел), пересылки (сдвиги), арифметические (сложение двоичных чисел). Все остальные арифметические операции, как известно, реализуются через операцию сложения.

Каждая микрооперация выполняется под воздействием *микрокоманды*. Микрокоманды в виде двоичных посылок (импульсов) поступают на входы схем управления соответствующих устройств и расшифровываются. Расшифровка состоит в том, что выбираются соответствующие данной микрокоманде выходные линии дешифратора, на которых появляются сигналы (импульсы). Они воздействуют на связанные с выбранными линиями элементы схем устройств, переключая их определенным образом и заставляя выполнять действия, соответствующие микрооперации. Подавая на вход адаптера серии таких микроопераций, можно управлять работой устройства.

Такие простые действия, как сложение двух двоичных чисел, выполняются по одной или нескольким микрокомандам. Но компьютеры уже давно превратились из «считалок» в мощные информационные системы для накопления,

хранения и обработки больших массивов информации. Поэтому большинство их действий – это не простые вычислительные операции, а сложные операции с устройствами по обмену данными, упорядочиванию, группировке или определенной выборке данных и т. д., требующие выполнения больших последовательностей простейших действий устройств. Такие операции, как «записать файл на диск» или «считать файл с диска», «распечатать информацию на экран или на принтер» и т. д., требуют для своего выполнения больших последовательностей простейших микроопераций, или *программ в машинных кодах*, о чем мы уже упоминали.

Программирование в машинных кодах и управление устройствами с помощью программ, написанных в машинных кодах, называют работой на физическом уровне. Управление на физическом или аппаратном уровне является сложной задачей – для этого надо очень хорошо знать структуру и принципы работы системы. Степень детализации при написании программ на этом уровне наиболее высока, кроме того, достигается наивысшее быстродействие выполнения таких программ. Хороший программист может написать программу непосредственно в машинных кодах, но обычно, в целях облегчения процесса программирования, используется язык программирования чуть более высокого уровня – *ассемблер*. Непосредственно в машинных кодах программы писали на заре зарождения компьютерной техники. Сейчас язык ассемблера - это тот уровень, ниже которого работать просто бессмысленно. *Такие программы всегда являются строго специализированными, они жестко привязаны к аппаратуре, к конкретному оборудованию.*

Таким образом, действия прикладных программ или пользователя, связанные с аппаратными средствами, должны представляться в виде выполнения стандартных наборов машинных программ. Любая из них запускается по указанию пользователя или прикладной программы в виде команды. Эти машинные программы обычно записаны в постоянном запоминающем устройстве (ПЗУ) компьютера. *Часть их, предназначенная для выполнения операций обмена информацией между устройствами компьютера, носит название базовой системы ввода/вывода (BIOS).*

Операционная система и языки программирования высокого уровня имеют свои стандартные наборы команд, с помощью которых общаются с компьютером. В операционной системе для общения пользователя с компьютером используется пользовательский интерфейс – командный процессор или оболочка операционной системы. В MS-DOS он хранился в файле COMMAND.COM. Оболочка – это та часть MS-DOS, которая обеспечивает работу системы в режиме командной строки и возможность общения пользователя с компьютером. Большая часть команд MS-DOS (более 20) непосредственно была встроена в COMMAND.COM (их можно было увидеть при просмотре этого файла на экране), например, для копирования файлов – *COPY*, для просмотра оглавления диска – *DIR* и т. д. Другая часть оформлена в виде самостоятельных программ-утилит: для форматирования дисков – *FORMAT*, для поиска файлов – *FIND*, для проверки жесткого диска и увеличения его свободного пространства – *CHKDSK* и др.

Графический пользовательский интерфейс Windows, OS/2 и других операционных систем имеет свои наборы команд для общения программ и пользователя с компьютером. Эти команды, как и языки программирования высокого уровня, не привязаны непосредственно к конкретной аппаратуре, к конкретному типу компьютера, т. е. они *машинно-независимы*. Они не могут управлять непосредственно устройствами компьютера. Поэтому *работу пользователя в среде операционной системы и программирование на языках высокого уровня, когда управление устройствами и обращение к аппаратным средствам выполняется косвенно, называют работой на логическом уровне*. Тем не менее, часто приложения, написанные на языках программирования высокого уровня, включают в себя подпрограммы, работающие непосредственно с аппаратурой. Так, компьютерные игры очень часто используют непосредственный доступ к видеоадаптерам и аудиоустройствам с целью ускорения работы с графикой и музыкальными файлами. Кроме того, существуют задачи, для решения которых программы просто обязаны обращаться непосредственно к аппаратуре. Например, программы низкоуровневого форматирования дисков должны напрямую взаимодействовать с адаптерами дисков. Другим примером являются менеджеры памяти, позволяющие управлять всей физически имеющейся в компьютере оперативной памятью.

Обычно выполнение своих команд операционная система осуществляет с помощью машинных программ, записанных в ПЗУ. Следовательно, имея посредника в виде системной *BIOS* и стандартный интерфейс обращения к ней, операционная система, как и прикладные программы, оказывается независимой, отделенной от аппаратных средств и может работать на компьютерах любой конфигурации. *Именно ПЗУ BIOS обеспечивает совместимость между аппаратными средствами и операционной системой*. Для обеспечения работы операционной системы в разных моделях компьютеров устанавливаются разные микросхемы ПЗУ, содержащие машинные коды программ работы с конкретным оборудованием. Коды, записанные в ПЗУ разных типов компьютеров, естественно, разные, поэтому ПЗУ *BIOS*, разработанное для одной системы, нельзя устанавливать в другие, так как некоторые устройства или вообще не будут работать, или будут работать некорректно.

С учетом сказанного, упрощенно иерархические уровни работы в IBM-совместимых компьютерах, можно представить в виде схемы (рис. 26). Из рисунка видно, что не только программы пользователя, но и сама операционная система может обращаться непосредственно к аппаратным средствам.



Рис. 26. Уровни работы с ПК

Операционная система подключается к *BIOS*, и часть ее становится расширением *BIOS*, за счет чего обеспечивается большой набор услуг для использования другими программами. Кроме того, операционная система яв-

ляется сама посредником между *BIOS* и прикладными программами на языке высокого уровня. Программисту не нужно каждый раз «изобретать велосипед», так как операционная система предоставляет в его распоряжение множество дополнительных служебных программ помимо тех, что включены в системную *BIOS*. Стандартными процедурами обработки файлов, включенными в состав ОС, может воспользоваться любая программа. В операционных системах Windows введен запрет на прямое обращение к аппаратным средствам, минуя ПЗУ *BIOS* и строенные возможности самой операционной системы. Программы, не соответствующие этим правилам, должны быть переписаны, в противном случае они не смогут работать в среде Windows.

В идеальном случае прикладные программы должны быть изолированы от аппаратных средств всегда посредством BIOS и ОС.

Раздел 3. ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

Лекция 21

Понятие о телеобработке и вычислительных сетях

Как показывает практика, совместное использование вычислительных ресурсов повышает надежность работы системы в целом, значительно (в 1,5–2 раза) сокращает эксплуатационные расходы, составляющие основную долю расходов на обработку информации, позволяет рационально сочетать преимущества централизованной и децентрализованной обработки информации благодаря приближению средств сбора исходной и выдачи результатной информации непосредственно к местам ее возникновения и потребления. Поэтому уже в 70-е гг. прошлого столетия предпринимались усилия к рассредоточенной обработке информации.

В настоящее время вычислительные сети представляют собой высшую организационную форму применения ЭВМ.

Для современных вычислительных сетей характерно:

- применение средств приема-передачи данных и каналов связи для организации обмена информацией в процессе взаимодействия средств ВТ;
- объединение многих достаточно далеко удаленных друг от друга ЭВМ или отдельных вычислительных систем в единую распределенную систему обработки данных;
- наличие широкого спектра периферийного оборудования, используемого в виде абонентских пунктов и терминалов пользователей, подключаемых к узлам сети передачи данных;

- использование унифицированных способов сопряжения технических средств и каналов связи, облегчающих процедуру наращивания и замену оборудования;

- наличие операционной системы, обеспечивающей надежное и эффективное применение технических и программных средств в процессе решения задач пользователей вычислительной сети.

Особенностью эксплуатации вычислительных сетей является не только приближение аппаратных средств непосредственно к местам возникновения и использования данных, но и разделение функций обработки и управления на отдельные составляющие с целью их эффективного распределения между несколькими ЭВМ, а также обеспечение надежного и быстрого доступа пользователей к вычислительным и информационным ресурсам и организация коллективного использования этих ресурсов.

Возможность концентрации в вычислительных сетях больших объемов данных, общедоступность этих данных, а также программных и аппаратных средств обработки и высокая надежность их функционирования – все это позволяет улучшить информационное обслуживание пользователей и резко повысить эффективность применения ВТ.

В условиях вычислительной сети предусмотрена возможность:

- создавать распределенные базы данных, размещаемые в памяти различных ЭВМ;

- специализировать отдельные ЭВМ (группы ЭВМ) для эффективного решения определенных классов задач;

- организовать параллельную обработку данных многими ЭВМ;

- автоматизировать обмен информацией и программами между отдельными ЭВМ и пользователями сети;

- стабилизировать и повышать уровень загрузки ЭВМ и дорогостоящего периферийного оборудования;

- резервировать вычислительные мощности и средства передачи данных на случай выхода из строя отдельных из них с целью быстрого восстановления нормальной работы сети;

- перераспределять вычислительные мощности между пользователями сети в зависимости от изменения их потребностей и сложности решаемых задач;

- сочетать работу в широком диапазоне режимов: диалоговом, пакетном, режимах «запрос-ответ», а также сбора, передачи и обмена информацией.

Рассмотрение вычислительных сетей начнем с локальных (корпоративных) сетей, создаваемых в пределах одного помещения или одной организации.

Локальная сеть (LAN – Local Area Network), или локальная вычислительная сеть (ЛВС) – это сочетание компьютеров, кабелей, плат сетевых адаптеров, сетевой операционной системы и сетевых прикладных программ. Локальные сети называются так потому, что сетевые адаптеры и прочие устройства не могут передавать сообщения на расстояния больше нескольких сотен метров. Локальная сеть позволяет организовать совместное использование файлов, программ-приложений, принтеров, модемов, электронной почты и диско-

вого пространства, работать с многопользовательскими программами и т. д. Локальные сети подразделяются на **равноправные** и **сети с сервером**. В равноправной сети все ресурсы имеют одинаковый статус. Эти сети в настоящее время встречаются редко. В локальной сети с сервером каждый персональный компьютер называется *рабочей станцией*, за исключением одного или нескольких, на которые возложено выполнение функций *серверов*. **Вообще сервер – это совместно используемый в сети ресурс**. В зависимости от назначения сервера он может быть специализированным *файл-сервером*, хранящим множество файлов, используемых или обрабатываемых рабочими станциями, *принт-сервером* для распечатки данных с рабочих станций, *коммуникационным, или Web-сервером* для обмена информацией между удаленными рабочими станциями или выхода их в Интернет, *сервером приложений*, – *совместно используемым в сети программным продуктом, например, программы управления гетерогенными, разноплатформенными базами данных* и т. д.

Сервер – наиболее сложный и дорогостоящий программно-аппаратный комплекс корпоративной сети, централизованный информационно-вычислительный ресурс, предназначенный для коллективного использования и обеспечивающий возможности совместной работы пользователей сети. Сервер, в отличие от рабочих станций, которые должны быть снабжены максимально развитыми средствами ввода и отображения информации, представляет собой компьютер с оборудованием и программным обеспечением для обслуживания всей сети. В первую очередь, он обеспечивает доступ рабочих станций к файлам, хранящимся на его жестких дисках. К серверу подключаются принтеры, модемы, общие приложения (например, электронная почта), факсы и другое оборудование. В качестве серверов используются особо надежные, высокопроизводительные компьютеры, так как при обслуживании всей сети они выполняют намного больший объем работы, чем рабочие станции. Многие серверы стоят в несколько раз дороже обычных компьютеров. Для них очень важно иметь большой объем оперативной и дисковой памяти, обеспечение исключительной надежности, высокой производительности ввода/вывода, дублирование устройств и хранимых данных. Они должны иметь средства контроля за состоянием сервера, средства бесперебойного питания и т. д. Емкость и быстродействие жесткого диска в сервере должны быть максимальными. Но не всегда установка одного диска большой емкости бывает наилучшим решением. Если сервер должен одновременно отвечать на запросы многих пользователей, то лучше установить в нем, например, 5 *SCSI* дисков емкостью по 120 Гбайт каждый, чем один диск емкостью 600 Гбайт. В сервере должно быть достаточное количество слотов для установки как необходимых на данный момент адаптеров, так и тех, что могут понадобиться в дальнейшем. Обычно роль серверов выполняют специализированные компьютеры.

В последнее время наибольшее распространение получают *рэковые* конструкции (*Rack* – стойка) серверов по следующим причинам:

- часто это является наиболее эффективным решением для крупных центров обработки данных, где требуется размещать большое количество серверов

на ограниченной площади и где наблюдается тенденция отделения систем внешней памяти от систем обработки данных. Внешняя память должна развиваться независимо от процессорной обработки и представлять собой самостоятельный ресурс, доступный множеству процессоров. В этих случаях собственная внешняя память сервера может иметь всего несколько дисков для размещения операционной системы, служебных файлов или программных приложений;

- серверы в рэке значительно удобнее обслуживать, можно сэкономить на оборудовании, используя с помощью переключателя всего один монитор и одну клавиатуру для всех серверов рэка;

- в последние годы быстрыми темпами развивается рынок поставщиков услуг Интернета и приложений. Аналитики предсказывают, что корпорации будут предпочитать аренду этих услуг покупке своего оборудования и программного обеспечения. Этот растущий рынок *провайдеров* (поставщиков услуг) почти целиком ориентируется на использование рэковых моделей серверов.

На рынке серверов универсального типа наиболее распространены устройства, создаваемые в архитектуре *Intel (IA)*. Ими, в частности, оснащена информационная система ОАО «РЖД», включающая центральное ядро и 17 ИВЦ дорог, а также 17 ИВЦ железных дорог в странах СНГ и Балтии. 75 % серверов *Intel*-архитектуры выпускают четыре фирмы: *Compaq Computer*, *IBM*, *Dell Systems* и *Hewlett-Packard*. Сама фирма *Intel* не выпускает серверы под своей торговой маркой, но она производит «строительные блоки» в виде серверных компонент. Основная стратегия *Intel* заключается в том, что компании-сборщики серверов на основе этих блоков изготавливают свои собственные изделия, подбирая то, что наиболее подходит заказчику с точки зрения характеристик конечного изделия.

Важнейший параметр сервера – его производительность – напрямую зависит от специфики решаемых задач. Например, установка сервера начального уровня обеспечит нормальную работу рабочей группы, а это по меркам зарубежных менеджеров по информационным технологиям – от 50 до 300 человек. Если сервер будет использоваться исключительно как файловый, то в типовой конфигурации он сможет нормально обслуживать всю рабочую группу (300 человек). Если серверу добавить функции поддержки электронной почты, то размер обслуживаемой рабочей группы надо уменьшить до 100–150 человек. Если же сервер будет выполнять еще и функции *Web*-сервера (для сети *Интернет* или *IntraNet*), то возможности сервера по обслуживанию рабочей группы окажутся еще меньше. Если же сервер выполняет еще и функции *брандмауэра*, который отделяет сеть предприятия от внешней сети с целью ее защиты, то количество активных рабочих станций сокращается до минимума.

В каждой рабочей станции и сервере установлены платы сетевых адаптеров (карт). Все рабочие станции и серверы соединяются между собой с помощью кабелей. Адаптеры должны соответствовать типу системной шины компьютера, производительности и сетевому протоколу конкретной сети. Что касается младших моделей серверов, то подключение стандартного сетевого адаптера в условиях стандартных задач не оказывает существенного влияния на

производительность сервера. Но если сервер высокого уровня обслуживает транзакционные задачи (транзакция – прием порции данных от пользователя, ее обработка и выдача ответного сообщения, например, продажа билетов на поезд), а установлен в него тот же самый простой сетевой адаптер, будет наблюдаться существенная деградация производительности сервера. Учитывая, что сетевая карта периодически обращается к процессору, то, например, в четырех процессорном сервере из-за обращений сетевой карты будет постоянно нарушаться вычислительная работа всех четырех процессоров. В данном случае следует применять более сложную сетевую карту, которая берет на себя функции обработки сетевого трафика (потока сообщений в сети), освобождая тем самым ресурсы центрального процессора для счетной работы.

Ряд производителей, в частности *Intel*, *3Com*, *SMC Networks* и др., поддерживают в своих сетевых адаптерах технологии, способные объединять несколько сетевых карт в единое *виртуальное* устройство с агрегированной пропускной способностью. Так, в сервер *Intel*-архитектуры можно установить до четырех сетевых адаптеров *Intel Server PRO/100+* с быстродействием 100 Мбит/с каждый и получить канал связи сервера с сетью, имеющий пропускную способность в дуплексном режиме 800 Мбит/с. *Следует подчеркнуть, что в сетевых картах, разработанных корпорацией Intel, нагрузка на центральный процессор сервера минимизирована.*

Сети, в которых часть компьютеров выполняет функции серверов, являются *неравноправными* и называются сетями типа *клиент/сервер*. В дополнение к своей операционной системе на каждой рабочей станции устанавливается сетевая программа-клиент, которая позволяет ей взаимодействовать с серверами. В свою очередь, серверы работают под управлением сетевой программы-сервера (например, *Novell NetWare*), которая обеспечивает их взаимодействие с рабочими станциями и позволяет последним обращаться к файлам, хранящимся в сервере.

Технология обработки информации в сети по схеме клиент–сервер является в настоящее время во всем мире одной из передовых (составляет 88 % успешно используемых технологий).

Для создания ответвлений от магистрального кабеля к компьютеру используют специальные согласующие устройства. Кроме того, могут понадобиться и другие специфичные узлы и блоки: распределительные щитки, трассовые усилители, коммутационные устройства и т. д. Прокладку линий связи, нарезку и монтаж кабелей и разъемов должны выполнять профессионалы, поскольку от качества выполнения этих работ зависит согласование импедансов линий. Если оно плохое, в сети появятся отраженные сигналы, а это неизбежно приведет к сбоям при передаче данных.

Сетевые адаптеры принимают и передают сообщения, а по кабелям эти сообщения распространяются. Однако набор соединенных между собой компьютеров становится сетью только тогда, когда установлены *протоколы обмена сообщениями* различных уровней. *Протокол – это правила взаимодействия компьютеров.* Описать в одном протоколе все правила невозможно. Поэтому

сетевые протоколы строят *многоуровневыми*. Например, на нижнем уровне описываются правила передачи небольших порций (*пакетов*) информации с одного компьютера на другой, в соответствии с которыми *маршрутизаторы* (*routers*)^{*} сети автоматически выбирают оптимальный маршрут для доставки пакетов данных до места назначения. Если какая-то часть была искажена помехами при передаче, то на этом уровне запрашивается повтор передачи только искаженной части информации. Протокол следующего уровня описывает, как большие массивы данных разбить на небольшие части (*пакеты*) для передачи и собрать их снова на приемном конце. При этом *пакеты* передаются по протоколу нижнего уровня.

Суммарная пропускная способность сети в значительной степени зависит от кабеля, используемого для соединения элементов сети, и может составлять от 10 Мбит/с до 100 Мбит/с. Кабели на витой паре легко обеспечивают пропускную способность 10 Мбит/с. Современные кабели на витой паре категории 5 позволяют обеспечить пропускную способность 100 Мбит/с, однако для этого нужны более дорогие сетевые адаптеры и концентраторы. *Концентратор*, или *хаб*, имеет несколько портов – гнезд для подключения сетевых кабелей, посредством которых он соединяется с сетевыми адаптерами компьютеров или иными сетевыми устройствами.

Для обеспечения передачи данных на значительные расстояния и для более быстрой передачи данных применяются линии связи на волоконно-оптическом кабеле. Такие сети, как правило, относятся к распределенным региональным или глобальным сетям, хотя могут быть и корпоративными, как сеть железной дороги.

Сети объединяют все большее количество компьютеров (некоторые локальные сети состоят из нескольких тысяч компьютеров), поэтому для безопасности и удобства работы в них сети программным образом (на логическом уровне) с помощью коммутаторов стали делить на отдельные сегменты – *виртуальные компьютерные сети*, аналогично делению одного физического жесткого магнитного диска на несколько логических. Организуется как бы ряд каналов доступа только к строго определенному числу компьютеров в сети.

Широкое использование компьютерных сетей в современной электронной экономике привело к тому, что на крупных предприятиях, в корпорациях, транспортных организациях, таких, например, как федеральные железные дороги, количество информационных сетей, взаимодействующих между собой, резко возрастает. Например, в корпоративной сети Свердловской железной дороги уже сейчас находится в эксплуатации более 350 локальных сетей. В условиях рынка требуется, чтобы информационные системы работали надежно и обеспечивали высокий уровень безопасности. Все приложения, особенно связанные с электронным бизнесом, должны быть доступны 24 часа в день и 365 дней в году. В каждой локальной сети имеется не по одному серверу, кроме того, эти се-

* **Router**, или маршрутизатор, – программно-аппаратный комплекс, физически соединяющий компьютерные сети друг с другом и с помощью специального программного обеспечения передающий пакеты информации из одной сети в другую.

ти структурно неоднородны, построены на различных программно-аппаратных платформах и имеют каждая свои средства управления. *Остро встает проблема централизованного системного управления ресурсами сетей* (не содержать же для каждой сети несколько администраторов!) и снижения эксплуатационных расходов. Для резкого снижения эксплуатационных затрат необходимы аппаратные и программные средства централизованного управления, что позволит предприятиям интегрировать информационные сети и управление бизнес-процессами. Такие продукты начали появляться на рынке информационных технологий, например, *OpenView* от *Hewlett-Packard*, *Unicenter* от *Computer Associates* или *NetView* от *Tivoli* (подразделение фирмы *IBM*).

Лекция 22

Классификация сетей. Способы коммутации и передачи данных

Вычислительные сети классифицируются по различным признакам. Сети, состоящие из программно-совместимых ЭВМ, являются *гомогенными*, или *однородными*. Если ЭВМ, входящие в сеть, программно несовместимы, то такая сеть называется *неоднородной* или *гетерогенной*.

По типу организации передачи данных различают сети:

- ✓ с коммутацией каналов;
- ✓ с коммутацией сообщений;
- ✓ с коммутацией пакетов.

Иногда сети используют смешанные системы передачи данных.

По характеру реализуемых функций выделяют следующие сети:

- ✓ *вычислительные*, предназначенные для решения задач, требующих вычислительной обработки исходной информации;
- ✓ *информационные*, предназначенные для получения справочных данных по запросам потребителей информации;
- ✓ *смешанные*, в которых реализуются вычислительные и информационные функции.

По способу управления вычислительные сети делятся на сети с *децентрализованным*, *централизованным* и *смешанным* управлением. В первом случае каждая ЭВМ, входящая в состав сети, включает полный набор программных средств для координации выполняемых сетевых операций. Сети такого типа сложны и дороги, так как операционные системы отдельных ЭВМ разрабатываются с ориентацией на коллективный доступ к общему полю памяти сети. При этом в каждый конкретный момент времени доступ к общему полю памяти предоставляется только для одной ЭВМ. А координация работы ЭВМ осуществляется под управлением **единой** операционной системы сети.

В условиях смешанных сетей под централизованным управлением ведется решение задач, обладающих высшим приоритетом и, как правило, связанных с обработкой больших объемов информации.

По структуре построения (топологии) сети подразделяются на *одноузловые* и *многоузловые*, *одноканальные* и *многоканальные*. Топология вычислительной сети во многом определяется структурой сети связи, т. е. способом соединения абонентов друг с другом и ЭВМ. Различают следующие топологии сетей: радиальная или звездообразная, кольцевая, многосвязная («каждый с каждым»), иерархическая, «общая шина» и другие, рис. 27.

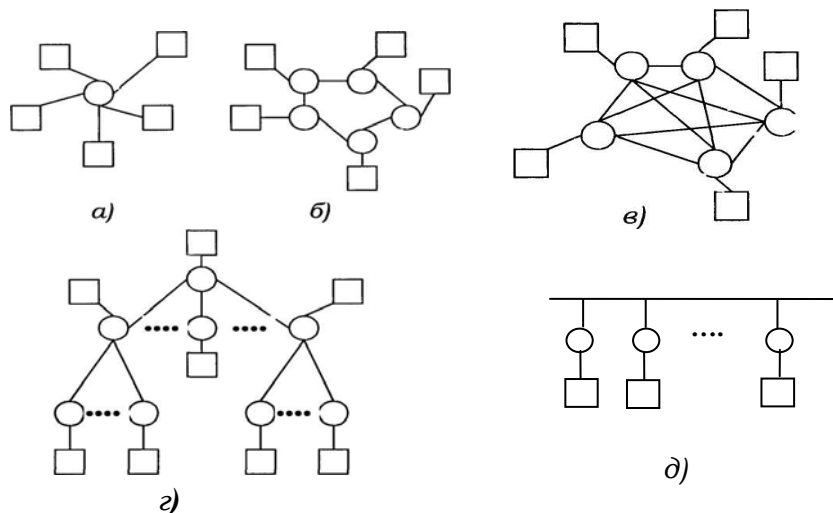


Рис. 27. Основные типы структур построения сетей ЭВМ

В сети *с общей шиной* одна из машин служит в качестве системного обслуживающего устройства, обеспечивающего централизованный доступ к общим файлам и базам данных, печатающим устройствам и другим вычислительным ресурсам. Сети данного типа приобрели большую популярность благодаря низкой стоимости, высокой гибкости и скорости передачи данных, легкости расширения сети (подключение новых абонентов к сети не сказывается на ее основных характеристиках). К недостаткам шинной топологии следует отнести *необходимость использования довольно сложных протоколов и уязвимость в отношении физических повреждений кабеля*.

Кольцевая топология характеризуется тем, что информация по кольцу может передаваться только в одном направлении, и все подключенные компьютеры могут участвовать в ее приеме и передаче. При этом абонент-получатель должен пометить полученную информацию специальным маркером, иначе могут появиться «заблудившиеся» данные, мешающие нормальной работе сети. *Кольцо особенно уязвимо в отношении отказов:* выход из строя какого-либо сегмента кабеля приводит к прекращению обслуживания всех пользователей (последовательная структура – ничего не поделаешь). Разработчики сетей приложили немало усилий, чтобы справиться с этой проблемой. Защита от повреждений или отказов обеспечивается либо замыканием кольца на обратный (дублирующий) путь, либо переключением на запасное кольцо. И в том, и в другом случае сохраняется общая кольцевая технология.

Иерархическая сеть (конфигурация типа «дерево») представляет собой более развитый вариант структуры, построенной на основе общей шины. Дерево

образуется путем соединения нескольких шин с корневой системой, где размещаются самые важные компоненты сети. Оно обладает необходимой гибкостью для того, чтобы охватить средствами сети несколько этажей в здании или несколько зданий на одной территории, и реализуется, как правило, в сложных системах, насчитывающих десятки и сотни абонентов.

Радиальную, или звездообразную, конфигурацию можно рассматривать как дальнейшее развитие структуры «дерево с корнем» с ответвлением к каждому подключенному устройству. В центре сети обычно размещается коммутирующее устройство, обеспечивающее жизнеспособность системы (рис. 28). Сети подобной конфигурации находят наиболее частое применение в автоматизированных учрежденческих системах управления, использующих центральную базу данных. Звездообразные сети, как правило, менее надежны, чем сети с общей шиной или иерархические, но путем дублирования оборудования центрального узла эта проблема решается просто.

К недостаткам можно отнести значительное потребление кабеля (иногда в несколько раз превышающее расход в аналогичных по функциональным возможностям сетях с общей шиной или иерархических). Наиболее сложной и дорогой является

многосвязная топология, в которой каждый узел связан со всеми другими узлами сети. Эта топология применяется очень редко, в основном там, где требуются исключительно высокие надежность сети и скорость передачи данных. На практике чаще встречаются *гибридные* сети, приспособленные к требованиям конкретного заказчика и сочетающие фрагменты нескольких топологий.

В зависимости от используемых *систем передачи данных и средств вычислительной техники* локальные сети подразделяются на четыре группы.

К *первой группе* относятся сети, ориентированные на массового пользователя. Такие сети объединяют в основном персональные компьютеры с помощью недорогих систем передачи данных, обеспечивающих передачу информации на расстояние 100–500 м со скоростью 2 400 – 19 200 бод^{*}.

Ко *второй группе* относятся сети, объединяющие, кроме персональных компьютеров, микропроцессорную технику, встроенную в технологическое оборудование (кассовые аппараты, средства автоматизации проектирования, обработки документальной информации и др.), а также средства электронной почты. Система передачи данных таких сетей обеспечивает передачу информации на расстояние до 1 км со скоростью от 19 200 бод до 1 Мбод. Стоимость

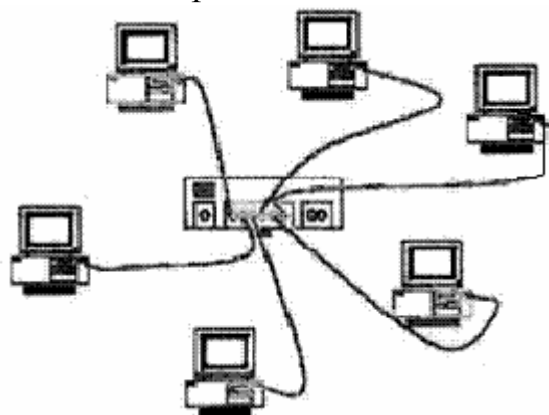


Рис. 28. Подключение рабочих станций к распределительному устройству (хабу)

* Бод – скорость передачи данных по каналу связи, равная количеству изменений *состояния модулированного сигнала*, используемого для передачи данных. Если, например, частота или фаза изменяются 1000 раз в секунду, то скорость передачи равна 1000 бод.

передачи данных в таких сетях примерно на 30 % превышает стоимость передачи в сетях первой группы.

К *третьей группе* относятся сети, объединяющие персональные компьютеры, мини-ЭВМ и ЭВМ среднего класса. Эти сети используются для организации управления сложными производственными процессами с применением робототехнических комплексов и гибких автоматизированных модулей, а также для создания крупных систем автоматизации проектирования, систем управления научными исследованиями и т. д. Системы передачи данных в таких сетях имеют среднюю стоимость и обеспечивают передачу информации на расстояние до нескольких километров со скоростью 120 Мбод.

Для *четвертой группы* характерно объединение в своем составе всех классов ЭВМ. Такие сети применяются в сложных системах управления крупными производственными объединениями и даже отдельной отраслью. Они включают в себя основные элементы всех предыдущих групп. В рамках данной группы сетей могут применяться различные системы передачи данных, в том числе обеспечивающие передачу информации со скоростью от 10 до 50 Мбод на расстояние до 10 км. В своем составе они могут содержать разветвленную сеть соединений между различными абонентами, по функциональным возможностям сети этой группы мало, чем отличаются от региональных сетей, обслуживающих крупные города, районы и области.

Способы коммутации и передачи данных. Основная функция систем передачи данных в условиях функционирования вычислительных сетей заключается в организации *быстрой и надежной передачи информации* произвольным абонентам сети, а также в *сокращении затрат* на передачу данных. Небывало быстрое развитие электроники и микропроцессорной техники позволили затраты на техническое обеспечение вычислительных сетей сократить за последние 10–15 лет примерно в **десять раз**, а затраты на организацию и эксплуатацию каналов связи сократились только **вдвое**. Важнейшая характеристика сетей передачи данных – *время доставки информации* – зависит от структуры сети передачи данных, пропускной способности линий связи, а также от способов соединения каналов связи между взаимодействующими абонентами сети и способ передачи данных по этим каналам. В настоящее время различают системы передачи данных с постоянным включением каналов связи (некоммутируемые каналы связи) и коммутацией на время передачи информации по этим каналам.

При использовании некоммутируемых каналов связи средства приема-передачи абонентских пунктов и ЭВМ постоянно соединены между собой, т. е. находятся в режиме *«on-line»*. В этом случае отсутствуют потери времени на коммутацию, обеспечивается высокая степень готовности системы к передаче информации, более высокая надежность каналов связи и, как следствие, достоверность передачи информации. Недостатками такого способа организации связи являются низкий коэффициент использования аппаратуры и линий связи, высокие расходы на эксплуатацию сети. Рентабельность подобных сетей достигается только при условии полной загрузки этих каналов.

При коммутации абонентских пунктов и ЭВМ только на время передачи информации, т. е. в режиме *«off-line»*, принцип построения узла коммутации

определяется способами организации прохождения информации в сетях передачи данных.

Передача информации в сети между абонентами осуществляется многоуровневыми протоколами обмена, что позволяет максимально эффективно организовать взаимодействие абонентов сети. Как правило, обмен информацией между абонентами сети на нижнем уровне осуществляется с помощью фиксированных блоков (фрагментов) информации, которые называют *пакетами* или *кадрами*. При коммутации пакетов перед началом передачи сообщение разбивается на короткие пакеты фиксированной длины, которые затем передаются по сети. Сетевой адаптер и его управляющая программа ориентированы на передачу и прием именно пакетов. В пункте назначения эти пакеты вновь объединяются в первоначальное сообщение, а так как их длительное хранение в запоминающем устройстве узла связи не предполагается, то пакеты передаются от узла к узлу с минимальной задержкой во времени. При коммутации пакетов их фиксированная длина обеспечивает эффективность обработки пакетов, предотвращает блокировку линий связи и значительно уменьшает емкость требуемой промежуточной памяти узлов связи. Кроме того, сокращается время задержки при передаче информации, т. е. скорость передачи превышает аналогичную скорость при коммутации сообщений целиком.

Формат типичного пакета (кадра) приведен на рис. 29. Все поля вместе об-

Адрес отправителя	Адрес получателя	Тип кадра	Сообщение (данные)	Контрольная сумма (CRC)
----------------------	---------------------	-----------	-----------------------	----------------------------

Рис. 29. Типичная структура кадра информации

разуют оболочку сообщения – «конверт», в который упаковывается передаваемая информация. Структуры кадров в различных сетях могут быть разными, но некоторые разделы присутствуют во всех вариантах. К ним относятся сетевой адрес отправителя, сетевой адрес получателя, идентификатор содержимого кадра, данные или сообщение и контрольная сумма или код *CRC* (*Cyclic Redundancy Check*) – *циклический избыточный контроль* для обнаружения возможных ошибок при передаче. Кадры сообщений могут быть переданы:

- для открытия сеанса связи с другим адаптером;
- при передаче данных, например, файла;
- для подтверждения приема кадра данных;
- при посылке общего сообщения всем адаптерам;
- при закрытии сеанса связи.
- для проверки работы сети и т. д.

Вложение кадров. Структура кадра также может быть разбита на несколько уровней (рис. 30). На нижнем уровне определяется, как заставить сетевой адаптер передать сообщение, но в нем нет таких понятий, как файл-сервер или перенаправление файлов. На высшем уровне, напротив, оперируют этими понятиями, но совершенно не интересуются устройством и даже названием сети.

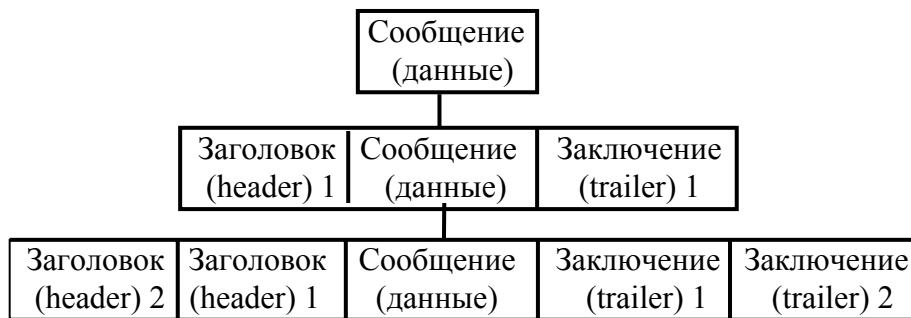
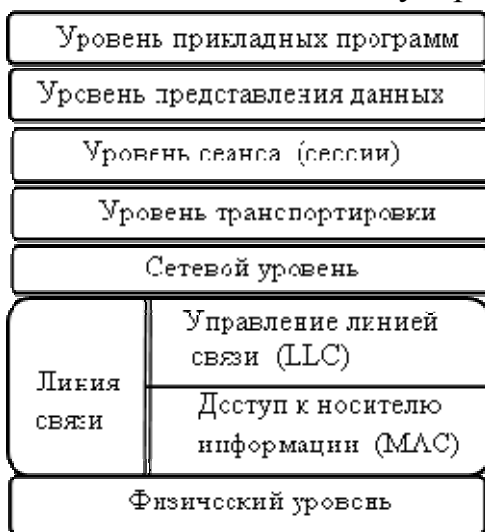


Рис. 30. Многоуровневая структура кадра

Различные фирмы-производители сетей по-разному разбивают на этапы процедуру передачи сообщений, но всегда руководствуются при этом *моделью OSI*.

Модель OSI. Международной организацией по стандартам (*ISO*) разработа-



ботана эталонная модель *OSI (Open System Interconnection)* взаимодействия открытых информационных сетей (рис. 31), принятая в качестве международного стандарта. В коммуникационных протоколах модели *OSI* предусмотрено семь уровней, характеризующих любую систему связи и взаимодействующих на строго иерархической основе по принципу «снизу вверх». Пока что в большинстве сетевых операционных систем используется не более четырех уровней. Каждый уровень должен быть «самодостаточен» и связан с остальными вполне определенным интерфейсом. Кратко рассмотрим каждый из уровней модели.

Рис. 31. Эталонная модель *OSI*

Физический. Осуществляет сопряжение с каналом. На этом уровне определены физические и электрические параметры коммуникационных устройств сети (разъемов, витых пар, волоконно-оптических и коаксиальных кабелей, буферных усилителей и т. д.). Это чисто аппаратный уровень. Несмотря на то, что операции на остальных уровнях могут быть реализованы как с помощью специализированных интегральных схем, так и программно, все же они считаются программными по отношению к физическому уровню.

Уровень линии связи. Осуществляет управление передачей информации по каналу. На этом уровне определяются те электрические сигналы (импульсы), которые поступают в сеть или принимаются из сети. Здесь и только здесь формируются структура и способы электрического представления передаваемой информации (последовательности битов, способы кодирования, эстафеты). На этом же этапе обнаруживаются и корректируются ошибки (организуются запросы на повторную передачу ошибочно переданной информации). Поскольку этот уровень достаточно сложен, его часто разделяют на две ступени: управления доступом к носителю информации (*MAC – Media Access Control*) и управ-

ления линией связи (*LLC – Logical Link Control*). Процедуры подуровня *MAC* касаются непосредственной организации доступа к сети (передача эстафеты, контроль за столкновениями) и управления сетью. На подуровне *LLC*, более высоком, чем *MAC*, происходят передача и прием сообщений пользователей.

Управление каналом (линией связи) включает:

- генерацию стартового канала и организацию начала передачи информации;
- передачу информации по каналу;
- проверку получаемой информации и исправление ошибок;
- отключение канала при его неисправности и восстановление передачи после его ремонта;
- генерацию сигнала окончания передачи и перевод канала в пассивное состояние.

Сетевой уровень. На этом уровне осуществляется адресация и определение маршрутов прохождения пакетов данных. Процедуры этого уровня «прокладывают» путь информации между системой-отправителем и системой-получателем и ответственны за «доставку» сообщений по необходимым адресам.

Уровень транспортировки. Реализуемые на этом уровне функции управляют процессом передачи информации по пути, определенном на предыдущем уровне. Если в сети пересылается одновременно более одного пакета, процедуры этого уровня управляют очередностью их передачи, да и вообще движением информации в сети. Одной из важных операций является поиск и уничтожение «двойников» – дублирующих друг друга пакетов.

Сессионный уровень. Реализованные на этом уровне функции позволяют скоординировать работу приложений, выполняющихся на двух рабочих станциях, в единый процесс {*сессию или сеанс*), т. е. организовать очень сложный по своей структуре диалог. На этом уровне формируется сеанс, происходит управление пересылкой пакетов, имеющих к нему отношение, и организуется завершение сессии.

Уровень представления данных. Если в сети совместно должны работать компьютеры разных типов (например, фирм *IBM, Apple, DEC* и др.) то, как правило, приходится прибегать к преобразованию данных в форматы, принятые в этих компьютерах. Процедуры этого уровня предназначены для таких преобразований.

Уровень прикладных программ. Именно этот уровень в модели *OSI* «видит» прикладная программа. Здесь сообщение, которое должно быть передано по сети, «входит» в модель, «спускается» вниз до физического уровня, передается на другую рабочую станцию, а затем «поднимается» по уровням вверх до тех пор, пока не дойдет до программы-приложения на другом компьютере.

Последние три уровня непосредственно связаны с организацией взаимодействия прикладных программ пользователей, а также с вводом, хранением, обработкой данных и выдачей результатов. Все процессы, проходящие на этих уровнях, носят название прикладных. Это главные процессы в коммуникационных системах. Именно ради них создаются сети, в том числе и локальные.

Каждый из уровней модели *OSI* выполняет указания уровня, расположенного над ним. Так, физический уровень обслуживает каналный уро-

вень, который принимает распоряжения сетевого уровня и т. д. В результате прикладной уровень использует сервис всех остальных уровней процессов взаимодействия. Однако уровни работают так, чтобы в нужных случаях можно было проверить работу других уровней. Так, если канальный уровень случайно пропустит ошибку, появившуюся при передаче информации, то ее определит и исправит транспортный уровень.

Одной из причин, по которой многие сетевые операционные системы оказываются чисто «фирменными» (в отличие от систем с *открытой архитектурой*) является то, что в них не выполняются требования международного стандарта *OSI*.

Лекция 23

Методы доступа в ЛВС

В большинстве сетей существует единственный канал связи. С технической точки зрения это означает, что для передачи данных не используется какая-либо разновидность модулированных сигналов, что позволяет обычно организовать в одной линии несколько независимых каналов. Поэтому в локальной сети *в каждый конкретный момент времени друг с другом могут взаимодействовать только два сетевых адаптера*.

Существует два принципиально отличающихся друг от друга типа локальных сетей: *с обнаружением столкновений* и *с передачей эстафеты*. В институте инженеров по электротехнике и электронике *IEEE (Institute of Electrical and Electronic Engineers)* разработан ряд стандартов на характеристики и параметры сигналов в сетях обоих типов. Эти стандарты известны как *IEEE 802.3* (сеть типа *Ethernet*) и *IEEE 802.5* (сеть типа *Token Ring*). Практически работающие сети всегда имеют некоторые отклонения от этих стандартов. Так, например, из множества сетей *Ethernet* полноценной сетью этого типа можно считать только сеть *100BaseT*, поскольку в ней структура кадров соответствует стандарту *IEEE 802.3*, а для доступа к каналу передачи данных используется протокол *CSMA/CD (Carrier sense multiple access and collision detection)* – с множественным доступом по опросу состояния канала и разрешением конфликтов.

Сети типа *Ethernet* относятся к самым распространенным сетям с обнаружением столкновений, к которым можно подключать самое разнообразное оборудование, включая компьютеры фирм *Apple, IBM, DEC* и их аналоги, компьютеры, работающие под управлением операционной системы *UNIX* и др.

Если в сети не происходит передач данных (нет запросов рабочих станций и ответов серверов), сигналы в ней отсутствуют. После того как рабочая станция передала свой запрос на передачу по кабелю сети, он освобождается для других посылок. Но что произойдет, если две или более рабочих станции попытаются использовать сеть одновременно? Возникает так называемое столкновение *{collision}*. Оба столкнувшихся абонента прекращают передачи, а через некоторое время предпринимают новую попытку. Обычно на выход из столкно-

вения система затрачивает менее одной микросекунды. Интервал времени, на который каждый из абонентов прекращает передачу, выбирается случайным образом. После перерыва один из них (любой) практически всегда начнет передавать свои посылки раньше, чем другой. Контролем за столкновениями и повторными передачами «ведает» плата сетевого адаптера, поэтому эти операции никак не сказываются на работе основной программы.

Многие считают, что причина плохой производительности сетей типа *Ethernet* заключается в том, что при большом количестве пользователей и интенсивном обмене в них возникает слишком много столкновений, поэтому сокращается время полезной работы. Несомненно, для сетей с очень большим количеством рабочих станций этот фактор существенен. Но, как показывает практика, более чем в 90 % случаев проблемы возникают не из-за столкновений, а из-за низкого качества кабельных соединений и сбоев сетевых адаптеров.

В сети *Ethernet* данные передаются во всех направлениях со скоростью от 10 до 100 Мбит/с. Все пакеты данных принимаются всеми компьютерами, но факт приема подтверждает только тот из них, которому данный пакет предназначен (при совпадении сетевого адреса компьютера и адреса, указанного в пакете).

Преамбула	Адрес получателя	Адрес отправителя	Тип пакета	Данные	CRC
8	6	6	2	46-1500	4

Рис. 32. Пакет, или «кадр», сети *Ethernet*

Преамбула. Это поле используется для синхронизации и формирования границ кадров. Первые семь байт преамбулы представляют двоичную последовательность 10101010, а последний байт – 10101011.

Адрес получателя. В этом поле записывается адрес принимающей рабочей станции. В соответствии с принятой системой адресации первые три байта – это адрес, назначаемый группе, а последние три байта – локальный адрес устройства в группе. У первого (левого) бита первого байта особый смысл. Если этот бит нулевой, то следующий далее адрес получателя является физическим адресом, уникальным в среде *Ethernet*. Если первый бит первого байта равен единице, то это означает, что кадр «широковещательный», т. е. предназначен для приема несколькими станциями. В этом случае оставшаяся часть адреса получателя может относиться либо к группе логически связанных между собой рабочих станций, либо ко всем станциям в сети (все биты единичные).

Адрес отправителя. По этому полю идентифицируется рабочая станция, передающая пакет данных. Первый бит первого байта всегда нулевой.

Тип кадра. В этом поле записываются два байта, идентифицирующие тип протокола высшего уровня, которым был сформирован (или которым должен быть обработан) данный пакет. Если это поле используется, то в сети можно одновременно работать с несколькими протоколами высокого уровня, причем сообщения, подготовленные в рамках одного протокола, не будут мешать обменам по другим протоколам.

Данные. Содержащаяся именно в этом поле информация должна быть передана адресату.

CRC. В этом поле записываются 4 байта остатка контрольной суммы, вычисленного по методу CRC-32. С целью обнаружения ошибки передачи при приеме рабочая станция снова по тому же методу вычисляет остаток контрольной суммы для пакета и сравнивает результат с переданным в поле CRC.

Сеть Token Ring. Одной из распространенных сетей с передачей эстафеты является сеть типа *Token Ring* (Token – эстафета).

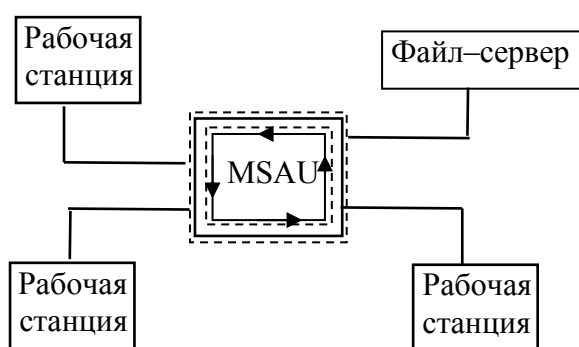


Рис. 33. Сеть с передачей эстафеты

Сеть с передачей эстафеты можно рассматривать как кольцо. *Топология сети может быть любой*, но кадры информации всегда передаются по кругу от одной рабочей станции к другой (рис. 33). Рабочая станция передает кадр в устройство много-станционного доступа MSAU, которое перенаправляет его в следующую станцию. В свободной сети рабочие станции передают друг другу «эстафетную палочку». Эстафета представляет собой трехбайтное сообще-

ние, означающее, что сеть свободна. Сеть становится занятой тогда, когда рабочая станция принимает эстафету и превращает ее в кадр данных, в котором содержится передаваемое станцией сообщение. Кадр переходит от адаптера к адаптеру до тех пор, пока не доберется до станции-получателя, которая принимает кадр для обработки и подтверждает его прием, вставляя в кадр строго определенные биты. После этого кадр данных продолжает свой путь по сети до тех пор, пока не доберется до своего отправителя. Станция-отправитель в случае правильного его приема освобождает сеть, снова превращает кадр в эстафету и запускает ее по кругу. Сеть с передачей эстафеты устроена так, что столкновения в ней исключены.

Отправитель кадра до тех пор, пока тот не обойдет всю сеть и не вернется назад, передает в сеть пустые символы. Типичная задержка в кольце даже со скоростью передачи данных всего лишь 4 Мбит/с. составляет от 50 до 100 символов, то есть ничтожно мала даже при 100 или 200 рабочих станциях. В сетях же со скоростью передачи свыше 16 Мбит/с. рабочая станция, являющаяся отправителем, запускает новую эстафету сразу после передачи своего кадра с сообщением. Последующие станции передают кадр дальше по сети, но сразу после передачи могут сами передавать свой кадр с новой эстафетой. Если бы можно было заглянуть в такую сеть, то можно было бы увидеть, как за каждым кадром информации бегут эстафеты, а следом за ними другие кадры данных вместо длинного хвоста пустых символов.

Определенным рабочим станциям и файл-серверам можно назначить приоритеты для того, чтобы они могли чаще получать доступ к сети.

Иногда некоторые станции ошибаются и теряют эстафету. Поэтому одна из рабочих станций обязательно выполняет роль *активного наблюдателя*, т. е.

она несет дополнительную ответственность за управление кольцом. Эта станция обеспечивает синхронизацию в кольце, при необходимости «запускает» новые эстафеты, а при определенных обстоятельствах формирует диагностические кадры. Активный наблюдатель выбирается при инициализации сети, и им может стать любая рабочая станция. Если по какой-либо причине она выходит из строя, ее функции передаются *резервному наблюдателю*.

За исключением волоконно-оптических систем, сети *Token Ring* являются самыми дорогими. Расходы на такую сеть оправдываются только при интенсивных обменах и большом количестве рабочих станций, когда количество столкновений в сети начинает превышать разумные пределы. Такие сети создаются в крупных фирмах, особенно в тех случаях, когда в качестве основных используются большие компьютеры (мейнфреймы).

Кадры *Token Ring*. В стандарте сети определены три формата пакетов сообщений: *эстафета, кадр информации и код прерывания*.

Эстафета, как уже говорилось, состоит из трех байт и содержит три поля по байту каждое. Поле *SD* – начальный ограничитель – располагается в начале эстафеты, а также в начале каждого кадра, передаваемого по сети. В этом поле передаются не только двоичные единицы и нули, но и уникальные последовательности электрических сигналов, характерные только для начального ограничителя, и которые нельзя ни с чем перепутать – четыре недвоичные посылки (длительность каждой соответствует передаче одного бита) и четыре обычных бита.

Следующее поле *AC* разделено на четыре подполя: *PPP T M RRR*, где *PPP* – биты приоритета, *T* – бит эстафеты, *M* – бит наблюдения и *RRR* – биты резервирования. Сетевой адаптер может назначить эстафете или кадру один из восьми приоритетов. Высший уровень приоритета – 7. Рабочая станция может превратить эстафету в кадр и войти в сеть только в том случае, если она приняла эстафету, приоритет которой ниже приоритета самой станции. Бит эстафеты *T* равен 1 для эстафеты и 0 – для кадра. Бит наблюдения *M* устанавливается в 1 станцией-наблюдателем и сбрасывается в нуль любой другой станцией, передающей эстафету или кадр. Если к активному наблюдателю пришла эстафета или кадр с единичным битом *M*, то это значит, что они прошли полный круг и не были обработаны ни одной рабочей станцией. Активный наблюдатель в этом случае считает, что произошел сбой в сети, прерывает передачу и запускает новую эстафету.

Биты резервирования используются совместно с битами приоритета. Рабочая станция резервирует за собой следующий доступ к сети, обозначив свой приоритет и задав соответствующие значения битам резервирования, если ее приоритет выше текущего значения этих битов. Когда рабочая станция запускает новую эстафету, она устанавливает биты приоритета в соответствии со значениями подполя *RRR* только что принятого кадра. Если в процесс передачи не вмешивается станция с более высоким приоритетом, чем у той, что задала значения битов резервирования, то она будет первой в очереди на превращение эстафеты в кадр.

Последнее поле *ED* - конечного ограничителя, как и поле *SD*, содержит уникальную комбинацию единиц и нулевых посылок, которую нельзя спутать ни с чем. Это поле передается в конце каждой эстафеты.

Кадр информации. На рис. 34 приведен второй формат пакета, передаваемого в сети, и длины его полей в байтах. Кадры информации могут содержать сообщения, которые сетевая операционная система или программа-приложение посылают другому компьютеру в сети. Кроме того, с их помощью иногда передаются внутренние сообщения, необходимые платам сетевых адаптеров для управления сетью.



Рис. 34. Кадр информации

Кадр информации состоит из следующих полей:

- последовательность начала кадра *SFS* – *Start Frame Sequence*;
- адрес получателя информации *DA* – *Destination Address*;
- адрес отправителя информации *SA* – *Source Address*;
- поле передаваемой информации;
- контрольная кадровая последовательность *FCS*–*Frame Check Sequence*;
- последовательность конца кадра *EFS* – *End Frame Sequence*.

Как уже говорилось, все поля вместе образуют оболочку сообщения – «конверт», в который упаковываются управляющие данные (*MAC-кадр*) или пользовательская информация (*LLC-кадр*). *MAC-кадры* предназначены для "внутреннего употребления" и используются сетевыми адаптерами для управления сетью. Каждый кадр начинается с последовательности начала кадра, содержащей три поля: начальный ограничитель (*SD*), управление доступом (*AC*). Эти два поля аналогичны тем, что используются в эстафете. В поле *FC* – управления кадром указывается тип кадра (*MAC* или *LLC*). Два первых бита, равные 00, указывают на *MAC-кадр*, а 01 – на *LLC-кадр*. Значения 10 и 11 зарезервированы на будущее. Следующие шесть бит указывают тип *MAC-кадра*:

000011 – объявление эстафеты. Если резервный наблюдатель посчитает, что активный наблюдатель вышел из строя, он начинает посылать кадры объявления эстафеты. После этого резервные наблюдатели начинают общаться между собой, выясняя, который из них станет новым активным наблюдателем;

000000 – тест дублирования адреса. Посылается рабочей станцией при ее подключении к сети, если она хочет убедиться в том, что ее адрес уникален;

000101 – присутствие активного наблюдателя. Довольно часто передается станцией-наблюдателем, чтобы сообщить другим рабочим станциям о своей работоспособности;

000110 – присутствие резервного наблюдателя. Передается станцией, имеющей статус резервного наблюдателя;

000010 – аварийный кадр. Передается при появлении серьезной неисправности, например, при обрыве кабеля или в том случае, если рабочая станция начала передачу, не дожидаясь эстафеты. Определив, какая рабочая станция передала аварийный кадр, диагностическая программа может обнаружить место, где находится неисправность;

000100 – очистка. Передается после инициализации сети и после определения нового активного наблюдателя.

Поле адреса получателя *DA* равно 2 или 6 байтам. Если оно равно 2 байтам, то по первому биту можно определить, является ли адрес групповым или индивидуальным. Если поле равно 6 байтам, то первый бит аналогично предыдущему определяет адрес индивидуальный или групповой. Второй бит означает систему адресации – локальную или глобальную (см. аналог в *Ethernet*). Остальные биты представляют собой адрес рабочей станции, которой предназначен кадр.

У поля адреса отправителя *SA* формат такой же, как у поля получателя.

В области данных могут быть записаны пользовательские данные, предназначенные для обработки (или сформированные) протоколом среднего уровня, например, *IPX*, *TCP/IP* или *NetBIOS*. Но в поле данных может содержаться и один из рассмотренных *MAC-кадров*. Длина поля данных формально не ограничена, но его максимальный размер зависит от того, насколько долго одна рабочая станция может безраздельно распоряжаться сетью.

В поле контрольной кадровой последовательности *FCS* записывается остаток контрольной суммы, вычисленной по методу *CRC-32*. Поле используется для проверки правильности передачи информации.

Поле конечного ограничителя ED. О назначении этого поля в эстафете уже говорилось. В кадре у него появляется дополнительный смысл. Помимо уникальных недвоичных сигналов, в него включены два однобитных подполя. Бит промежуточного кадра (*Intermediate Frame*) равен 1 в том случае, если данный кадр является частью многокадровой передачи, и равен 0, если кадр является последним или единственным в цепочке. Бит обнаружения ошибки (*Error Detected*) при первой передаче кадра равен 0. Сетевой адаптер каждой рабочей станции при ретрансляции кадра контролирует возможные ошибки передачи (например, проверяя поле *CRC*). Если какой-либо из адаптеров обнаруживает ошибку, биту обнаружения ошибки присваивается значение 1. Последующие сетевые адаптеры, приняв кадр с установленным в 1 битом обнаружения ошибки, просто передают его дальше. Адаптер-отправитель, получив обратно свой кадр в таком виде, пытается передать его заново.

Поле состояния кадра FS. В этом поле содержится четыре зарезервированных бита *R* и два подполя: *бит распознавания адреса A* и *бит копирования кадра C*. Биты расположены так: *A C RR A C RR*. Поскольку при вычислении *CRC* поле состояния кадра *FS* не учитывается, то все подполя его для страховки дублируются. Передающая станция при формировании кадра устанавливает бит распознавания адреса (*A*) в нуль.

Принимающая рабочая станция устанавливает этот бит в единичное состояние, сообщая тем самым, что она восприняла свой адрес получателя. Бит

копирования (*C*) также сначала равен 0, но в тот момент, когда содержимое кадра копируется в память принимающей станции (когда станция принимает сообщение), он устанавливается в 1 только в том случае, если кадр принят без ошибки. Если кадр возвращается к станции-отправителю с единичными битами *A* и *C*, то она считает, что переданная ею информация принята успешно.

Если при возвращении кадра к станции-отправителю окажется, что бит распознавания адреса остался нулевым, это значит, что станции-получателя в сети нет – она либо выключена, либо вышла из строя.

Возможна еще одна ситуация: адрес получателя распознан, но бит копирования кадра не установлен в 1. Это значит, что при передаче кадра произошла ошибка и адресат не принял информацию.

Если бит распознавания адреса и бит копирования установлены в 1, но бит обнаружения ошибки тоже 1, станция-отправитель делает вывод, что ошибка в передаче возникла уже после того, как кадр с информацией был успешно принят.

Код прерывания. Это последний формат пакета сообщения. Он может быть послан в любой момент в любом месте потока информации и используется для прерывания или отмены текущей передачи. Код прерывания состоит только из начального *SD* и конечного *ED* ограничителей, рассмотренных раньше в первых двух форматах пакетов.

Волоконно-оптический интерфейс. Волоконно-оптический интерфейс (*FDDI - Fiber Distributed Data Interface*) был создан в институте *ANSI* значительно позже, чем были разработаны протоколы *Ethernet* и *Token Ring*. В качестве передающей среды используется волоконно-оптический кабель с пропускной способностью около 100 Мбит/с. Разработчики *FDDI* старались сделать его максимально соответствующим стандарту *Token Ring*. *FDDI* отличается более высокой скоростью передачи информации на значительно большие расстояния.

Между способами управления эстафетой в сетях *FDDI* и *Token Ring* существуют два основных отличия. В *Token Ring* новая эстафета «запускается» только тогда, когда переданный кадр возвращается обратно к станции-отправителю. В *FDDI* новая эстафета формируется передающей рабочей станцией сразу же после окончания передачи кадра информации. В стандарте *FDDI* не используются подполя приоритета и резервирования, которые предусмотрены в *Token Ring* и нужны для распределения системных ресурсов. Вместо этого все подключенные рабочие станции подразделяются на две группы: *асинхронные*, т. е. такие, для которых моменты доступа к сети не имеют значения, и *синхронные*, для которых требования к интервалам времени между передачами очень жесткие.

Формат кадра *FDDI* очень похож на кадры *Token Ring*. В кадре сети *FDDI*, точно так же, как и в сети *Token Ring*, могут быть переданы управляющие (подуровня *MAC*) или пользовательские (подуровня *LLC*) данные.

Синхронизация передач осуществляется полем *преамбулы*. Поле *управления кадром* определяет, как будет обрабатываться кадр (синхронно или асинхронно), является кадр *MAC*-кадром или *LLC*-кадром (тогда определяется его

тип), определяется длина адреса кадра: 16 или 48 бит. Остальные поля практически совпадают с полями кадра *Token Ring*.

Интерфейс АТМ. Это совершенно новый способ передачи информации в компьютерных сетях. Протокол обмена информацией в режиме *АТМ* на физическом уровне абсолютно не похож на прежние. Основная идея заключается в том, что данные упаковываются в ячейки стандартного размера по 53 байта, снабженные идентификаторами типа данных (текст, видеоизображения, звук, графика и др.). Наличие идентификатора позволяет «интеллектуальным» распределительным устройствам сортировать данные по их типам и следить за тем, чтобы последовательные ячейки передавались адресату в нужном порядке. Скорость передачи информации может достигать 660 Мбит/с.

Локальные беспроводные сети *Wi-Fi*. На современном этапе развития сетевых технологий технология беспроводных сетей *Wi-Fi* (*wireless fidelity* – беспроводная связь) является наиболее удобной в условиях, требующих мобильности, простоты установки и использования. Поэтому во всем мире стремительно растет потребность в беспроводных сетях, особенно в сфере бизнеса. Мобильный Интернет и мобильные локальные сети открывают корпоративным и домашним пользователям новые сферы применения карманных ПК, ноутбуков, мобильных телефонов и т. д. Локальные беспроводные сети (*WLAN* – *wireless LAN*) могут использоваться в офисах для подключения мобильных сотрудников с ноутбуками и носимыми терминалами, в местах скопления пользователей – в аэропортах, бизнес-центрах, на железнодорожных станциях и т. д. Они также подходят для людей, которым по работе необходимо перемещаться по помещению, например в супермаркете или на складе. Так, для учета отпускаемых и получаемых товаров носимые с собой терминалы, постоянно соединенные радиоканалом с корпоративной сетью, позволяют сразу отражать все изменения в центральной базе данных склада.

Wi-Fi технологии становятся все более совершенными, и качество их соединения и безопасность стремительно приближаются к возможностям обычного, широко используемого проводного соединения. Так, например, корпорация *Intel* 12 марта 2003 г. представила технологию *Intel Centrino* для мобильных ПК – компьютеров нового поколения со встроенными функциями беспроводной связи. Технология, которую представляет торговая марка *Intel Centrino* для мобильных ПК, включает в себя процессор *Intel Pentium M*, семейство наборов микросхем *Intel 855* и сетевой интерфейс *Intel Pro/Wireless 2100*. Все компоненты технологии оптимизированы, проверены и протестированы для совместной работы в мобильных системах.

Стандарт *RadioEthernet IEEE 802.11* определяет порядок организации беспроводных сетей на двух нижних уровнях модели *ISO/OSI*: на физическом уровне и уровне управления доступом к среде (MAC-уровне). Другими словами, как и в *Ethernet*, протокол *TCP/IP* накладывается поверх протокола, описывающего передачу информации по каналу связи. Подобно проводному *Ethernet*, *IEEE 802.11* определяет протокол использования единой среды передачи. В стандарте предусмотрено обеспечение безопасности данных, которое включает

аутентификацию для проверки того, что узел, входящий в сеть, авторизован в ней, а также шифрование для защиты от подслушивания. На физическом уровне стандарт предусматривает два типа радиоканалов и один инфракрасного диапазона.

В локальной беспроводной сети есть два типа оборудования: клиент (обычно это компьютер, укомплектованный беспроводной сетевой картой, но может быть и иное устройство) и точка доступа, которая выполняет роль моста между беспроводной и проводной сетями. Точка доступа содержит приемопередатчик, интерфейс проводной сети, встроенный микрокомпьютер и программное обеспечение для обработки данных.

Сети IntraNet. В заключение следует отметить, что в последнее время очень многие корпоративные сети стали использовать технологии *WWW*. Такие сети называют *IntraNet*. Программы для просмотра *Web*-страниц сети Интернет – *Web*-браузеры – обеспечивают доступ к самой разной информации: базам данных, трехмерной графике, текстовым документам и т. д. Это снижает затраты на обучение сотрудников, уменьшает эксплуатационные расходы и позволяет легко подсоединять свои сети к сети Интернет. Кроме того, иногда Интернет используют как транспорт между отдельными частями распределенной большой корпоративной сети.

Для обеспечения безопасности используются специальные программы или программно-аппаратные комплексы – *брандмауэры*. Через них проходит весь поток данных между внутрикорпоративной сетью или ее частью и Интернет. При этом брандмауэр проверяет допустимость пакетов данных, наличие полномочий на передачу тех или иных данных через Интернет и т. д.

В то же время пользователь Интернет может получить доступ к ресурсам других сетей благодаря наличию *межсетевых шлюзов*. Шлюз – это средство межсетевого взаимодействия, размещенное между ЛВС и более крупной системой, например, мейнфреймом или большой сетью с коммутацией пакетов, использующей другие коммуникационные протоколы. Шлюз представляет собой комбинацию программных и аппаратных (включая собственные процессоры и память) средств, используемых для преобразования протоколов. Как правило, шлюз работает медленнее, чем мост или маршрутизатор.

Обеспечение безопасности информации в ЛВС

Как уже указывалось в предыдущей лекции в настоящее время практически все локальные сети подсоединены к Интернету, который стал универсальным средством сотрудничества и общения людей всего мира, средством общемирового вещания и распространения больших массивов информации. Сегодня Интернет становится мощным инструментом электронной коммерции и ведения бизнеса, лишенного каких-либо временных или пространственных ограничений. Вместе с тем Интернет является очень трудно контролируемым каналом распространения информации, и это приводит к тому, что средства *WWW* часто используются для получения несанкционированного доступа к конфиденциальной и закрытой коммерческой информации со стороны злоумышленников и различного рода мошенников. Специфика бизнеса, человеческий фактор, несовершенство законодательства и технические недостатки современных информационных систем обуславливают повышенный риск корпоративных информационных сетей. Об этом достаточно подробно говорилось в лекции 13. В данной лекции мы еще раз с разных позиций рассмотрим вопросы обеспечения безопасности информации в локальных вычислительных сетях.

В локальных вычислительных сетях сосредоточивается информация, исключительное право на пользование которой принадлежит определенным лицам или группам лиц, действующим в соответствии с должностными обязанностями. Такая информация должна быть защищена от всех видов постороннего вмешательства: чтения лицами, не имеющими права доступа к информации, и преднамеренного изменения информации. К тому же в вычислительных сетях должны приниматься меры по защите вычислительных ресурсов и средств связи от их несанкционированного использования, т. е. должен быть исключен доступ к сети лиц, не имеющих на это права. Физическая защита системы и данных может осуществляться только в отношении рабочих компьютеров и узлов связи и оказывается невозможной для средств передачи, имеющих большую протяженность. По этой причине должны использоваться средства, исключающие несанкционированный доступ к данным и обеспечивающие их секретность.

Исследования практики функционирования систем обработки данных и вычислительных сетей показали, что существует много возможных направлений утечки информации и путей несанкционированного доступа в системах и сетях.

В их числе можно отметить следующие:

- чтение остаточной информации в памяти системы после выполнения санкционированных запросов;
- копирование носителей информации и файлов информации с преодолением мер защиты;
- маскировка под зарегистрированного пользователя;

- маскировка под запрос системы;
- использование программных ловушек;
- использование недостатков операционной системы;
- незаконное подключение к аппаратуре и линиям связи;
- злоумышленный вывод из строя механизмов защиты;
- внедрение и использование компьютерных вирусов и др.

Обеспечение безопасности достигается комплексом организационных, организационно-технических и программных мер.

Организационные меры защиты:

- ограничение доступа в помещения, в которых происходит подготовка и обработка информации;
- допуск к обработке и передаче конфиденциальной информации только проверенных должностных лиц;
- хранение магнитных носителей и регистрационных журналов в закрытых для доступа посторонних лиц сейфах;
- исключение просмотра посторонними лицами содержания обрабатываемых материалов на дисплее, принтере, в распечатках и т. д.;
- использование криптографических кодов при передаче по каналам связи ценной информации;
- уничтожение красящих лент от принтеров, бумаги и иных материалов, содержащих фрагменты ценной информации.

Организационно-технические меры защиты:

- осуществление питания оборудования, обрабатывающего ценную информацию, от независимого источника питания или через специальные сетевые фильтры;
- установка на дверях помещений кодовых замков;
- использование для отображения информации при вводе/выводе жидкокристаллических или плазменных дисплеев, поскольку дисплей с ЭЛТ дает такое высокочастотное излучение, что его изображение с экрана можно принимать на расстоянии нескольких сотен метров;
- уничтожение информации, хранящейся в ПЗУ и на магнитных дисках, при списании или отправке их в ремонт;
- установка клавиатуры и принтеров на мягкие прокладки с целью снижения возможности снятия информации акустическим способом;
- охрана территорий и помещений с помощью экранирования машинных залов металлическими листами, установки систем наблюдения и организации контрольно-пропускных систем.

Технические средства защиты

Защита информации в сетях и вычислительных системах с помощью технических средств реализуется на основе организации доступа к памяти с помощью:

- контроля доступа к различным уровням памяти компьютеров;
- блокировки данных и ввода ключей;
- ввода и контроля биометрических данных;
- выделения контрольных битов для записей с целью идентификации.

Архитектура *программных средств защиты* информации включает:

- контроль безопасности, в том числе контроль регистрации вхождения в систему, фиксацию в системном журнале, контроль действий пользователя;
- реакцию (в том числе звуковую) на нарушение системы защиты контроля доступа к ресурсам сети;
- контроль мандатов доступа;
- формальный контроль защищенности операционных систем (базовой общесистемной и сетевой);
- контроль алгоритмов защиты;
- проверку и подтверждение правильности функционирования технического и программного обеспечения.

Для надежной защиты информации и выявления случаев неправомерных действий проводится регистрация работы системы: создаются специальные дневники и протоколы, в которых фиксируются все действия, имеющие отношение к защите информации в системе. Фиксируются время поступления заявки, ее тип, имя пользователя и терминала, с которого инициализируется заявка. При отборе событий, подлежащих регистрации, необходимо иметь в виду, что с ростом количества регистрируемых событий затрудняется просмотр дневника и обнаружение попыток преодоления защиты. В этом случае можно применить программный анализ и фиксировать сомнительные события.

Используются также специальные программы для тестирования системы защиты. Периодически или в случайно выбранное время они проверяют работоспособность аппаратных и программных средств защиты. К отдельной группе мер по обеспечению сохранности информации и выявлению несанкционированных запросов относятся программы обнаружения нарушений в режиме реального времени. Программы данной группы формируют специальный сигнал при регистрации событий, которые могут привести к неправомерным действиям по отношению к защищаемой информации. Сигнал может содержать данные о характере нарушения, месте его возникновения и другие характеристики. Кроме того, программы могут запретить доступ к защищаемой информации или симулировать такой режим работы (например, моментальная загрузка устройств ввода-вывода), который позволит выявить нарушителя и задержать его соответствующей службой.

Один из распространенных способов защиты – явное указание секретности выводимой информации. В системах, поддерживающих несколько уровней секретности, вывод на экран терминала или печатающего устройства любой единицы информации (таблицы, записи или файла) сопровождается специальным грифом с указанием уровня секретности. Это требование реализуется с помощью соответствующих программных средств.

В отдельную группу выделены средства защиты от несанкционированного использования программного обеспечения. Они приобретают особое значение вследствие широкого распространения персональных компьютеров. Исследования, проведенные зарубежными экспертами, свидетельствуют о том, что на одну проданную копию оригинальной программы приходится минимум

одна нелегальная копия. А для наиболее популярных программ это соотношение достигает 1:7, особенно в России.

Большое внимание уделяется законодательным средствам, регулирующим использование программных продуктов. В соответствии с Законом РФ об информации, информатизации и защите информации от 25 января 1995 г. предусматриваются санкции к физическим и юридическим лицам за нелегальное приобретение и использование программных средств. Кроме того, Уголовный кодекс РФ предусматривает уголовную ответственность за преступления в сфере информационной безопасности.

Большую опасность представляют компьютерные вирусы, заражающие другие программы (см. лекцию 13). Если не принимать мер профилактики и борьбы с вирусами, то последствия могут оказаться плачевными.

Комплексное решение вопросов безопасности вычислительной системы или сети принято называть *архитектурой безопасности*, где выделяются угрозы безопасности, службы безопасности и механизмы обеспечения безопасности. Под *угрозой безопасности* понимается действие или событие, которое может привести к разрушению, искажению или несанкционированному использованию ресурсов сети, включая хранимую и обрабатываемую информацию, а также программные и аппаратные средства. Угрозы подразделяются на *случайные* (непреднамеренные) и *умышленные*. Источником первых могут быть ошибки в программном обеспечении, ошибочные действия пользователей, выход из строя аппаратных средств, отказ грозозащиты сетей передачи данных и др.

Умышленные угрозы преследуют цель нанесения ущерба абонентам сети или пользователям вычислительных систем и подразделяются на *активные* и *пассивные*. Пассивные угрозы не разрушают информационные ресурсы и не оказывают влияния на функционирование систем и сетей. Их задача – несанкционированно получить информацию. Активные угрозы преследуют цель нарушать нормальный процесс функционирования путем разрушения или радиоэлектронного подавления линий связи, вывода из строя компьютеров или операционных систем, искажения баз данных и т. д.

К основным угрозам безопасности относятся: раскрытие конфиденциальной информации, несанкционированное использование ресурсов вычислительных систем, отказ от информации.

Создаваемая *служба безопасности вычислительной сети* призвана обеспечить:

- подтверждение подлинности того, что объект, который предлагает себя в качестве отправителя информации в сети, действительно им является;
- целостность информации, выявляя искажения, вставки, повторы и уничтожение данных, передаваемых в сетях, а также последующее восстановление данных;
- секретность всех данных, передаваемых по каналам вычислительной системы;
- нейтрализацию попыток несанкционированного использования вычислительных ресурсов. При этом контроль доступа может быть либо избиратель-

ным, т. е. распространяться только на некоторые виды доступа к ресурсам, например на обновление информации в базе данных, либо полным;

- нейтрализацию угрозы отказа от информации со стороны ее отправителя и/или получателя;

- получателя информации доказательствами, которые исключают попытки отправителя отрицать факты передачи указанной информации или ее содержания.

К *механизмам обеспечения безопасности* относятся: идентификация пользователей, шифрование данных, электронная подпись, управление маршрутизацией и др.

Идентификация пользователей. Право доступа к определенным вычислительным и информационным ресурсам, программам и наборам данных, а также вычислительной сети в целом предоставляется ограниченному контингенту лиц. Для защиты вычислительной сети от несанкционированного доступа применяется идентификация пользователей (сообщений), позволяющая устанавливать конкретного пользователя, работающего за терминалом и принимающего либо отправляющего сообщения. Идентификация пользователей чаще всего производится с помощью паролей. *Пароль* — совокупность символов, известных подключенному к сети абоненту, — вводится им в начале сеанса взаимодействия с сетью, а иногда и в конце сеанса (в особо ответственных случаях пароль нормального выхода из сети может отличаться от входного). Наконец, система может предусматривать ввод пароля для подтверждения правомочности пользователя через определенные кванты времени.

Для защиты средств идентификации пользователей от неправомерного использования пароли передаются и сравниваются в зашифрованном виде, что исключает возможность прочтения паролей без знания ключей защиты.

Для идентификации пользователей могут использоваться и физические методы, например карточка с магнитным покрытием, на которой записывается персональный идентификатор пользователя, карточки с встроенным чипом. Для уменьшения риска злоупотреблений карточки, как правило, используются с каким-либо другим способом идентификации пользователя, например с коротким паролем. Наиболее надежным (хотя и наиболее сложным) является способ идентификации пользователя на основе анализа его индивидуальных параметров: отпечатков пальцев, рисунка линий ладони рук, радужной оболочки глаз и т. д.

Шифрование данных. Секретность данных обеспечивается методами криптографии, т. е. методами преобразования данных из общепринятой формы в кодированную (шифрование) и обратного преобразования (дешифрование) на основе правил, известных только взаимодействующим абонентам сети. Криптография применяется для защиты передаваемых данных, а также информации, хранимой в базах данных, в пакетах магнитных дисков, на гибких магнитных дисках и лентах и т. д.

К криптографическим средствам предъявляются требования сохранения секретности, даже когда известна сущность алгоритмов шифрования-дешифрования. Секретность обеспечивается введением в алгоритмы специальных

ключей (кодов). Зашифрованный текст превращается в исходный в том случае, когда в процессе и шифрования и дешифрования использовался один и тот же ключ. Область значений ключа выбирается столь большой, что практически исключается возможность его определения путем простого перебора возможных значений.

Лекция 25

Глобальные сети. Сеть Интернет

Глобальные компьютерные сети подобны телефонным сетям. Миллионы телефонных аппаратов в мире подключены каждый к своей телефонной станции, а станции соединены между собой каналами связи через коммутирующие узлы. Каждый телефонный аппарат имеет свой номер, каждая телефонная станция также имеет свой номер. Для соединения с абонентом мы набираем номер его аппарата, сигнал поступает на коммутатор нашей станции. По набранному номеру коммутатор подсоединяет линию связи с той станцией, которой принадлежит номер вызываемого абонента. Сигнал приходит на коммутатор той станции, по номеру ее коммутатор подсоединяет линию связи с абонентом. В этой цепи могут подключаться промежуточные станции, если нет прямого соединения нашей станции со станцией абонента. Но нас не интересует, по каким каналам, через какие станции и коммутирующие узлы проходит соединение. Главное, что это соединение произойдет – будь оно в одном городе, где живут абоненты, или за пределами страны.

Так же строятся и глобальные компьютерные сети.

Сеть является средством связи множества различных компьютерных систем. Она не создает никаких новых данных, не занимается их обработкой. Она обеспечивает легкий и быстрый доступ к огромному множеству существующих объектов, созданных разными исполнителями с помощью различных средств и программно-технических платформ. Большую роль в глобальных компьютерных сетях играют надежные, высокоскоростные *магистральные каналы связи*, позволяющие передавать огромные потоки разнородной информации (данные, видео, голос).

В качестве примера кратко рассмотрим сеть Интернет. Разработанные в конце 60-х гг. XX в. по заказу Министерства обороны США принципы организации появившейся на свет в 1970 году распределенной сети *ArpaNet* (равноправность узлов сети, алгоритмы маршрутизации, протоколы *TCP/IP*, пакетная коммутация сообщений и др.) оказались настолько хорошими, что многие организации стали создавать собственные сети на этих же принципах. Впоследствии эти сети стали объединяться между собой (как телефонные станции), образуя единую сеть с общим адресным пространством. Общую сеть стали называть Интернет[♦].

[♦] Дальше по тексту Интернет для простоты иногда будем называть просто Сеть.

Это общемировая совокупность сетей, связанных между собой, объединяющая десятки тысяч сетей и миллионы узлов более чем в 130 странах. Число пользователей сети во всем мире приближается к 1 млрд., из них более 35 млн. – в России. Доступ в Интернет предоставляется организациями, называемыми поставщиками услуг Интернета – провайдерами (*Internet Service Provider*). Пользователь может получить его из дома через модем или из офиса через Web-сервер локальной сети организации. Для подключения к провайдеру могут использоваться обычные телефонные линии, кабельные сети телевидения, радиоканалы связи или спутниковая связь. Провайдер обычно имеет одно или несколько подключений к магистральным каналам или крупным сетям, которые образуют главную «кровеносную» систему Интернета. Все компьютеры в сети можно условно разделить на два основных класса: *клиенты* и *серверы*.

Клиенты – это компьютеры, подключенные к сети, которые используют сеть для получения информации и работы с другими сервисами, например, электронной почтой. Программное обеспечение клиентов включает в себя *браузер* (универсальный клиент) – программу, которая обеспечивает загрузку и отображение Web-страниц. Кроме того, современные версии этого программного обеспечения позволяют выполнять небольшие клиентские приложения (скрипты и апплеты), написанные на специальных языках программирования, например *JavaScript* или *VBScript*, и расширяющие возможности обычных Web-страниц, а также осуществлять путешествия в виртуальных мирах. Программное обеспечение клиентов включает в себя также почтовые приложения, работающие с электронной почтой, мультимедийные средства для воспроизведения аудио- и видеороликов, клиентские модули систем групповой работы, которые обеспечивают создание единого информационного поля над распределенными клиентами и информационными системами.

Серверы – это мощные компьютеры, которые имеют канал связи с Интернетом высокой пропускной способности и соответствующее программное обеспечение, поддерживающее работу тех или иных служб, и, как правило, предоставляют информацию клиентам. Например, Web-серверы работают с протоколами высокого уровня HTTP и в ответ на запрос клиента формируют по заранее определенному алгоритму Web-страницу. Причем эта страница может быть либо создана заранее, и тогда ее называют статической, либо генерируется в ответ на запрос – это динамически созданная страница. Именно программное обеспечение определяет, является ли данный компьютер сервером или клиентом. Серверы обеспечивают работу всех служб и сервисов Интернета.

В настоящее время ПК и Интернет составляют не менее важную инфраструктуру, чем пути сообщения или энергосистема страны. Интеграция всех современных видов телекоммуникаций в единую глобальную информационную гипермедийную систему, которая одновременно является средой для сотрудничества и общения людей, средством общемирового вещания и распространения информации, а также мощным инструментом электронной коммерции и ведения бизнеса, лишенного каких-либо временных или пространственных ограничений, сделали роль Интернета незаменимой. Если изначально Интернет представляла собой средство распространения информации, то сегодня информаци-

онная концепция Сети отходит на второй план. На первое место выходит использование Интернета в коммерческой деятельности предприятий. Сеть становится неотъемлемой частью ведения бизнеса. Интернет играет существенную роль и в так называемой «конкурентной» разведке – сборе сведений о действиях и планах конкурентов с целью дальнейшего ослабления их рыночных позиций.

Протоколы Интернета. *Фундаментальным принципом Интернета является равнозначность всех объединенных в ней физических сетей.* В пределах каждой физической компьютерной сети подсоединенные к ней компьютеры (в разных сетях они могут быть разнотипные) используют тот или иной протокол передачи данных: *Ethernet, Token Ring, FDDI, ISDN, ATM*, беспроводные (радио и спутниковые) технологии, различного формата пакеты передачи данных и др. Технология Интернета не навязывает какой-то определенной топологии межсетевых соединений.

Основой построения Интернета является семейство протоколов TCP/IP. Полную совокупность таких протоколов, необходимых для успешного взаимодействия разных элементов в рамках сети данного типа, принято называть *семейством* или *стеком*. Для конечного пользователя представление о протоколе является представлением о некотором своде четко определенных правил, которые одинаково реализованы в разных системах (программах, шлюзах, пакетах данных, передаваемых по сети и т. д.). *Между механизмами коммуникаций, зависящими от данных физических сетей, и прикладными системами встраивается новое программное обеспечение, которое обеспечивает соединение различных физических сетей друг с другом.* При этом детали такого соединения «скрыты» от пользователей и им предоставляется возможность работать как бы в одной большой физической сети. Для соединения двух или более сетей используются *маршрутизаторы (routers)* – компьютеры, которые физически соединяют сети друг с другом и с помощью программного обеспечения передают пакеты информации из одной сети в другую.

Добавление новой сети к Интернету не влечет за собой ее подсоединения к некоторой центральной точке коммутации или к установке непосредственных физических соединений со всеми уже имеющимися в Интернете сетями. Маршрутизатор «знает» топологию Интернета за пределами тех физических сетей, которые он соединяет, и, основываясь на адресе сети назначения, передает пакет информации по тому или иному маршруту. В Интернете также реализован принцип независимости пользовательского интерфейса от физической сети, т. е. существует множество способов установления соединений и передачи данных, одинаковых для всех физических сетевых технологий.

Стек протоколов *TCP/IP: TCP (Transmission Control Protocol – протокол управления передачей), IP (InterNet Protocol – межсетевой протокол)* является многоуровневой структурой, приведенной на рис. 35.

Нижний (*физический*) уровень соответствует уровню доступа к сети. В протоколах сети он не регламентируется, но поддерживает все популярные стандарты протоколов физического и канального уровней других сетей. Протоколы этого уровня обеспечивают передачу пакетов данных в сети на уровне драйвера устройства в операционной системе и соответствующей сетевой платы ком-

пьютера. Используются они для преобразования адресов сетевого уровня в адреса физической сети и обратно.

Следующий уровень – это уровень *межсетевого взаимодействия*, который обеспечивает передачу пакетов информации из одной подсети в другую. Сетевой уровень – основа *TCP/IP*. Именно на этом уровне реализуется принцип межсетевого соединения, в частности маршрутизация пакетов через Интернет. На сетевом уровне протокол реализует ненадежную службу доставки пакетов по сети без установления соединения. Это означает, что будет выполнено все необходимое для доставки пакетов, однако эта доставка не гарантируется. Пакеты могут быть потеряны, переданы не в том порядке, продублированы и т. д. Служба, работающая без установления соединения, обрабатывает пакеты независимо друг от друга. Но главное – именно на этом уровне принимается решение о маршрутизации пакета по межсетевым соединениям. Именно *IP* определяет базовую единицу передачи информации в Интернете – *IP*-дейтаграмму, а программное обеспечение *IP* выполняет функции маршрутизации, выбирая путь для данных. *IP* используется обоими протоколами транспортного уровня. Для определения маршрута поддерживаются специальные таблицы, по которым осуществляется выбор на основе адреса сети, к которой подключен компьютер-адресат.

Хотя стек протоколов и называется *TCP/IP*, а сами протоколы являются важнейшими, тем не менее, они не единственные. Кроме *IP* на сетевом уровне используются также протоколы *ICMP* (Internet Control Message Protocol) и *IGMP* (Internet Group Management Protocol). *ICMP* отвечает за обмен сообщениями об ошибках и другой важной информацией с сетевым уровнем на другом хосте♦ или маршрутизаторе, а *IGMP* используется для отправки *IP*-дейтаграмм множеству хостов в сети.

Следующий уровень называется *транспортным*. На этом уровне функционирует протокол управления передачей *TCP*, который является также базовым протоколом Интернета, как и *IP*. Он обеспечивает надежную передачу сообщений между удаленными друг от друга различными компьютерами. *TCP* позволяет клиенту и серверу приложения устанавливать между собой логическое соединение и затем использовать его для передачи больших массивов информации, как если бы между ними существовало прямое физическое соединение. Протокол позволяет осуществлять дробление потока данных на пакеты, подтверждать получение пакетов, задавать тайм-ауты, организовывать повторную передачу в случае потери данных и др. Поскольку этот протокол реализует

Прикладной уровень	Протоколы: HTTP, SNMP, FTP, E-MAIL, TELNET, SMTP, TFTP, IRC, GOPHER и другие
Транспортный уровень	TCP, UDP
Сетевой уровень	IP, ICMP, IGMP
Канальный (физический) уровень или сетевой интерфейс	Не регламентируется. Сетевой адаптер и драйвер устройства: Ethernet, Token Ring, FDDI, X 25, SLIP, PPP и другие

Рис. 35. Стек протоколов TCP/IP

♦ Хост – любой компьютер, подключенный к Интернету.

гарантированную доставку информации, использующие его приложения получают возможность игнорировать все детали такой передачи. Второй транспортный протокол *UDP* (User Datagram Protocol) реализует гораздо более простой алгоритм передачи: он посылает пакеты данных с одной машины на другую, но не предоставляет никаких гарантий их доставки.

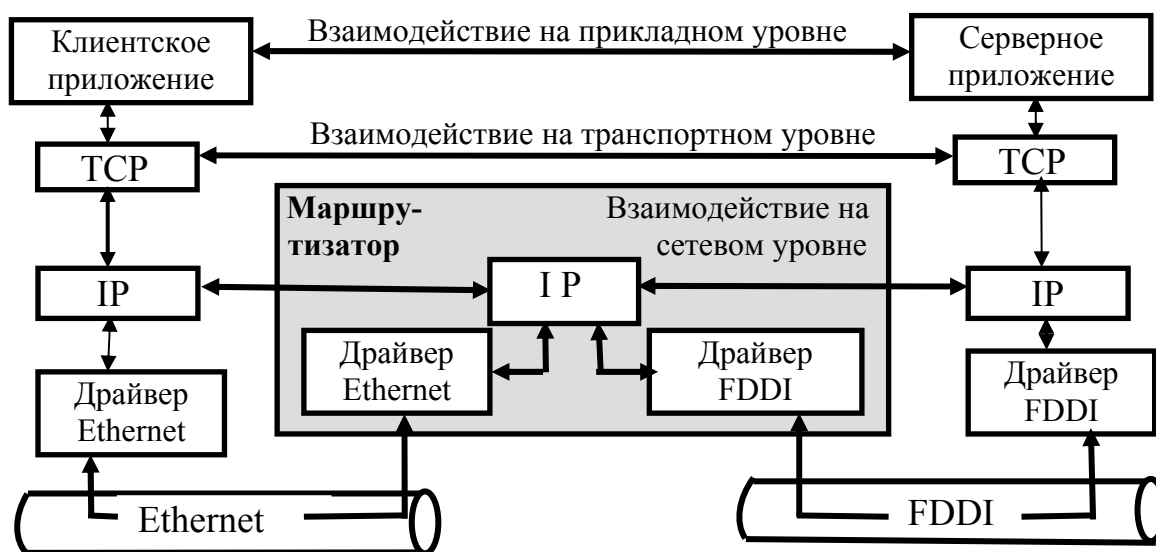


Рис. 36. Взаимодействие двух компьютеров в Интернете

Для конечного пользователя наиболее необходима компетентность на самом верхнем уровне, который называется *прикладным*. Прикладной уровень – это приложения типа клиент–сервер, базирующиеся на протоколах нижних уровней. В отличие от протоколов нижних уровней, протоколы прикладного уровня занимаются деталями конкретного приложения, и для них неважны способы передачи данных по сети. Между клиентом и сервером может быть несколько десятков маршрутизаторов и множество промежуточных физических сетей различных типов. Пользователь (приложение) в любом случае будет воспринимать все это как единую физическую сеть.

На рис. 36 показано взаимодействие двух компьютеров в Интернете из разных сетей с использованием стека протоколов *TCP/IP*. За долгие годы использования в сетях различных стран и организаций стек протоколов *TCP/IP* накопил большое количество протоколов и сервисов прикладного уровня. Мы очень кратко рассмотрим только некоторые из них.

Адресация компьютеров в Интернете. Каждый компьютер, подключенный к *Интернету*, называется *сайтом* (*Site-место*). Любой компьютер в сети имеет свой уникальный адрес, даже при периодическом временном подключении. В этом случае он каждый раз при подключении к сети может быть разным. Адрес однозначно определяет местонахождение компьютера в сети. *Адреса – важнейшая часть Интернета*. Каждому сетевому интерфейсу присваивается уникальный 32-битный адрес (*IP-адрес*), имеющий определенную структуру. Она задает идентификатор физической сети, к которой подсоединен компьютер, и уникальный идентификатор самого компьютера. На рис. 37 показаны различные классы *IP-адресов*.

Для 32-битных адресов принята десятичная нотация, в которой каждый байт адреса содержит десятичное число от 0 до 255. Адреса класса С, например,

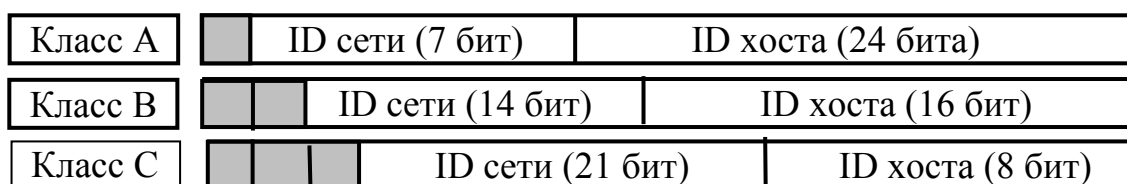


Рис. 37. Классы IP-адресов

охватывают диапазон от 192.0.0.0. до 223.255.255.255. Структура адресов различных классов делает очевидным их применение. Адреса класса С, в которых 21 бит отводится для идентификации сети и только 8 бит – для идентификации компьютера, присваиваются компьютерам локальных сетей небольших организаций, которые могут состоять не более чем из 255 машин. Самих локальных сетей может быть до 16 миллионов. Более крупные организации могут получить адреса класса В, которые способны обслужить до 64 тысяч локальных корпоративных или региональных сетей, в состав каждой из которых может входить до 64 тысяч рабочих станций. Адреса класса А присваиваются компьютерам, подключенным к ограниченному числу глобальных сетей весьма большого масштаба.

Компьютеры, подключенные к нескольким физическим сетям, имеют несколько IP-адресов – по одному для каждого сетевого интерфейса. Соответственно эти IP-адреса различаются своими сетевыми идентификаторами. Таким образом, адрес характеризует не отдельную машину, а ее сетевое соединение. Помимо адресов, предназначенных для одного хоста, существуют также *широковещательные* и *групповые* адреса.

Широковещательные IP-адреса позволяют обращаться ко всем компьютерам сети. В них поле идентификатора хоста состоит только из единиц. Групповые адреса используются для отправки сообщений определенному множеству адресатов. Такая возможность необходима для многих приложений, например, для отправки почты или новостей группе получателей, для проведения интерактивных конференций и др. Для поддержки групповой передачи хосты и маршрутизаторы используют протокол *IGMP*, который предоставляет всем системам в физической сети информацию о том, какие хосты к какой группе принадлежат в настоящее время.

Уникальный IP-адрес назначается каждому сетевому интерфейсу регистрационной службой специальной организации – *Internet Network Information Center (InterNIC)*, которая отвечает за выделение адресов сетям, объединенным в Интернет. Назначение идентификаторов хостам осуществляет системный администратор.

В связи с бурным ростом Интернета 32-битовая схема адресации версии IPv4 в конце XX в. уже не удовлетворяла потребности мировой сети, поэтому в 1991 г. был обнародован проект новой версии IPv6 со 128-битным форматом IP-адреса. Новая версия поддерживает автоматическое назначение адресов.

Помимо четырехкратного увеличения размера адреса, что обеспечивает адресное пространство объемом около 4 квадриллионов адресов в сравнении с существовавшими 4 миллиардами, новый стандарт позволяет реализовывать встроенные функции защиты от несанкционированного доступа, поддержку передачи данных мультимедиа в реальном времени и возможности автоматического реконфигурирования адресов.

Открытость, масштабируемость, универсальность и простота использования – далеко не полный перечень преимуществ *TCP/IP*, но у этого семейства протоколов есть и очевидные недостатки. Столь привлекательная простота доступа оборачивается для Интернета серьезнейшей проблемой защиты информации, которая приобретает особую остроту сейчас, когда мировая сеть все активнее используется для электронной коммерции, а террористические организации активизировали свою деятельность. Неупорядоченность передачи пакетов и невозможность отследить маршрут их продвижения также представляют собой важные проблемы, поскольку препятствуют реализации таких необходимых в современных коммуникациях возможностей, как передача мультимедийных данных в реальном времени. Наконец, как уже упоминалось, предоставляемый объем адресного пространства, особенно в связи с его неэффективным использованием, уже с большим трудом позволяет удовлетворять потребности гигантской и все разрастающейся сети. Многие проблемы должны быть сняты реализацией упоминавшегося протокола новой версии.

Контролем использования *TCP/IP*, определением основных направлений развития, разработкой и утверждением стандартов сегодня занимаются несколько организаций. Основной из них является *Internet Society (ISOC)* – профессиональное сообщество, которое занимается общими вопросами эволюции и роста Интернета как глобальной инфраструктуры исследовательских коммуникаций. Под управлением *ISOC* действует *Internet Architecture Board (IAB)* – организация, в ведении которой находится технический контроль и координация Интернета. *IAB* координирует направления исследований и новых разработок для *TCP/IP* и является конечной инстанцией при определении новых стандартов для Интернета. В *IAB* входят две основные группы: *Internet Engineering Task Force (IETF)* и *Internet Research Task Force (IRTF)*. *IETF* – инженерная группа, которая занимается решением ближайших технических проблем Интернета. Она делится на девять подгрупп в соответствии с основными областями (приложения, маршрутизация и адресация, защита информации и др.) и определяет спецификации, которые затем становятся стандартами Интернета. В частности, протоколы *IPv6* и *DHCP* являются плодом усилий *IETF*. В свою очередь, *IRTF* координирует долгосрочные исследовательские проекты по протоколам *TCP/IP* и технологии Интернета в целом.

Имена компьютеров в Сети. *TCP/IP* предоставляет пользователям возможность работать не только с адресами компьютеров, что затруднительно, но и на логическом уровне с их именами. Используется иерархическая доменная структура имен, в которой каждый уровень адресации называется доменом. Имя домена должно иметь смысл, легко запоминаться и вводиться с клавиатуры, а

также не использоваться другой организацией в Интернете. Выбранное имя домена регистрируется. Регистрацией доменов первого уровня занимается *InterNIC*, находящаяся по адресу www.internic.net. Обычно для этого из области *Registration Web*-страницы *InterNIC* получают текстовый бланк и заполняют его в любом текстовом редакторе или заполняют форму *WWW*, используя программу-браузер. *InterNIC* сообщаются некоторые данные о себе: кто будет контактировать с ней по административным, техническим и финансовым вопросам, касающимся домена, имена и *IP*-адреса серверов *DNS*, поддерживающих домен. Заполненная форма отсылается электронной почтой в *InterNIC*. Через некоторое время поступают два ответа: первый – подтверждение получения запроса, второй – разрешение на использование имени домена.

В России регистрацией доменов второго уровня в зоне *ru* занимается РосНИИРОС (НИИ общественных сетей – www.ripn.net) и делегирует право на их администрирование на основании заявки. Организации или физическому лицу, желающим зарегистрировать свой домен, следует обращаться к администратору какого-либо уже существующего домена. Процедура получения домена второго уровня в зоне *ru* достаточно проста, но требует соблюдения ряда требований, которые в целом соответствуют общепринятым мировым стандартам. Порядок регистрации и делегирования установлен документом «Правила и рекомендации администрирования домена *ru*».

Использование имен обеспечивается при помощи распределенной базы данных – системы доменных имен (*Domain Name System, DNS*), которая хранит таблицы соответствия имен и числовых *IP*-адресов на серверах *DNS*. Эта база данных является распределенной, поскольку ни один *DNS*-сервер в Интернете не обладает всей информацией об именах компьютеров. Каждый поддерживает свою базу данных и имеет серверную программу, к которой могут обращаться другие системы (клиенты) в сети.

Для подключения к Интернету и получения имени компьютера необходим *ISP (Internet Service Provider – поставщик услуг Интернета)*. *ISP* предоставляет клиентам доступ к Интернету по каналам связи, а также другие услуги, такие, как аренда пространства на сервере и создание *Web-страниц*. При выборе *ISP* основные критерии – местоположение его (независимо от типа используемого соединения цена растет с увеличением расстояния), цена, надежность обслуживания, набор предлагаемых сервисов, скорость связи, наличие у *ISP* лишних каналов, доступность обслуживающего персонала и сервисной службы. Обращаясь к *ISP*, необходимо указать сервисы и потребность в полосе пропускания. После заключения контракта *ISP* сообщит ваш адрес *IP*, маску подсети, имена *DNS*-серверов, проинструктирует о подключении к сети и порекомендует любое необходимое дополнительное оборудование.

В доменной системе имен начало адреса – часть сети, к которой подключен компьютер, а конец – номер самого компьютера. Домен верхнего уровня *ru* указывает на российскую часть Интернета. Домен низшего уровня определяет организацию, которой принадлежит адрес, например *usart.ru* – УрГУПС. Домен еще более низкого уровня определяет структурное подразделение организации,

например *nis.usart.ru* – научно-исследовательский сектор УрГУПС. Доменная система гарантирует, что во всей сети больше не найдется другого компьютера с данным адресом. Для доменов верхнего уровня существуют соглашения на названия имен, например:

Россия – *ru* (иногда *su*-от СССР);

Германия – *de*;

Франция – *fr*;

Англия – *uk*;

Украина – *ua* и т. д.

Домен верхнего уровня не всегда обозначает страну, часто – род деятельности:

сетевые организации – *net*;

военные организации *mil*;

государственные и правительственные учреждения – *gov* и т. д.

В связи с определенной "теснотой" в верхних уровнях сети ведающая мировой Интернет-политикой Международная ассоциация по присвоению доменных имен после долгих споров и согласований в середине ноября 2000 г. разрешила новые домены высшего уровня:

Европейский союз – *eu*;

музеи – *museum*;

организации туристического бизнеса – *aero*;

для личных Web-адресов – *name*;

для кооперативов – *coop*;

для коммерческих организаций – *biz*;

для профессиональных групп (врачей, бухгалтеров, адвокатов) – *pro*;

для «разных целей» – *info*.

Названия низшего уровня – произвольны и задаются по усмотрению создателей Web-страниц.

Самое левое имя адреса часто указывает на тип информации:

www – WEB-страница во Всемирной паутине;

ftp – сервер-хранилище файлов.

Когда вводится символьное имя, компьютер обращается к DNS-серверам (*Domain Name System*), которые хранят таблицы соответствия имен и числовых IP-адресов (*InterNet Protocol*).

При работе в *Интернете* чаще используют не просто доменные адреса, а универсальные указатели ресурсов *URL (Universal Resource Locator)*. В указателе кроме собственного адреса указывается, каким протоколом следует обращаться к данному ресурсу, какую программу следует запустить на сервере и к какому конкретно файлу следует обратиться на сервере. Так, например, в *URL* ресурсе *ftp://www.mycompany.ru/doc/resume.htm* – к файлу *resume.htm* в каталоге *doc* на сервере *www.mycompany.ru* следует обратиться по протоколу передачи файлов *ftp*. После доменного адреса может указываться номер порта, т. е. номер программы, которую надо загрузить на сервере (они почти всегда берутся по умолчанию, и пользователю не требуется его знать.). Здесь не следует путать но-

мер порта с портом в компьютере, являющимся путем для обмена информацией между его устройствами. Типы протоколов также определяются и подставляются в адреса современными программами самостоятельно. Поэтому их можно не указывать.

Часто *URL* называют *адресами ресурсов*, например, адреса электронной почты. Справа от символа @ располагается Интернет-адрес компьютера, на котором расположено почтовое отделение абонента. Этот адрес формируется так же, как и любое доменное имя. Слева от символа @ расположено имя абонента, например, *rajal@online.ural.ru*.

Обеспечивают пользователей всеми возможностями Интернета программы, работающие на компьютерах сети. При этом для обеспечения любой услуги, например *WWW*, *FTP* и других, всегда надо иметь две программы. Одна программа – «сервер» – занимается хранением и передачей информации по запросу других компьютеров, вторая программа – «клиент» – устанавливается на компьютере пользователя и служит для посылки запросов на сервер, получения и отображения полученной информации на экране компьютера пользователя. Чтобы программа «клиент» знала, с какой программой «сервер» она работает, как упоминалось выше, используются так называемые порты. Порт для Интернета – это число, которое связывает программы в сети. При работе в Интернете ваш компьютер посылает на узел сети номер порта, который говорит о том, какую программу–сервер необходимо запустить.

Лекция 26

Службы, или сервисы, Интернет

В Интернете имеется ряд протоколов, построенных на базовых протоколах *TCP/IP* и предлагающих разнообразный сервис. Информационными ресурсами Интернета является вся совокупность информационных технологий и баз данных, доступных при помощи этих технологий, существующих в режиме постоянного обновления. Кратко рассмотрим некоторые из ресурсов Интернета, предоставляемые на основе этих протоколов.

1. *Электронная почта e-mail (и-мэйл)*. Электронная почта – один из старейших и первых сервисов Интернета. Это чрезвычайно важный информационный ресурс Интернета – на протяжении длительного времени она является самым распространенным и дешевым средством обмена информацией. При этом стоимость электронной почты не зависит от расстояния. Электронная почта широко распространена в документообороте. Помимо того, что она представляет собой самое массовое средство электронных коммуникаций, через нее можно принять или послать сообщения еще в два десятка международных компьютерных сетей, часть из которых вовсе не имеет прямого подключения к Интернету.

Принцип работы с электронной почтой очень схож с работой обычной почты. С помощью специальной программы (например, *Интернет Mail* из пакета

Интернет Explorer) или обычного текстового редактора подготавливается письмо и кладется в папку для исходящих посланий на вашем компьютере, в которую можно положить множество различных писем. После этого надо соединиться с почтовым отделением, расположенным на сервере поставщика услуг Интернета, и обменяться почтой: переслать подготовленные к отправке письма и принять пришедшую на ваш адрес корреспонденцию в папку входящих писем, которая расположена на вашем ПК. Пересылка писем из почтового отделения по Интернету происходит без вашего участия. Письмо, в конце концов, попадет в почтовое отделение получателя, а чтобы получатель его получил, ему надо подсоединиться к своему почтовому отделению. Почтовые серверы постоянно подключены к Интернету, тогда как компьютеры участников переписки могут устанавливать соединение с ними по мере необходимости. Таким образом работает пересылка файлов по протоколу *UUCP* (*ЮЮСиПи* – *Unix-to-Unix Copy Program*). Долгое время этот протокол был единственным в нашей стране сервисом Интернета. В настоящее время лучше использовать современную электронную почту, работающую по протоколам *SMTP* (*Simple Mail Transfer Protocol* – *простой протокол пересылки почты*) и *POP3* (*Post Office Protocol* – *протокол почтового офиса*). В отличие от *UUCP*, который не использовал *TCP/IP* и, строго говоря, не является протоколом Интернета, последние два построены на основе базового *TCP/IP* и являются стандартными протоколами Интернета. Кроме того, технические особенности реализации этих протоколов обеспечивают большую скорость пересылки писем по сравнению с *UUCP*. Достоинством *UUCP* является его возможность доставлять почту по плохим телефонным каналам, хотя и медленнее. Для России это является одной из проблем. Объем данных, передаваемых пользователями Рунета друг другу, постоянно растет – это и архивы фотографий из отпуска, и объемные презентации для клиентов, и многое-многое другое. До настоящего времени пересылать такие большие файлы по Интернету было довольно затруднительно – требовалось загрузить их отдельно на специальные сервисы, называемые *файлхостингами*, скопировать ссылку в текст электронного письма и только потом его отправить.

В настоящее время ведущие порталы Рунета предлагают разные варианты решения этой проблемы. Например, если вы пользуетесь почтовой службой *mail.ru*, то документы, превышающие максимальный размер вложения, теперь автоматически загружаются на следующий файлхостинг *Файлы@Mail.Ru* (<http://files.mail.ru>). С каждым таким посланием можно передать сразу 20 файлов общим объемом 20 Гигабайт. Все файлы к письму можно загружать одновременно, а не по очереди, что несомненно очень удобно.

Как уже отмечалось, письмо можно отослать и не пользователю Интернета. **В настоящее время практически все локальные компьютерные сети имеют выход в Интернет.** Они соединяются *шлюзами*, которые подключены одновременно к Интернету и к другой сети. Чтобы переслать электронное письмо пользователю другой сети, достаточно правильно указать его адрес. Письмо будет передано шлюзу, который отошлет его получателю.

Почтовый адрес состоит из двух частей: первая часть – это идентификатор пользователя, записываемый перед знаком «коммерческого эй» (@-

собака) и вторая часть – доменный адрес машины, который записывается после знака @. Например, *yuri@cc.usart.ru*. По почте можно получить доступ ко многим ресурсам Интернета, которые имеют почтовых работников (специальные программы автоматического обслуживания), отвечающих на запросы.

2. Вторая по распространенности служба Интернет называется *группа новостей Usenet*, или *телеконференции*, работающая по протоколу *NNTP–Network News Transfer Protocol*. Система построена по типу электронных досок объявлений или «коллективных» почтовых ящиков, когда любой пользователь может поместить свою информацию в одну из групп новостей *Usenet* и эта информация станет доступной другим пользователям, которые на данную группу новостей подписаны. Эта служба работает примерно так же, как и электронная почта, но получаемые письма доступны для общего обозрения, а не только для конкретного пользователя, как в *e-mail*. Новости разделены по иерархически организованным тематическим группам, и имя каждой группы состоит из имен подуровней. Существуют как глобальные, так и локальные для какой-либо страны сети, или организации иерархии. Набор групп, получаемых локальным сервером телеконференций, определяется администратором этого сервера и наличием таких групп на других серверах, с которыми обменивается новостями данный сервер. Существует более 3500 телеконференций, размещенных на серверах в Интернете. Телеконференции позволяют обмениваться мнениями с единомышленниками, живущими в любой точке земного шара. Эти электронные доски объявлений могут читать любые пользователи по желанию. Пользователь *Usenet* может взаимодействовать только со «своим» сервером и не может обратиться к произвольному серверу сети. Однако доступ он получает ко всей информации, которая присутствует в данной информационной системе, так как серверы обмениваются ею между собой. В соответствии с протоколом *NNTP* информация между серверами автоматически обновляется таким образом, что сообщение, направленное на любой из серверов, в конечном итоге в течение 2–3 часов оказывается на каждом из них. Каждый узел сети, получивший новое сообщение, передает его всем узлам, с которыми он обменивается новостями. Общий список телеконференций публикуется в сети. Именно этим способом распространяется большинство сообщений Интернета, например списки наиболее часто задаваемых вопросов (*FAQ*) или реклама программных продуктов. По *Usenet* можно получить и вирус, если заказывать и распаковывать все подряд, что приходит на ваш почтовый адрес. *Usenet* – хорошее место для объявления международных конференций и семинаров.

3. *Интернет* является крупнейшим хранилищем файлов в мире. Служба *Интернет* по протоколу *FTP (File Transfer Protocol – протокол передачи файлов)* позволяет получать и передавать файлы. *FTP* – служба прямого доступа, требующая полноценного подключения к Интернету. Эта служба является одним из основных способов распространения бесплатных программ и различных дополнений и исправлений к коммерческим версиям программ.

Система файловых архивов – это огромное распределенное на разных машинах сети, в том числе и функционирующих на разных платформах, храни-

лище всевозможной информации, накопленной за время существования Интернета. Информация в *FTP*-архивах разделена на три категории:

1. Защищенная информация, режим доступа к которой определяется ее владельцами и разрешается по специальному соглашению с потребителем. Сюда относятся коммерческие архивы, например, коммерческие версии программ, закрытые национальные и международные некоммерческие ресурсы, информация частных благотворительных фондов и т. д.

2. Информационные ресурсы ограниченного использования, к которым относятся, например, программы класса *shareware* (*Trumpet Winsock*, *Netscape Communicator* и др.). В данный класс могут входить ресурсы ограниченного времени использования, или ограниченного времени действия, т. е. пользователь может использовать текущую версию на свой страх и риск, но никто не будет оказывать ему поддержку.

3. Свободно распространяемые информационные ресурсы, или *freeware*, если речь идет о программном обеспечении. Сюда относится все, что можно свободно получить по сети без специальной регистрации. Свободно распространяемое программное обеспечение не имеет сертификата качества, но, как правило, его разработчики открыты для обмена опытом.

Любой пользователь может воспользоваться услугами анонимного доступа к этому хранилищу и скопировать интересующие его материалы. Объем программного обеспечения в архивах *FTP* составляет терабайты информации, и ни один пользователь или администратор сети не может просто физически обозреть эту информацию. Кроме программ в этих архивах можно найти стандарты Интернет – *RFC (Request for Comments)*, пресс-релизы, книги по различным отраслям знаний, главным образом по компьютерной тематике, различные рефераты и многое другое. Практически любой архив строится как иерархия директорий. Многие архивы дублируют информацию из других архивов (так называемые зеркала – *mirrors*). Для того чтобы получить нужную информацию, вовсе не обязательно ждать, когда она будет передана из Австралии или Южной Африки, можно поискать «зеркало» где-нибудь ближе. *Стержень технологии составляет FTP-протокол. Это один из старейших протоколов Интернета, который входит в его стандарты.*

Данная служба позволяет адресовать файловые архивы *FTP* из программ-клиентов *WWW* (см. дальше). При этом программа-клиент браузер (от *browse* – *просматривать*), обозреватель или навигатор должна поддерживать протокол *FTP*. В этом случае в окно, например *Location Netscape Navigator*, для доступа к архиву *ftp.kiae.su* следует ввести *URL*-адрес: *ftp://ftp.kiae.su*. В настоящее время доступ по *FTP*-протоколу осуществляется из множества мультипротокольных интерфейсов (*Интернет Explorer*, *Netscape Navigator*, *Mosaic*) или графических *ftp*-оболочек типа *ftptool*. Все они гораздо удобнее и проще в использовании, но потребляют значительно больше ресурсов. В данной схеме возможно указание не только имени схемы, адреса *FTP*-архива, но и идентификатора пользователя и даже его пароля. Наиболее часто для доступа к публичным архивам *FTP* используется адрес *ftp://polyn.net.kiae.su/pub/index.txt*

В этом случае записана ссылка на архив *polyn.net.kiae.su* с идентификатором «*anonymous*» или *ftp* (анонимный доступ). Если есть необходимость указать идентификатор пользователя и его пароль, то можно это сделать перед адресом машины: *ftp://nobody:password@polyn.net.kiae.su/users/local/pub*.

Здесь параметры отделены от адреса машины символом @, а друг от друга – двоеточием. В некоторых системах можно указать и тип передаваемой информации, но данная возможность не стандартизирована. Следует также учитывать, что употребление идентификатора и пароля пользователя не рекомендуется, так как данные передаются незашифрованными и могут быть перехвачены. Реальная защита в *WWW* осуществляется другими средствами и построена на других принципах.

В корневом каталоге обширно ветвящегося дерева подкаталогов на FTP-сервере, к которому вы получили доступ, обычно можно разыскать текстовый файл с именем *Index* или подобный ему, в котором содержится характеристика ресурсов данного архива. Поиск FTP-сервера в Интернете – задача сложная и трудоемкая. Для ее облегчения существует специальная поисковая программа *Archie*, которая позволяет просканировать FTP-архивы и найти тот, который устраивает пользователя по составу программного обеспечения и коммуникационным условиям. Работать с *Archie* можно по электронной почте. Программа *Archie* использует базу данных, в которую входят не только имена файлов и каталогов, но также и описание пакетов программ, что особенно полезно, когда нужно найти не конкретный программный пакет, а что-либо, имеющее отношение к той или иной области.

4. Достаточно популярным сервисом в *Интернете* является *IRC (Интернет Chat Relay – беседа через Интернет или система разговорных комнат – чатов)*. Это интерактивная служба, способствующая общению людей через Интернет. Система похожа на группы новостей, но обмен сообщениями в ней ведется без задержек. Подключившись к группе обсуждающих ту или иную проблему, вы набираете свое сообщение на клавиатуре, и оно мгновенно становится доступно другим участникам разговора. Аналогично вы видите сообщения своих собеседников.

5. Наиболее популярен сервис *Интернета*, называемый страницы *Всемирной паутины WWW (World Wide Web – мировая виртуальная файловая система)*. Это самая современная и удобная, самая популярная и интересная служба Интернета прямого доступа, требующая полноценного подключения к Интернету. В основу данной системы положено понятие *гипертекста*. Гипертекстовые документы создаются с помощью специального языка *HTML (Hyper Text Markup Language – язык разметки гипертекста)*. Внешне гипертекст отличается от обычного текста тем, что часть слов или целые строки в нем выделены особым шрифтом или цветом и чувствительны к появлению на них указателя манипулятора «мышь». При попадании указателя на них он меняет свой вид (например, стрелочку на ладонь), а щелчок левой клавиши мыши приводит к загрузке в программу просмотра нового документа, привязанного так называемой гипертекстовой ссылкой к выделенному слову или строке. Ссылки в до-

кументах *WWW* организованы таким образом, что каждый информационный ресурс глобальной сети однозначно адресуется и документ может ссылаться как на другие документы на этом же сервере, так и на документы на других серверах Интернета. Причем пользователь не замечает этого и работает со всем информационным пространством Интернета как с единым целым. Взаимодействие компьютеров осуществляется по принципу клиент–сервер, точнее, клиент–серверы: существует множество серверов, которые по запросу клиента возвращают ему гипермедийный документ – документ, состоящий из частей с разнообразным представлением информации (текст, звук, графика, трехмерные объекты и т. д.), в котором каждый элемент может являться ссылкой на другой документ или его часть. Ссылки *WWW* указывают не только на документы, специфичные для самой службы *WWW*, но и на другие службы и информационные ресурсы Интернета. Большинство браузеров не только понимают гиперссылки, но и являются программами-клиентами других служб Интернета, например *FTP*, *Usenet*, электронной почты и т. д. Таким образом, программные средства *WWW* являются универсальными для различных служб Интернета, а сама информационная система играет интегрирующую роль.

Документ во Всемирной паутине, составленный на языке *HTML* и доступный для просмотра специальной программой *WEB*-браузером, называется *WEB-страницей*. В случае использования фреймов каждый фрейм считается отдельной страницей. Совокупность *WEB*-страниц, объединенных по смыслу и физически находящихся на одном сервере, называется *WEB-сайтом*. Слова в одном документе как бы «привязаны» к другим документам. Просматривая одну страницу и делая в ней ссылку на другую, вы на экране тут же увидите эту страницу, хотя она может находиться на компьютере другого типа с другой операционной системой и в другой части земного шара. Щелкая мышью по различным ссылкам, можно за несколько минут проскакать по *WEB*-серверам в десятке разных стран. Так как ссылки могут указывать на любой документ (или *WWW-страницу*), находящийся в *Интернете* в любом месте земного шара, то данная система и получила название Всемирной паутины. Для работы с *WWW* используется специальный протокол *HTTP (Эйч Ту Ту Пу) – Hyper Text Transfer Protocol – протокол передачи гипертекста*. Принцип работы со Всемирной паутиной очень похож на работу с энциклопедией: вы читаете статьи, находите интересующие вас ссылки и по ним читаете другие статьи.

Для доступа к ресурсам *WWW* пользователь задает адрес ресурса на основании схемы адресации, в которой указываются ее идентификатор (тип протокола), адрес машины, в качестве которого допустимо использование *IP*-адреса, *TCR*-порт (идентификатор прикладной программы, с которой устанавливается связь), путь в директории сервера, критерий для поиска и метка. Метка используется для ссылки на точку внутри файла *HTM*. Например, на сервере с простым адресом <http://www.nua.ie/surveys> представлены региональные или отраслевые исследования аудитории Интернета. Отделенная символом # метка *first* в адресе <http://polyn.net/kiae.su/altai/volume4.html#first> показывает, что делается ссылка на точку внутри файла *volume4.html*, а в следующем адресе предпола-

ется, что *html* – документ с возможностью поиска по двум ключевым словам. При этом в зависимости от поисковой машины знак «+» будет интерпретироваться как «AND», либо как «OR»:

http://polyn.net.kiae.su/isindex.html?keyword1+keyword2.

В отдельную группу служб Интернета можно выделить новые службы, не имеющие такого широкого распространения, как вышеприведенные, и не имеющие всеми признанных единых стандартов. В их основе также лежит использование Интернета как среды передачи информации. В частности, к этой группе можно отнести:

- средства передачи голоса по каналам связи Интернета, предоставляющие услуги телефонной и факсимильной связи;
- программные средства для проведения видео- и аудиоконференций;
- системы ширококвещательной передачи мультимедийной информации, например телепередачи.

Отдельную группу представляют также информационно-поисковые системы, рассматриваемые далее.

Имеется также ряд старых служб Интернета, основанных на базовом протоколе *TCP/IP*, которые в настоящее время в связи с появлением службы *WWW* распространены незначительно. К ним относятся:

Gopher – распределенная информационная система, в основу интерфейсов которой положена идея иерархических каталогов. До сих пор основные информационные ресурсы *Gopher* сосредоточены в университетах.

WAIS – распределенная информационно-поисковая система Интернета. В основу системы положен принцип поиска информации с использованием логических запросов, основанных на применении ключевых слов.

WHOIS – служба содержит информацию о пользователях сети, их электронные и обычные адреса, идентификаторы и реальные имена.

TRICKLE – это доступ по почте к архивам FTP, который организован через специальный шлюз. Этот шлюз имеет специальные навигационные средства для поиска нужной информации. Пользователь может вести с ним своеобразный диалог по почте, выбирая нужную информацию путем ввода специальных команд *TRICKLE*.

Наиболее перспективной областью использования Интернета является электронная коммерция, банковские услуги, платежные системы и реклама. В настоящее время многие фирмы, не жалея средств, развивают свои сайты для придания лучшего облика «лицу» фирмы во Всемирной паутине. *Web*-сайты дают уникальную возможность для фирмы ненавязчиво предоставить большой объем информации о себе и производимых продуктах. Многие фирмы при выпуске новых продуктов специально создают сайт с их детальным описанием, а потом через него получают отзывы покупателей и оперативно корректируют свою деятельность в отношении этих изделий. Значительный экономический эффект дает управление оборотными средствами предприятий за счет ускорения их оборачиваемости. Можно осуществлять коммерческие сделки непосредственно с помощью Интернета, управляя средствами на своем счету в банке. Можно связаться с поставщиком товаров, имеющим свой сервер в Ин-

тернете, выбрать товар и сделать заказ на его доставку, а затем произвести его оплату со своего счета в банке. Кроме представления описания товара посредством Интернета можно осуществлять его продажу. Уже многие компании имеют свои *online*-магазины. Более подробно о развитии Интернет-бизнеса см. [9] в списке дополнительной рекомендованной литературы.

Информационные ресурсы и сервис Интернета образуют единое мировое пространство, которое поддерживается правительствами, крупнейшими университетами, корпорациями, научными центрами всего мира, предоставляя пользователям уникальные возможности, которые трудно себе представить. Отсутствующие или непривычные сегодня услуги сети вскоре могут прочно войти в нашу повседневную жизнь.

Лекция 27

Поиск информации в Интернете. Классификация *Web* - сайтов

Стратегия поиска. При поиске информации и использовании ресурсов Интернета встает вопрос: как найти необходимую информацию в условиях, когда ее поток непрерывно лавинообразно и до определенной степени бесконтрольно возрастает. Сегодня в распоряжении пользователей Интернета находятся миллиарды документов, информация в которых подвергается быстрому изменению. Навигационные методы просмотра, обеспечиваемые рассмотренными выше протоколами, становятся малоэффективными. Наиболее успешной попыткой обуздать информационный хаос в Интернете является культивирование поисковых систем самого широкого профиля. В их функции входит автоматическое или полуавтоматическое сканирование (просмотр) узлов (хостов) сети, сопровождающееся индексированием (созданием баз данных) и классификацией их ресурсов (построением каталогов, структурированных по различным критериям) с возможностью последующего обслуживания поисковых запросов клиентов. Следовательно, стратегия поиска обобщенной поисковой системы представляется функциональной цепочкой вида:

Сканирование → Индексирование → Классификация → Обслуживание

В мире Интернета технологии *WWW* произвели революцию, следствием которой стали следующие факторы:

- неуклонное нарастание числа серверов, реализующих *http*-протокол;
- перенесение наиболее востребуемых ресурсов на *Web*-узлы с серверов, поддерживающих другие протоколы доступа;
- разработка системы межпротокольных шлюзов *WWW-Gopher*, *WWW-FTP*, *WWW-Telnet* и др.

Существование шлюзов между протоколами прикладного уровня позволяет, например, поисковой системе *WWW* сканировать ресурсы *FTP* - архивов, тем не менее, инфраструктура межпротокольных шлюзов оказывается недоста-

точно для формирования однородного информационного пространства. В результате для исчерпывающего профессионального поиска информации в сети следует прибегать к специальным поисковым средствам, характерным для среды того или иного протокола, а не ограничиваться наиболее развитыми сегодня средствами поисковых систем WWW, надеясь на полноту охвата остальной части Интернета благодаря шлюзам.

По этой причине поиск информации в Интернет сегодня поднимается на уровень технологии.

Рассмотрим подробнее приведенную выше функциональную цепочку стратегии поиска информации.

Сканирование. В процессе сканирования ресурсов Интернета используются специальные программы – *сканеры* типа агентов, пауков, кроулеров (*Crawler*) и роботов. Работа таких программ обычно происходит в автоматическом режиме и состоит в последовательном обходе узлов сети на основе заданного собственного алгоритма, который может отдавать определенные предпочтения тем или иным узлам как на основе их географической или профильной принадлежности, так и частоты изменения находящихся на них ресурсов. Кроме того, учитываются интересы компаний, стремящихся включить свои серверы в индексную базу данной поисковой системы и проинформировать о них широкий круг пользователей сети. Для ускорения работы *сканеры* размещаются на нескольких машинах, каждая из которых выполняет свое задание. Так, робот на одной машине может «скачивать» новые страницы, которые еще неизвестны поисковой системе, а на другой – страницы, которые были просканированы не менее месяца, но и не более полугода назад. При этом все машины используют одно и то же хранилище. Параллельная работа *сканера* на нескольких машинах позволяет легко справляться с заданием при увеличении количества узлов Сети, которые нужно сканеру обойти – достаточно распределить работу по сканированию на большее число машин.

Общий поиск информации в Сети осуществляют программы-*пауки* (*Spider*). Они сообщают о содержании найденного документа, просматривают заголовки, некоторые ссылки, индексируют его и извлекают итоговую информацию. Проиндексированную информацию они посылают базе данных поисковой системы. *Агенты* – самые «интеллектуальные» из поисковых средств. Они могут делать больше, чем просто искать. Они могут даже выполнять транзакции от вашего имени, искать сайты специфической тематики и возвращать списки сайтов, отсортированных по их посещаемости. *Агенты* могут обрабатывать содержание документов, находить и индексировать различные виды информации. Некоторые, например, индексируют каждое отдельное слово в документе, в то время как другие индексируют только наиболее важных 100 слов в каждом, индексируют размер документа и число слов в нем, заголовки, подзаголовки и т. д. Они могут быть запрограммированы для извлечения информации из уже существующих баз данных поисковых систем. При этом независимо от информации, которую *агенты* индексируют, они передают ее обратно базе данных поисковой системы. *Кроулеры* просматривают заголовки и возвращают только первую

ссылку. *Роботы* могут переходить по различным ссылкам разной глубины вложенности, выполнять индексацию и даже проверять ссылки в документе. Таким образом, важной характеристикой системы является число уже отсканированных узлов и скорость работы сканирующих программ. В процессе сканирования поисковой системе приходится получать доступ к ресурсам сети. Естественно, что такой доступ реализуется в рамках одного из протоколов прикладного уровня. В связи с этим принято различать поисковые системы по области сканирования. Это гипертекстовые базы данных *Web*, *FTP*-архивы, ресурсы всемирного пространства *GopherSpace* и др.

Индексирование выполняет также специальная программа – *индексатор*, которая занимается обработкой ресурсов, доставленных сканерами. Ее задача – формирование базы данных поисковой системы, организованной по определенным принципам. В первую очередь предметом сканирования являются текстовые документы. В результате такой операции для каждого документа формируется набор ключевых слов, по которым затем на стадии обслуживания поискового запроса пользователю выдаются адреса проиндексированных ресурсов. Информационные объекты нетекстового характера (графика, видео, аудио) в общем случае также могут идентифицироваться и быть представлены в соответствующих базах данных – индексах. Информация, находящаяся в хранилище в сжатом виде, делится на несколько частей. Эти части распределяются между машинами, на которых запущена программа-индексатор. После того как вся информация обработана, начинается объединение результатов, а также расстановка меток: откуда взят тот или иной документ. Частичные индексные базы и основная база, к которой обращается поисковая машина, имеют одинаковый формат, поэтому сборка единой базы из частных индексных баз представляет собой простой и быстрый процесс, происходящий со скоростью чтения данных с диска. Специальная программа составляет таблицы перенумерации документов базы и объединяет содержимое всех частей. Среди страниц с одинаковыми адресами выбирается наиболее свежая версия. Если при скачивании веб-адресов (*URL*) последней информацией была ошибка (в Рамблере, например, 404), то это означает, что запрашиваемая страница уже не существует, и она временно удаляется из индексной базы. Параллельно осуществляется объединение в один документ страниц, которые имеют одинаковое содержимое, но разные *URL*.

Классификация ресурсов является дополнительной функцией поисковой системы, которая предполагает, например, присвоение при индексировании пометки о принадлежности данного информационного объекта к определенному типу. Несмотря на стремительное нарастание в Интернете средств мультимедиа, на сегодняшний день основным носителем информации в ней остается текстовый документ. Тем не менее, даже при поиске документа возникает серьезная проблема. Список документов, полученных в результате отклика на пользовательский запрос, сводящийся, например, к указанию единственного ключевого слова для поиска совпадений с ним в пространстве *WWW*-страниц, может содержать в себе тысячи пунктов. Но и в этой ситуации результат может иметь практическое значение, если отображаемый список представляет документы в по-

рядке убывания их предполагаемой значимости для пользователя. Такое упорядочивание списка документов по релевантности реализуется каждой поисковой системой в рамках своего алгоритма. Благодаря тому что для разметки текста в WWW используется язык HTML, обладающий средствами особого выделения, например заголовков разного уровня на странице, оказывается достаточно просто реализовать алгоритм сканирования Web-страницы с учетом разного веса различных текстовых полей. Более того, многие поисковые системы WWW позволяют заранее задавать в поисковом запросе то текстовое поле, в котором должен встретиться данный термин.

Чтобы определить порядок, в котором список документов будет показан по запросу, база данных применяет алгоритм ранжирования. Различные поисковые системы используют различные алгоритмы ранжирования, однако, основные принципы определения релевантности следующие:

- количество слов запроса в текстовом содержимом документа (в *html*-коде);
- тэги, в которых эти слова располагаются;
- местоположение искомых слов в документе;
- удельный вес слов, относительно которых определяется релевантность, в общем количестве слов документа.

Обслуживание пользователей той или иной поисковой системой строится на разработке информационно-поискового языка, естественным образом связанного со структурой базы данных. Типичными являются два основных подхода: пользователю предоставляется возможность вести поиск интересующей его информации либо путем осмысленного на каждом шаге перемещения по дереву иерархического каталога, уже построенного и жестко определенного системой, либо путем реализации собственного поискового запроса в рамках поддерживаемого системой поискового языка. Конечной точкой обоих путей является локализация и извлечение соответствующего информационного объекта.

Поисковые системы WWW. На сегодня в Интернете доступно значительное число поисковых систем. Среди англоязычных это, например, Excite, Alta Vista, Lycos, WebCrawler, и т. д. Среди русскоязычных поисковых систем наиболее распространены Яндекс, Rambler, Google, Апорт и др. Задача поисковых систем состоит в представлении ссылок на сайты Интернета в соответствии с введенным пользователем запросом.

Корпорация *Digital Equipment Corporation*, поддерживающая в Сети систему Alta Vista, для пользователей, говорящих по-русски, реализовала версию программы, которая позволяет искать информацию во всех русскоязычных кодировках. Как и в англоязычной версии, система поддерживает режимы простого и сложного запросов, а вместе с последним и возможность задавать дату, более сложную логику и возможность упорядочивания результатов. Однако эффективность поиска документов, набранных в кириллице по ключевым словам, может быть нулевой. Дело в том, что морфология русского языка (прежде всего многообразии падежных форм) не дает правильного числа совпадений с терминами в документе, если грамматическая форма термина в поисковом запросе и в документе различаются.

В этом смысле представляют интерес поисковые системы, учитывающие морфологию русского языка, такие, как программные продукты на основе ядра поисковой системы Yandex (<http://yandex.ru>) компании *Comp Tec International*, реализованной на базе данных сервера издательского дома «Открытые системы» (<http://win.vww.osp.ru>), причем поисковый язык является достаточно развитым. Yandex включает модули морфологического анализа и синтеза, индексации и поиска, а также набор вспомогательных модулей, таких, как анализатор документов, языки разметки, конверторы форматов и др. Алгоритмы морфологического анализа и синтеза, основанные на базовом словаре, умеют нормализовывать слова, т. е. находить их начальную форму, а также строить гипотезы для слов, не содержащихся в базовом словаре. Система индексирования позволяет создавать компактный индекс и быстро осуществлять поиск с применением ло-

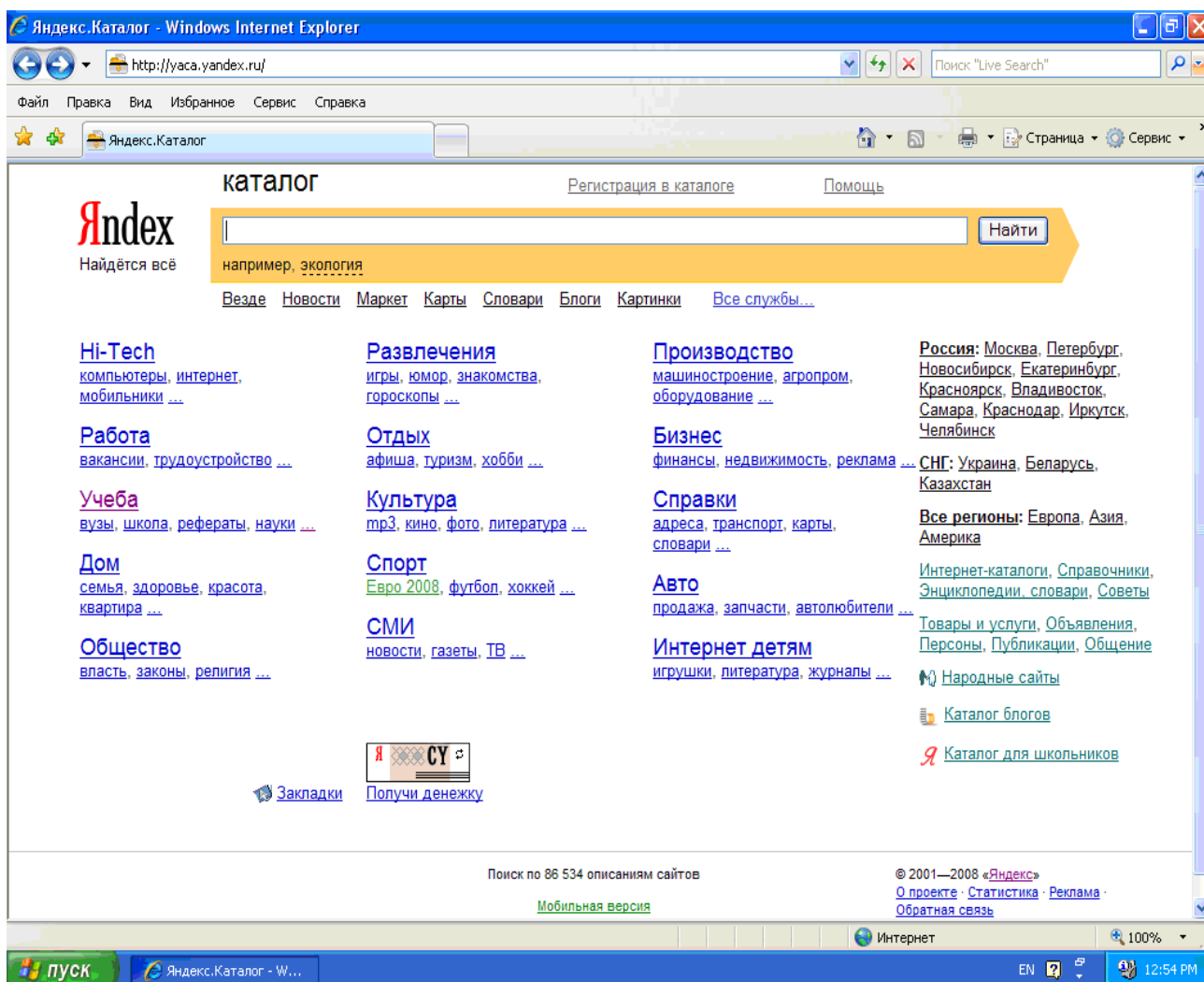


Рис. 38. Начальная страница поисковой системы Yandex

гических операторов. Реализация Yandex для WWW имеет шлюз к поисковой системе Alta Vista и обеспечивает доступ по следующему адресу <http://w.vw.coniptek.ru/alta.henl>. Yandex ежедневно просматривает сотни тысяч Web-страниц в поисках изменений или новых ссылок, коллекция которых по-

стоянно растет. Yandex не требует знания специальных команд для поиска. Достаточно в поле ввода «Найти» набрать вопрос, например «где найти дешевые компьютеры», и вы получите результат – список страниц, где встречаются эти слова. Yandex отыщет по названию *Web*-страницу компании или файл с нужной картинкой.

Начальная страница поисковой системы Yandex приведена на рис. 38.

Большую популярность у пользователей имеет достаточно мощная и быстрая русская поисковая система Rambler (<http://rambler.ru>), поддерживаемая в Сети компанией Stack Ltd. (г. Пущино), которая предоставляет возможность поиска как в *Web*, так и в системе телеконференций при распознавании всех кодировок кириллицы и с применением обычных логических коннекторов. Поисковая система содержит информацию о более чем 12 миллионах документов, расположенных на серверах России и стран СНГ и обрабатывает ежедневно не менее 500 тысяч поисковых запросов, сканируя 48 тысяч *Web*-серверов, используя несколько одновременно работающих программ-роботов. Начальная страница поисковой системы Rambler приведена на рис. 39.



Рис. 39. Начальная страница поисковой системы Rambler

Запрос может состоять из одного или нескольких слов, разделенных пробелами, при этом регистр не имеет значения, потому что и при построении индекса, и при поиске по запросу все заглавные буквы «понижаются». Могут быть использованы как русские, так и английские слова и словосочетания. По умолчанию находятся только те документы, в которых встретились все введенные вами слова. Слова запроса могут быть соединены логическими связками *And* («И») и *Or* («ИЛИ»). Вместо указанных связок (или в сочетании с ними) могут использоваться также символы «&» и «|». Части запроса могут быть сгруппированы с помощью круглых скобок. Возможна многократная вложенность скобок в сочетании с логическими операторами. Rambler умеет искать слова во всех формах (например, узел, узлы, узлом и т. д.). Чтобы слово находилось во всех формах, перед ним надо поставить служебный символ #. В меню детального запроса (справа от основного) такой режим может быть включен для всех слов: «*Расширение запроса: все формы слов*». Служебный символ @ перед словом позволяет находить не только само слово, но и однокоренные слова. В меню детального запроса это будет: «*Расширение запроса: все однокоренные*». Система позволяет использовать и обобщающие знаки: ? и * для обозначения произвольной части слова и произвольного символа. Ограничить поиск частями документов, такими, как название документа, его заголовок, URL и др. можно через меню детального запроса «*Искать в...*». В этом же меню можно выбрать «*Сортировать по...*», чтобы потребовать поместить в начало списка самые свежие (или, наоборот, самые старые) документы. Можно также ограничить поиск документами, созданными в определенный период времени. Для этого необходимо в детальном запросе указать «*От даты... до даты...*». Можно ограничить поиск документа только на одном языке, выбрав режим в меню детального запроса «*Язык документа...*». По умолчанию поиск выполняется по документам на всех языках. По умолчанию же результаты поиска выдаются порциями по 15 документов. На странице детального запроса можно установить «*Выдавать по...*», что увеличит число документов до 30 или 50, а при установке «*Форма вывода...*» можно получить описания документов с увеличенной или уменьшенной подробностью. Все перечисленные правила могут быть использованы совместно друг с другом в необходимой последовательности.

Также очень быстро развивается и становится наиболее популярной русская поисковая система Google (<http://www.google.ru/>), начальная страница которой приведена на рис. 40. Несмотря на впечатляющие рекламные заявления создателей поисковых систем как в отношении числа индексируемых документов, содержащихся в их базах данных, так и в отношении изощренности алгоритмов поиска и ранжирования, только их практическое использование для нужд конкретного пользователя может дать ответ на вопрос, какая из поисковых систем и в какой степени адекватна возлагаемым на нее задачам.

Каталоги. Аналогичную поисковым системам функцию выполняют серверы-каталоги, с той лишь разницей, что поиск осуществляется пользователем по иерархически организованной тематической структуре каталогов. Регистрация ресурсов в каталогах, в отличие от поисковых систем, не является автома-

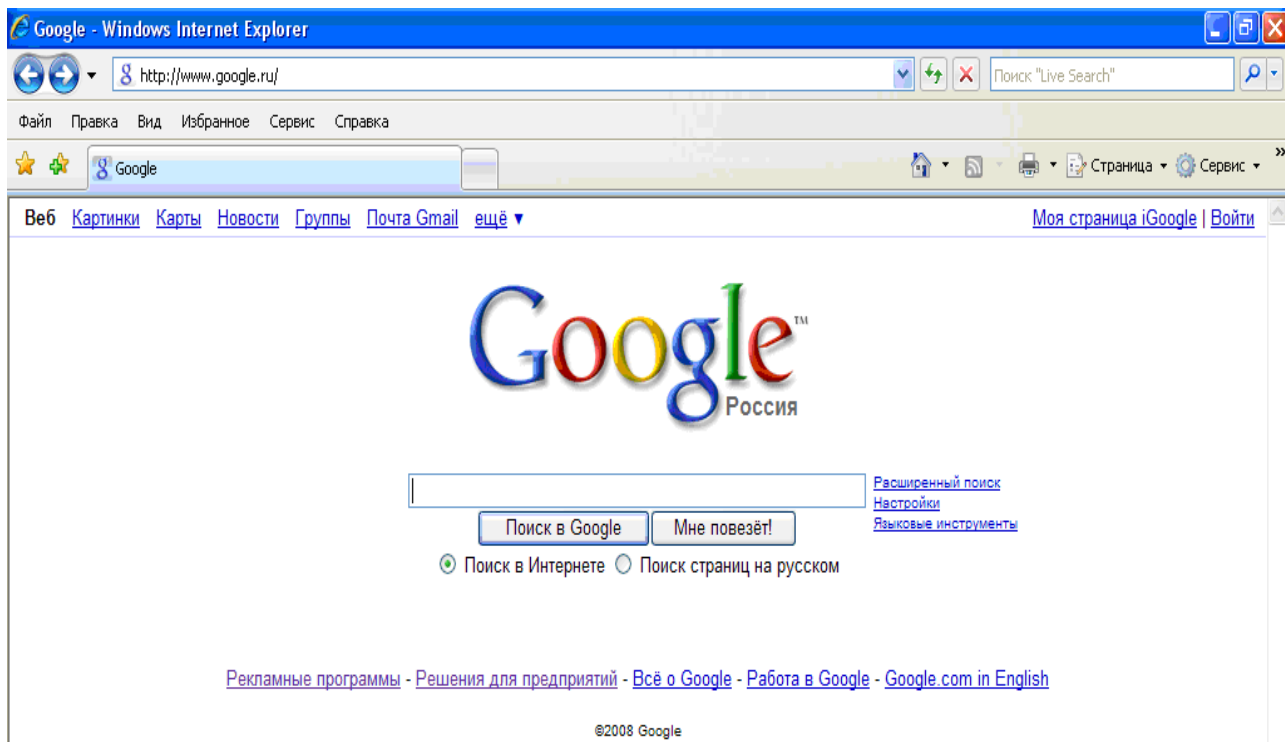


Рис. 40. Начальная страница поисковой системы Google

тической, и инициатором регистрации является владелец ресурса. Для регистрации в каталоге необходимо либо заполнить определенную форму, либо послать запрос, содержащий указание, в какой раздел вы хотите поместить свою



Рис. 41. Начальная страница каталога Yahoo!

страницу, краткое описание сайта и список ключевых слов для поиска страницы в каталоге.

Самым широко известным каталогом является Yahoo! Он существует с конца 1994 года, содержит порядка миллиона *Web*-страниц и считается самым популярным и старейшим из основных каталогов Интернета (www.yahoo.com). Начальная страница каталога Yahoo! приведена на рис. 41.

Среди каталогов российской части сети можно выделить STARS.RU (www.stars.ru), List.RU (www.list.ru), The List of Russian Web Servers (www.weblisr.ru) и Russia on the Net (www.ru), начальная страница которого приведена на рис. 42.

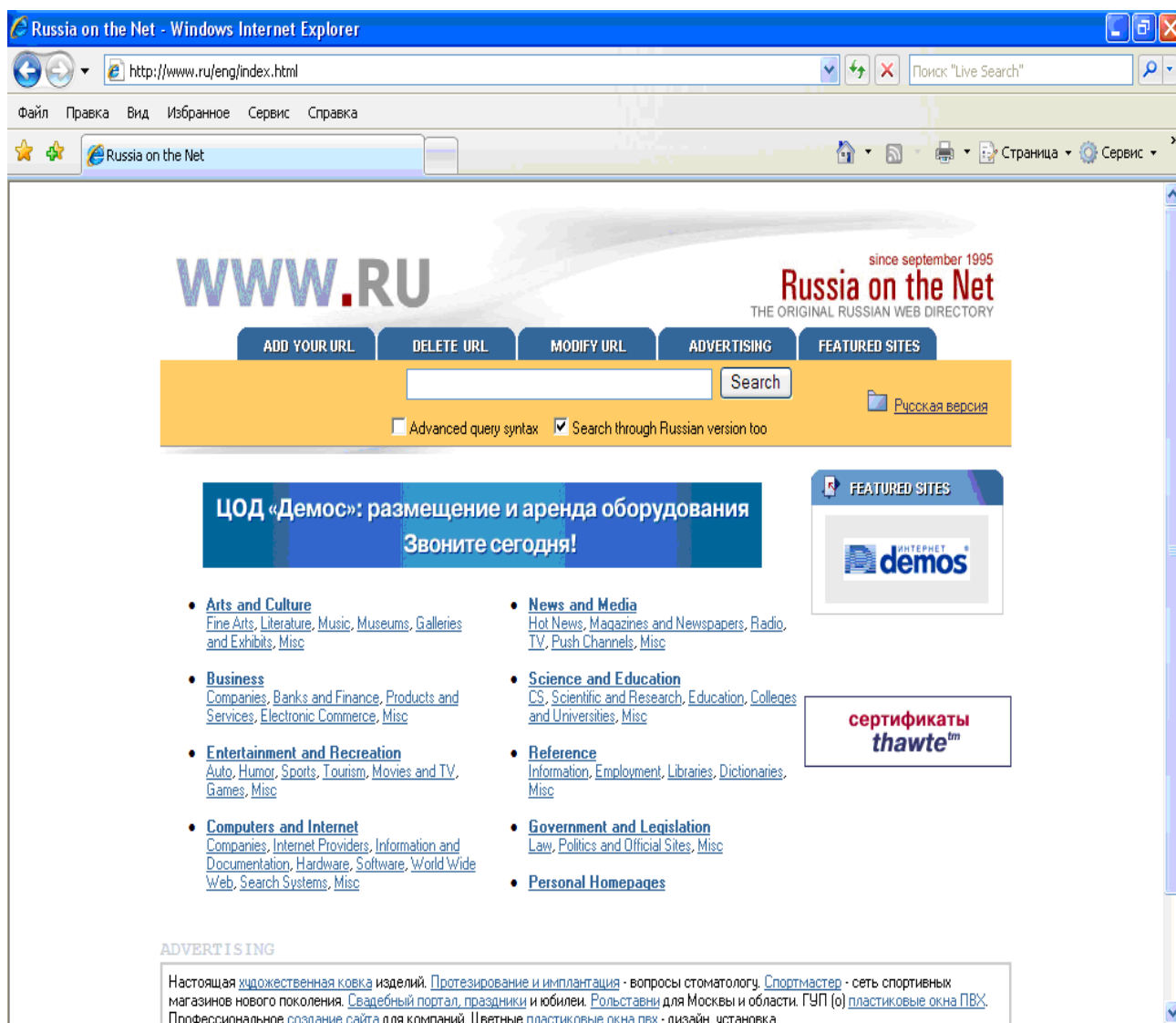


Рис. 42. Начальная страница каталога Russia on the Net

Некоторые поисковые системы имеют при себе и каталог. Индексы для самой поисковой системы добываются поисковой машиной, а каталог пополняется через регистрационную форму или модераторами системы. Примерами таких поисковых систем являются уже упоминавшиеся выше системы Excite (www.excite.com), Lycos (www.lycos.com), WebCrawler (www.webcrawler.com).

Порталы, или иницирующие сайты. Последнюю группу серверов управления трафиком в Интернете представляют иницирующие сайты. Если цель поисковых систем или каталогов состоит в поиске информации и последующем перенаправлении пользователей к найденным ресурсам в соответствии с запросом, то иницирующие сайты используют более комплексный подход и наряду с функциями поиска информации и перемещения к ней пользователей предоставляют достаточно широкое по тематике информационное содержание. Основная задача, которую они решают данной стратегией, состоит в том, чтобы как можно дольше удерживать пользователей именно на их сервере и только в крайнем случае перенаправлять их на внешние ресурсы. Таким образом, иницирующие сайты можно одновременно отнести как к поисковым средствам, так и к группе конечных сайтов. Наиболее крупные из иницирующих серверов получили название порталов.

Портал – это Web-сайт, предназначенный для специфической аудитории, который обеспечивает:

- △ объединение информационного наполнения и доставку важной для данной аудитории информации;
- △ совместную работу и коллективные услуги;
- △ доступ к услугам и приложениям для избранной аудитории, предоставляемый на основе строгой персонализации.

Порталы можно разделить на три вида: мегапорталы, вертикальные порталы и так называемые порталы «бизнес для бизнеса», или B2B-порталы.

Мегапорталами называются оригинальные порталы Интернета. Многие из них начинали как механизмы поиска и быстро выросли до мегапорталов, например, Yahoo!, Lycos и America Online. Их аудиторией является практически все Интернет-сообщество.

Вертикальные порталы являются порталами для специфических рыночных ниш. Иногда их называют «ворталы» – в(ертикальные п)орталы. Они обслуживают узкоспециализированные сообщества или рынки. Вертикальный портал существует практически для каждой аудитории, имеющей свою нишу в Интернете, и каждый такой рынок обычно имеет более одного вертикального портала.

Порталы типа «бизнес для бизнеса» (B2B) являются своего рода электронным рынком, который создан для того, чтобы предприятия могли взаимодействовать друг с другом или совершать общие деловые операции. Такие порталы предоставляют своим клиентам множество механизмов электронного бизнеса (например, закупки, проведение прямых и обратных аукционов).

Одним из самых популярных международных мегапорталов является Excite (www.excite.com), о котором уже речь шла как об одной из популярных международных поисковых систем. Так, портал Excite сочетает в себе отличные инструменты индивидуализации, превосходный механизм поиска с прогнозированием запросов пользователя и развитые средства организации сообществ.

Среди российских порталов одно из первых мест занимает информационный портал компании «Россия-Он-Лайн» (www.online.ru) – одного из ведущих российских поставщиков услуг Интернета.

Классификация Web-сайтов. Проведенный краткий обзор показал, что для использования ресурсов Интернета необходимо хорошо представлять существующие виды Web-сайтов, выполняемые ими функции, а также логику навигации по ним пользователей. По выполняемым функциям все Web-сайты можно разделить на две основные группы: *навигационные сайты* и *конечные сайты*.

Такое деление обусловлено общей логикой навигации пользователей в среде Интернета. Навигационные сайты нами рассмотрены выше. Общая же классификация Web-сайтов приведена на рис. 43.

Кратко рассмотрим вторую группу Web-сайтов.

Конечные сайты. В эту группу сайтов входят конечные «адресаты», предоставляющие разнообразные услуги, ради которых пользователи, собственно, и посещают Интернет. Этими услугами может быть предоставление данных, документации, финансовых сводок, информации о курсах акций, фирмах, их продукции и услугах, различные способы времяпрепровождения, продажа товаров и многое другое.

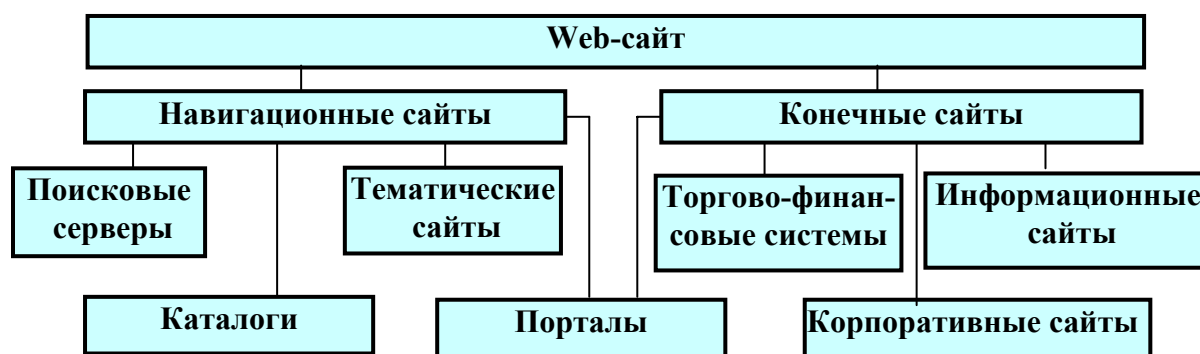


Рис. 43. Классификация Web-сайтов

Корпоративные сайты обеспечивают виртуальное присутствие фирм и их предложений в Интернете. Сайты этого типа можно разделить на два типа: информирующие серверы и рекламные серверы, хотя четкую границу между ними провести бывает довольно трудно. *Информирующие серверы* обеспечивают детальную информацию о фирме и/или ее предложениях. Такие серверы могут иметь довольно разнообразное строение, и для облегчения навигации в случае сложного строения сервера или большого объема представленной информации может использоваться функция поиска. *Рекламный сервер* может состоять из одной или нескольких Web-страниц и содержать различную рекламную информацию. Информация о продуктах или услугах фирмы обычно обеспечивается в том контексте, в котором она имеет значение для потребителя. Такой тип сервера особенно подходит для продуктов, которые не требуют представления большого количества информации, при этом основное взаимодействие на потребителей оказывается скорее в эмоциональном плане, чем в информационном.

Хорошим примером корпоративного сайта, выполняющего как информационные, так и рекламные задачи, является сайт одного из крупнейших в Санкт-Петербурге поставщиков услуг Интернета — компании «Петерстар» (www.peterstar.ru).

Информационные сайты. Название группы говорит само за себя – это серверы, главной своей целью ставящие предоставление той или иной информации пользователям Интернета. Наиболее характерное деление в группе можно провести по признаку необходимости оплаты для доступа к информации или для ее представления на Web-сервере.

Рассмотрим для начала платные серверы. В этом случае для доступа к информации пользователи платят определенную сумму. Возможен и другой вариант, когда для пользователей услуги бесплатны, а плата взимается с фирм за включение информации о них в базу данных Web-сервера. Основной проблемой эффективного функционирования Web-сервера такого типа является задача предоставления уникальной информации, поскольку Интернет является информационно насыщенной средой, где существует много альтернатив поиска требуемой информации. Наиболее характерным примером серверов, относящихся к этой группе, являются Web-серверы, предоставляющие, во-первых, финансовую информацию и, во-вторых, информацию о рынке, т. е. информацию о фирмах, их продукции и услугах с возможностью проведения выборок по регионам, отраслям и т. д.

Очень часто на сайте наряду с платным предоставлением информации реализуется и бесплатное, например, по отдельным тематикам. Таким образом, происходит совмещение двух вышеназванных моделей, которое способствует привлечению большего количества посетителей, формированию и продвижению имиджа компании, а также является дополнительным средством рекламы.

Например, среди российских серверов ведущее место по предоставлению финансовой информации занимает сервер РИА «РосБизнесКонсалтинг» (www.rbc.ru). Другим примером Web-сайта, предоставляющего как бесплатную, так и платную информацию по широкому спектру рыночных отношений, фирм и товаров, является Web-сервер Информационная коммерческая система «Международные информационные рынки» (www.icsmir.ru).

Торговые и финансовые системы представляют сайты, оказывающие пользователям различные услуги от продажи товаров до проведения финансовых операций. Эти сайты непосредственно реализуют деловые процессы категорий В2С (бизнес–коммерция) – продажа товаров конечным потребителям, С2С – взаимодействие между физическими лицами и В2В – ведение бизнеса между предприятиями.

Следует отметить, что выделенные функциональные категории коммерческих Web-сайтов позволяют продемонстрировать две основные задачи, стоящие перед их владельцами: привлечение посетителей на сервер и получение как можно большего числа повторных посещений сервера. Первоначальное посещение сервера зависит от группы серверов управления трафиком, благодаря которым пользователи могут узнать о существовании сервера. Однако дальнейший успех может быть достигнут только при условии значительного количества повторных посещений сервера пользователями за счет выполнения сервером функций второй группы, т. е. предоставления пользователям качественных услуг – информации, продажи продуктов или удовлетворения других их желаний и потребностей.

О мерах безопасности при работе в Интернет. В заключение хотелось бы еще раз напомнить пользователям Интернета о тех неприятностях, которые могут возникнуть при несоблюдении мер безопасности при работе с информацией из Сети (см. лекцию 13). Вопросы безопасности и ответы на них очень хорошо освещены в справочной системе браузера *Internet Explorer* версии 7.0, где их можно посмотреть. Для этого нужно открыть окно этой программы, (рис. 44). В случае какой-либо опасности появляется **Панель информации** с предупреждением. Нужно щелкнуть мышкой на панели информации, а затем – на любой из команд *Факторы риска* или *Подробнее* в появившемся контекстном меню. При открытии веб-документа в случае опасности также может появиться окно *Загрузка файла*, в котором имеется предупреждение о возможной опасности при открытии этого файла. В этом случае нужно щелкнуть гиперссылку *Факторы риска* (рис. 44), которая аналогична команде из контекстного меню.

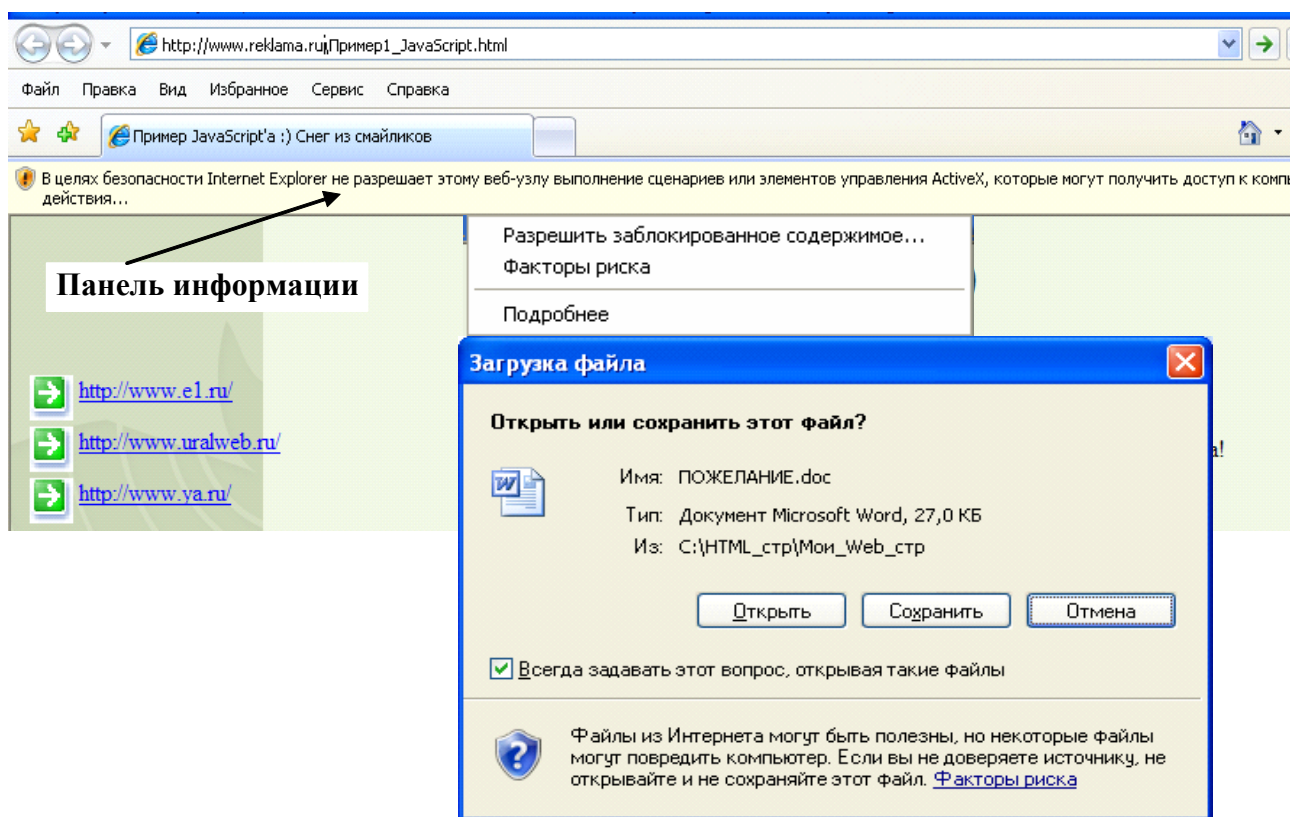


Рис. 44. Окно Internet Explorer 7.0 в случае какой-то опасности

После этого появится новое окно Internet Explorer, разделенное по вертикали на две части (рис. 45).

В левой части на вкладках *Содержание* и *Указатель* перечислены темы, касающиеся работы пользователей в Интернет, в частности и меры безопасности, а в правой – появляется список вопросов в виде гиперссылок по выбранной теме. Список можно расширить, щелкнув на гиперссылке *См. также*. Появляется дополнительное окно *Найденные разделы*, в котором можно выбрать вопросы из расширенного списка. При щелчке на любом вопросе появляется подробный ответ на данный вопрос.

Здесь же начинающим работать с Интернетом имеются полезные сведения и советы по поиску нужных веб-страниц, по работе с избранным в веб-обозревателе, по настройке домашних страниц, по распечатке и сохранению данных страницы и многое другое. Опытных пользователей заинтересует информация по настройке и надстройкам веб-обозревателя, специальным его возможностям и т. д.

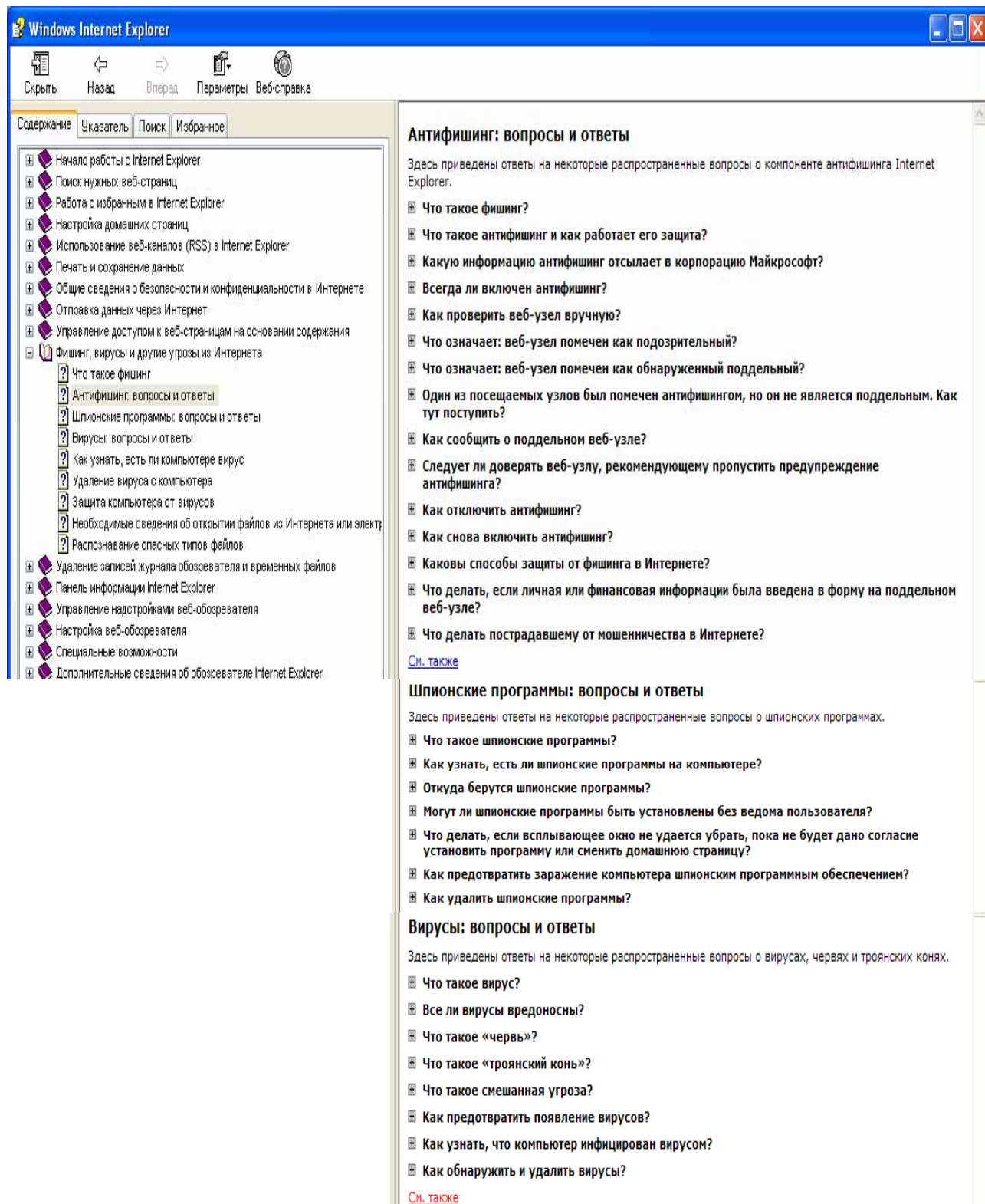


Рис. 45. Окно для поиска вопросов и ответов по безопасной работе с программой Internet Explorer 7.0

ЗАКЛЮЧЕНИЕ

В рыночных отношениях, к которым перешла Россия, своевременность, достоверность и полнота информации являются эффективным средством маркетинговой, финансово-кредитной, инвестиционной и другой деятельности. Современный период жизни общества характеризуется небывало высоким ростом объема информации. К известным ранее видам ресурсов – материальным, трудовым, энергетическим, финансовым – прибавился новый, ранее не учитываемый вид – информационный ресурс. Только на основе своевременного пополнения, накопления, переработки информационного ресурса, т. е. владения достоверной, полной и своевременной информацией, возможно рациональное управление любой сферой человеческой деятельности, правильное принятие решений. Основная задача информатики – создание материально-технической базы для удовлетворения информационных потребностей промышленной и деловой сфер, органов государственного управления и других областей человеческой деятельности.

Современная материально-техническая база информатики позволяет широко использовать автоматизированные рабочие места (АРМ) специалистов во всех сферах и на различных уровнях управленческой деятельности, создавать вычислительные системы, которые в пределах специализированной предметной области способны принимать решения на уровне экспертов-профессионалов, и информационно-коммуникационные сети, формируемые на базе ЭВМ и систем передачи данных. Особенностью вычислительных сетей является не только приближение аппаратных средств непосредственно к местам возникновения и использования данных, но и разделение функций обработки и управления на отдельные составляющие с целью их эффективного распределения между несколькими ЭВМ, а также обеспечение надежного и быстрого доступа пользователей к информационно-вычислительным ресурсам и организация коллективного использования этих ресурсов. Последнее особенно ярко проявляется в Интернете.

Влияние Интернета коснулось как внешних отношений между компаниями и их партнерами или клиентами, так и внутренней структуры самих компаний. Появились не только новые направления ведения бизнеса, но и принципиально изменились уже существующие. Исследования показывают, что компании, недоучитывающие потенциал электронного бизнеса и электронной коммерции, могут утратить свои коммерческие позиции под натиском более способных к адаптации в условиях новых реалий рынка конкурентов. Чтобы такого не произошло, необходимо чутко следить за развитием современных информационных технологий, стараться быть на шаг впереди рынка по использованию инноваций. Современное состояние электронной коммерции находится только на начальном этапе своего развития, но уже сегодня путь, пройденный за последние годы, захватывает дух и заставляет задуматься о том, куда придет мир в ближайшем будущем. Хочется надеяться, что Россия выберет правильный путь

и ее развитие будет определяться объективными факторами развития мировой экономики, в которой электронные формы взаимодействия с каждым годом играют все более значимую роль и становятся обязательными элементами для интеграции страны в мировую экономику.

В условиях перехода к рыночным отношениям интенсивно развиваются новые формы организации труда, производственных и межличностных отношений, растет потребность в разнообразной информации и, в частности, в оперативных сведениях коммерческого и правового характера. Все эти изменения требуют, чтобы будущие специалисты с высшим образованием, являясь основными пользователями компьютерных информационных систем, были готовы к работе в новых условиях, владели основами информационных технологий, умели оценивать действия информационных систем и использовать телекоммуникационные сети для ведения рекламной, торговой, финансовой и другой деятельности в режиме реального времени.

Приложение. **КРАТКИЙ ПОНЯТИЙНО-ТЕРМИНО- ЛОГИЧЕСКИЙ СЛОВАРЬ КУРСА**

Адрес – числовое значение (номер ячейки) обычно в шестнадцатеричной системе счисления, определяющее местонахождение данных в оперативной памяти (RAM) компьютера.

Адрес компьютера в Интернет – уникальный идентификатор физической сети, к которой подсоединен компьютер, и уникальный идентификатор самого компьютера. Адрес характеризует не отдельную машину, а ее сетевое соединение. Пользователи в основном работают не с адресами компьютеров, что затруднительно, а на логическом уровне с их именами, представляющими иерархическую доменную структуру, где начало определяет часть сети, к которой подключен компьютер, а конец – номер самого компьютера. Использование имен обеспечивается при помощи распределенной базы данных – системы доменных имен (*Domain Name System, DNS*), которая хранит таблицы соответствия имен и числовых *IP*-адресов на серверах *DNS*.

Администратор базы данных – сотрудник предприятия, отвечающий за администрирование баз данных архитектуры клиент/сервер и обладающий правами (разрешениями) использовать все команды, которые только могут быть выполнены в СУРБД. Он отвечает за систему безопасности, включающую контроль доступа пользователей к СУРБД и выполнение функций резервирования и восстановления.

Активный объект – в Windows функционирующее в данный момент приложение, или окно, в которое осуществляется ввод.

Алгоритм – точное, сформулированное на определенном языке конечное описание, определяющее содержание и последовательность *исполнительных* элементарных операций обработки, переводящих исходные данные в искомый результат.

Арифметический оператор – оператор, выполняющий арифметические действия: сложение (+), вычитание (–), умножение (× или ·), и деление (: или /).

Архиватор – программа, позволяющая объединять несколько файлов с различной информацией в одном сжатом файле, т. е. осуществляющая упаковку и распаковку совокупности информации.

Архитектура компьютера – его описание, включающее описание пользовательских возможностей программирования, системы машинных команд, системы адресации к памяти и периферийным устройствам, организации памяти и др. Другими словами, – это стандарты на устройства и их подключение, на программы управления устройствами, на BIOS, операционные системы и т. д.

Архитектура компьютера многопроцессорная – архитектура компьютера при наличии в нем нескольких процессоров, что позволяет параллельно обрабатывать несколько потоков данных и несколько потоков команд.

Аутентификация – процесс проверки имени (идентификатора) пользователя и его пароля при регистрации пользователя в системе (сети).

База данных – совокупность взаимосвязанных, хранящихся вместе данных при наличии такой минимальной избыточности, которая допускает их использование оптимальным образом. Базы данных используются для накопления, хранения и выдачи данных для различных отчетов, справок, запросов и других документов. Наиболее распространены *реляционные* базы данных. Это многомерные базы, информация в них хранится в большом количестве различных таблиц, связанных между собой с помощью определенных признаков (ключевых полей). Реляционные базы данных создаются и управляются специальными программами, которые носят название *Система управления реляционными базами данных* (СУРБД).

Базовая система ввода-вывода (BIOS-Basic Input/Output System) – комплекс небольших программ, записанных в микросхемы энергонезависимых стираемых и программируемых постоянных запоминающих устройств (ПЗУ). При включении ПК эти программы

запускаются первыми, еще до загрузки операционной системы. Они предназначены для проверки устройств ПК при включении питания (тестирование памяти, системной платы, видеоадаптера, клавиатуры, контроллера жесткого магнитного диска и др.). В частности, в BIOS хранится небольшая программа поиска и загрузки с диска в оперативную память операционной системы (ОС). После загрузки ОС BIOS вместе с процессором обеспечивает прикладным программам выполнение операций обмена информацией между устройствами ПК. Эти программы записываются на фирме-изготовителе ПЗУ.

Баннер (Banner – фантик) – рекламный носитель, представляющий собой маленький графический файл (GIF, JPEG), помещаемый на Web-страницу издателя и имеющий ссылку на Web-сайт рекламодателя.

Библиотека динамической компоновки (Dynamic Link Library-DLL) – файл, содержащий набор функций Windows для выполнения операций определенного класса. Большинство библиотек DLL имеют расширение .dll, однако некоторые из них, как, например, Gdi32.exe, имеют расширение .exe. Функции, входящие в DLL, при необходимости вызываются (*активизируются*) для выполнения требуемых действий по запросу приложений.

Бит – эталонная единица *объемного* способа измерения количества информации. Это элементарная информация, передаваемая сообщением об одном из двух равновероятных исходов события, самая маленькая единица информации, обрабатываемая компьютером. Происходит от сокращения двух английских слов – **B**inary **digi**T – двоичное число. Бит может принимать только два значения: 1 (интерпретируется как «истина») или 0 (интерпретируется как «ложь»). Производные единицы от эталонной: байт = 8 бит, слово = 16 бит, килобайт = 2^{10} = 1024 байта, мегабайт = 2^{20} = 1024 килобайта = 1048576 байт и т. д. Существует также энтропийный способ измерения информации. В вычислительной технике он не используется.

Брандмауэр – программно-аппаратный комплекс, используемый для отделения сети *Интранет* от глобальной сети и обеспечения ее безопасности.

Браузер – программа, снабженная графическим интерфейсом пользователя и предназначенная для просмотра Web-страниц. Браузер дает возможность просматривать содержание узлов сети и перемещаться от одного узла к другому. Используемые в настоящее время браузеры дополнительно включают в себя средства защиты. Из всех браузеров наиболее широко используется *Microsoft Internet Explorer*.

Булева алгебра – *алгебра логики*, являющаяся одной из основных частей математической логики, основанной на применении алгебраических методов к формальной логике. Названа в честь Джорджа Буля, английского логика и математика. Является одним из основных инструментов в теории релейных схем, в теории ЭВМ и дискретных автоматов, в прикладной информатике и кибернетике, в программировании и т. д. Использование аппарата алгебры логики в теории устройств дискретного действия открывает широкие возможности для анализа и синтеза переключательных схем, являющихся основой построения всех компьютеров.

Буфер обмена – специально выделенная операционной системой область оперативной памяти ПК для временного хранения фрагментов текста, графических объектов, элементов управления, форм, таблиц и т. д. при их копировании или вырезании. Копируемый или вырезанный фрагмент из приложения помещается в буфер во всех форматах, которыми обладает приложение. Объекты из буфера обмена могут быть вставлены в любой документ в указанное место в том формате, который больше всего подходит приложению-приемнику. Это выполняется по команде *Специальная вставка* из пункта меню *Правка*. В новых версиях Windows буфер обмена позволяет хранить одновременно и независимо друг от друга от 12 до 24 объектов.

Видеоадаптер – электронная плата, обрабатывающая видеоданные и управляющая работой дисплея. Видеоадаптер определяет разрешающую способность дисплея и количество цветов. Он содержит оперативную видеопамять, в которой формирует видеоизображение, выводимое на экран, регистры ввода-вывода и модуль BIOS. Видеоадаптер посылает в дисплей сигналы управления яркостью лучей и сигналы развертки изображения.

Вирус – программный код, обладающий способностью к распространению (возможно, с модификацией) путем внедрения в другие программы.

Вкладки – подразделы диалоговых окон, которые открываются щелчком мыши на их ярлычках. Предназначены для расширения возможностей диалоговых окон.

Внедрение – вставка объекта, не связанного с исходным документом, в новый документ. Чтобы отредактировать внедренный документ, достаточно дважды щелкнуть на нем мышкой. Внедренный объект будет открыт в исходном приложении.

Восприятие информации – процесс преобразования сведений, поступающих в техническую систему или живой организм из внешнего мира, в форму, пригодную для дальнейшего использования.

Встроенные функции – готовые к использованию функции, которые включены в состав приложения, например в Excel (более 200) или в языки программирования высокого уровня, которые не нужно создавать как пользовательские.

Выражение – последовательности операций вычисления некоторых значений переменных, например: $X = S + 2.7 \cdot \sin(I) - 127.64$. Выражения строятся из констант и переменных с помощью различных операторов и обозначений стандартных функций.

Гетерогенная сеть – компьютерная сеть, содержащая входящие в нее программно несовместимые (разнородные) компьютеры (персональные компьютеры, мини-ЭВМ, мэйнфреймы). Как правило, это большие корпоративные сети, объединяющие несколько различных сетей, создаваемых в разные периоды времени.

Графический акселератор – специализированный графический сопроцессор, предназначенный для преобразования двумерных изображений в трехмерные, моделирования движущихся источников света и для других операций над изображениями. Обычно акселератор устанавливается на плате графического адаптера. Применение графических акселераторов освобождает центральный процессор от большого объема операций с видеоданными и увеличивает эффективность видеосистемы.

Двоичный файл – файл, в котором не содержатся строки текста. Выполняемые файлы (имеющие расширение .exe или .com), библиотеки динамической компоновки (.dll) и большинство файлов баз данных хранятся в двоичном формате.

Дефрагментация диска – размещение файла на диске не сплошным массивом, а в виде отдельных не связанных участков (пронумерованных кластеров).

Диалоговое окно – всплывающее дочернее окно в окне приложения, запрашивающее у пользователя информацию. Большинство команд из пунктов меню приложений открывают диалоговые окна с сообщениями, для ввода информации, для настройки приложения, открытия или запоминания документа и т.д.

Динамическая Web-страница – страница, содержание которой зависит от встроенных элементов и действий пользователя, просматривающего страницу в *браузере*.

Домен – в компьютерной сети доменом является группа рабочих станций и серверов, которые совместно используют общую базу данных диспетчера учетных записей (Security Account Manager – SAM) и позволяют пользователям регистрироваться для доступа к любому ресурсу данного домена с помощью идентификатора пользователя и пароля.

Драйверы устройств – программы, обслуживающие обмен информацией с внешними устройствами на уровне портов ввода-вывода и оформленные в соответствии с требованиями операционной системы. Эти программы всегда являются строго специализированными. Они жестко "привязаны" к аппаратуре, к конкретному оборудованию и учитывают все его особенности и возможности работы.

Заглушка – в программировании процедура или функция без программного кода, которая ничего не делает. В VBA заглушка состоит только из двух строк Sub *ИмяПроцедуры* и End Sub или Function *ИмяФункции* и End Function.

Загрузка – процесс перемещения с внешних устройств в оперативную память компьютера данных для обработки и программ для выполнения. Обмен данными центральный процессор может осуществлять только с оперативной памятью. Любые другие устройства по

отношению к процессору являются внешними, поэтому данные для обработки и программы для выполнения с внешних устройств должны быть загружены в оперативную память.

Запись – понятие записи используется в практике баз данных. Это логический эквивалент полной строки таблицы, включающий все поля (столбцы) этой строки. Поля записи должны быть поименованными.

Запрос – объект базы данных, который извлекает из таблиц информацию для просмотра, редактирования или анализа. Запросы могут быть на простую выборку данных, на вычисляемые поля в таблице, на заданный параметр или дату, на обновление, перекрестный и т. д. На основе результатов запроса можно создавать формы, отчеты, таблицы, страницы доступа к данным и др.

Запрос прерывания (IRQ-Interrupt ReQuest) – команда *INT n*, где *n* – номер прерывания, выставляемая центральному процессору для временного прерывания выполняемой им работы с целью выполнения неотложной другой работы, например, для приема символов, вводимых с клавиатуры, или обработки ситуации, когда встречается операция деления на нуль.

Защита информации – комплекс мероприятий, направленных на обеспечение сохранности информации от случайного разрушения или несанкционированного доступа с помощью шифрования, паролей, репликации и других мер безопасности.

Звездообразная топология – структура построения локальной вычислительной сети, в центре которой размещается коммутирующее устройство, обеспечивающее жизнеспособность системы.

Зеркальные диски – технология защиты данных на аппаратном уровне путем создания двух или более физических дисковых накопителей с абсолютно идентичной информацией. Запись данных в такой системе производится на все диски устройства зеркального набора одновременно. При выходе из строя одного дискового накопителя информация не теряется, и система может продолжать работать дальше.

Знак – минимальная, неделимая единица информации, выражающая определенное суждение о предмете, явлении или смысл чего-то. Новые знаки появляются тогда, когда назревает социальная необходимость в них. Например, с усложнением дорожного движения появились специальные знаки, каждый из которых несет определенную информацию. Точно так же с появлением и развитием математики появились специальные знаки для выполнения различных математических действий. Примеров можно приводить очень много.

Идентификатор – синоним термина *имя*, обычно применяется для имен переменных и констант в программировании.

Именованный диапазон – группа ячеек в рабочем листе *Excel*, которой присвоено какое-либо имя.

Индекс – для массивов данных это позиция определенного элемента по отношению к другим. Для первого элемента – обычно 0. Для файлов баз данных или таблиц *индексом* называется специальная таблица в виде файла или части файла, в которой значение поля в индексируемом файле соотносится с номером записи.

Инкапсуляция – недоступность пользователя к внутренней структуре объекта при объектно - ориентированном программировании.

Интернет – общемировая совокупность самых различных компьютерных сетей, связанных между собой на основе многоуровневого стека протоколов *TCP/IP* в единую глобальную сеть. Эта сеть не создает никаких новых данных, не занимается их обработкой, а обеспечивает легкий и быстрый доступ к огромному множеству информационных данных, созданных разными исполнителями с помощью различных средств и программно-технических платформ.

Интерпретатор – программа-транслятор, переводящая текст программы, написанный на языке программирования высокого уровня, в машинный код (язык команд процессора). Интерпретатор обрабатывает и исполняет каждую команду в отдельности поочередно, как бы моделируя некий компьютер, при этом файл программы в машинных кодах не создается.

Интерфейс – средства взаимодействия клиента и сервера. Часто употребляемое выражение – *пользовательский интерфейс* – означает средства взаимодействия между пользователем и компьютером. Платы адаптеров и контроллеров периферийных устройств компьютера составляют интерфейс между системной шиной данных и периферийными устройствами. Их управляющие программы (драйверы) являются программным интерфейсом между операционной системой и платами адаптеров. Интерфейс между двумя разными компьютерными сетями называется *мост*.

Интранет – внутренняя сеть организации (корпоративная сеть), использующая технологии WWW, что позволяет снизить затраты на обучение сотрудников, уменьшить эксплуатационные расходы и легко подсоединить свою сеть к сети Интернет. Часто Интернет используют как транспорт между отдельными частями распределенной большой корпоративной сети (например, сети железной дороги, раскинувшейся на сотни и тысячи километров). Для обеспечения безопасности используют программно-аппаратные комплексы – *брандмауэры*, которые проверяют допустимость пакетов данных, наличие полномочий на передачу тех или иных данных через Интернет и т.д.

Информационный ресурс государства – научные теории, открытия, изобретения, сведения о недрах земли и океанов, проекты новейшего оборудования и приборов, математические модели экономики и общества и многое другое. Это то, что определяет интеллектуальную мощь современного государства. Это *знания* многих поколений людей.

Информационная безопасность – защищенность информации и поддерживающей инфраструктуры от случайных или преднамеренных воздействий естественного или искусственного характера, исключающая недопустимые риски ее уничтожения, искажения и раскрытия, которые могут нанести неприемлемый материальный ущерб субъектам информационных отношений, в том числе владельцам и пользователям информации и поддерживающей инфраструктуры.

Информационное общество – общество, структуры, техническая база и человеческий потенциал которого приспособлены для оптимального превращения знаний в информационный ресурс и переработки последнего с целью перевода пассивных форм его (книги, научные журналы, технические отчеты о выполненных научно-исследовательских работах, базы данных и т.д.) в активные формы (модели, алгоритмы, проекты, программы и др.).

Информатика – раздел фундаментальной естественной науки, изучающей законы и методы производства, переработки, хранения и распространения всех видов информации в человеческом обществе, природе и технических устройствах. На сегодняшний день информатика представляет собой комплексную научно-техническую дисциплину. Информатика превратилась *в особую отрасль народного хозяйства, науки, своеобразное и важное подразделение экономики* промышленно развитых стран.

Информатизация – процесс создания, развития и всеобщего применения информационных средств и технологий, обеспечивающих достижение и поддержание уровня информированности всех членов общества, необходимого и достаточного для кардинального улучшения качества труда и условий жизни людей.

Информация – совокупность сведений о всевозможных объектах, явлениях и процессах окружающего нас мира, это отражение реального мира. С практической точки зрения информация – это нечто абстрактное. Абстрактная информация всегда представляется в виде конкретного *сообщения*.

Информационная система – организационно упорядоченная совокупность документов, информационных технологий, в том числе с использованием средств вычислительной техники и связи, реализующих информационные процессы, т. е. процессы сбора, накопления, хранения, обработки и выдачи информации пользователям. Так трактует это понятие Федеральный закон об информации.

Информационная технология – процесс использования технических средств, методов и правил сбора, накопления, хранения, обработки и передачи данных для получения качественно новой (вторичной) информации или информационного продукта.

Информационный язык – искусственный язык, предназначенный для записи информации в запоминающие устройства информационной машины или системы, например двоичный язык для записи любой информации в компьютер (чисел, текста, рисунков и графиков, музыки и др.) использующий два знака – цифры 0 и 1.

Искусственный интеллект (*artificial intelligence*) – раздел информатики, связанный с разработкой интеллектуальных программ для компьютеров.

Исходный код – читаемая форма (текст) программного кода, который создается на языке программирования высокого уровня. Транслятором (компилятором или интерпретатором) исходный код преобразуется в объектный код на машинном языке, при необходимости объединяется (компоуется) с файлами библиотек языка и записывается в выполняемый файл в двоичной системе счисления в виде цифр 0 и 1.

Клавиши доступа – комбинации клавиш, обеспечивающие доступ к пунктам и командам меню, макросам, пунктам настроек и установок в диалоговых окнах без использования мыши. Обычно это комбинация $\langle Alt + клавиша \rangle$. Клавиша доступа к пункту меню, команде меню, пункту настройки или установки параметра в диалоговом окне и т.д. обозначается подчеркнутым символом в названии. Нажав и не отпуская клавишу *Alt*, можно открыть меню, выбрать команду, выбрать в открывшемся диалоговом окне нужную закладку или установить требуемые настройки и параметры. Перемещаться в диалоговом окне можно клавишей *Tab* или *Shift + Tab* в зависимости от направления.

Кластер – группа последовательных секторов на диске. Кластеры для разных дисков имеют разные размеры, т. е. разное количество секторов, входящих в них. Кластер – минимальная информационная единица для операционной системы при записи и считывании информации. Даже при записи файла размером всего 20 байт под него на диске будет отведен кластер, который может составлять несколько десятков Кбайт. См. также **Сектор**.

Кластеризация (кластерный подход к разработке многопроцессорных высокопроизводительных вычислительных систем) – серверная архитектура, эмулирующая многопроцессорный режим в результате соединения нескольких отдельных компьютеров для того, чтобы они совместно выполняли приложение.

Ключ – используется в базах данных. Поле или несколько полей, значения которых идентифицируют запись в таблице (однозначно определяют каждую запись базы данных). Таблицы в базе данных индексируются именно по ключевым полям. В поле, являющемся ключевым, каждый элемент данных должен иметь уникальное значение.

Кнопка управляющего меню приложения (документа) – небольшие прямоугольные кнопки с миниатюрным изображением приложения, размещенные в верхнем левом углу открытого окна приложения для вызова управляющего меню приложения (документа). Дважды щелкнув на кнопке, можно закрыть документ или приложение.

Кнопка выбора, или переключатель (*OptionButton*) – элемент управления, позволяющий выбирать предлагаемые варианты.

Кнопочная форма (*switchboard*) – окно с формой, позволяющее работать с объектами базы данных, не вникая в суть реальной базы данных.

Код символа – числовое представление любого символа в памяти компьютера, записанное в двоичной системе счисления и зависящее от установленной на компьютере операционной системы. За стандарт всех ПК принят разработанный в 1961 г. *American Standard Code for Information Interchange-ASCII* – "Универсальный стандартный код обмена информацией" для печатаемых, управляющих и специальных символов. В первой половине таблицы этого кода расположены 8-значные двоичные коды управляющих символов, десятичных цифр, прописных и строчных букв латинского алфавита, а во второй половине таблицы располагаются национальные символы конкретной страны. В Windows 95 и выше используются другие кодировки – *ANSI* и 16-разрядная *Unicode*. Windows имеет встроенные средства, позволяющие ей преобразовывать символы из одной кодировки в другую, так что пользователь этого практически не замечает, переходя от одной версии Windows к другой. Фирмой *IBM* в

мейнфреймах используется кодировка *EBCDIC* (*Extended Binary Coded Decimal Interchange Code* – расширенный код двоично-десятичного обмена).

Команда – синоним понятия «инструкция к выполнению». Определяет действие, которое должно быть выполнено компьютером.

Командная кнопка – элемент управления, имеющий вид кнопки (прямоугольника), щелчок мышью на котором запускает в работу макрос или программу.

Комментарий – пояснение команды, оператора, выражения и т. д. внутри исходного текста программы, не предназначенное для компиляции или интерпретации в исполняемый код. В *VBA* комментарии начинаются с символа «апостроф» ('), но могут, как во многих языках программирования, выделяться ключевым словом *Rem*.

Компилятор – программа-транслятор, переводящая текст программы, написанный на языке программирования высокого уровня, в машинный код (язык команд процессора) целиком и создающая файл, который может выполняться самостоятельно.

Компьютер – управляемое программами электронное устройство, предназначенное для автоматизации процессов сбора, хранения, обработки и передачи информации.

Конкатенация – объединение двух выражений, обычно строковых, чтобы образовать более длинное выражение. В *VBA* и *Jet SQL* оператором конкатенации является символ *&*, например: *Труд & умение & настойчивость = успех в жизни*. В *VBA* для конкатенации разрешается использовать оператор *+*.

Контекстное меню – меню, появляющееся при щелчке правой клавишей мыши на каком-либо выделенном объекте, либо при нажатии клавиши *Application*. Контекстное меню показывает список команд или действий, которые могут быть выполнены над выбранным объектом.

Конфигурация пользователя – система настроек рабочего стола, приложений, панелей инструментов и т. д. создаваемая пользователем. Она хранится в *профиле* пользователя и отображается всякий раз, когда пользователь входит в систему под своим именем и паролем.

Корзина – программа операционной системы, предназначенная сохранять удаляемые с дисков папки, файлы и другие элементы с целью их восстановления при необходимости. Если Корзина настроена на сохранение удаляемых объектов, то их можно восстановить до "очистки" корзины по специальной команде из ее контекстного меню. Корзина может быть настроена так, что она не будет сохранять удаляемые объекты. Удаляемые объекты с дискет и флэш-памяти в Корзине в любом случае не сохраняются.

Кэш (склад, тайник) – устанавливаемая в микросхеме совместно с центральным процессором (внутренний кэш) или между центральным процессором и оперативной памятью (внешний кэш) небольшого объема дополнительная очень быстродействующая память. В ней хранится как бы слепок (копия блоков) оперативной памяти, к которым в ближайшее время будет обращение процессора. Кэш значительно ускоряет процесс обработки данных. В качестве кэша может быть блок основной памяти, зарезервированный программой для временного хранения информации. Обычно в таком кэше хранятся данные из прочтенных или записанных файлов, чтобы ускорить к ним доступ при повторном обращении.

Логический диск – часть *физического* жесткого магнитного диска, оформленная как самостоятельный *виртуальный* (несуществующий) магнитный диск со своим именем. Выполняется это с помощью специальной программы разбиения диска на разделы перед его форматированием. Во-первых, так удобнее пользователю работать – на нескольких логических дисках удобнее структурировать расположение информации, чем на одном большого объема. Во-вторых, современные жесткие диски выпускаются больших объемов, а некоторые операционные системы поддерживают (могут работать) диски ограниченного размера, и если диск не разбить на разделы, то он будет использоваться не полностью.

Логический оператор – один из операторов алгебры Буля (алгебры логики): отрицание – *NOT*, логическое сложение – *OR*, логическое умножение – *AND* и др.

Логический уровень работы с компьютером – работа пользователя с устройствами компьютера через посредников в виде графического пользовательского интерфейса операци-

онной системы или команд-утилит и программ, написанных на языках высокого уровня без привязки к конкретным устройствам. В операционных системах Windows и других введен запрет на прямое обращение к аппаратным средствам, минуя *ПЗУ BIOS* и встроенные возможности самой операционной системы.

Логический тип данных (*Boolean*) – тип данных, переменные и константы которого могут принимать одно из двух взаимоисключающих значений *True* (истина) или *False* (ложь), 1 или 0. Логический тип данных используется также для определения класса операторов, результатом которых тоже может быть значение *True/False*.

Локальная вычислительная сеть (*ЛВС, или LAN – Local Area Network*) – сочетание компьютеров, кабелей, плат сетевых адаптеров, сетевой операционной системы и сетевых прикладных программ. Сетевые адаптеры принимают и передают сообщения, по кабелям эти сообщения распространяются, однако набор соединенных между собой компьютеров становится сетью только тогда, когда установлены *протоколы обмена* сообщениями или правила взаимодействия объектов сети.

Макрос – набор различных команд и функций мощного языка программирования VBA, встроенного в пакет Microsoft Office, объединенных в одну макрокоманду со своим именем. Сложность макроса может быть от простой последовательности команд и функций до самостоятельного большого приложения. Собственно, все действия в приложениях пакета Microsoft Office, выполняемые по командам меню, с помощью кнопок панелей инструментов или в диалоговых окнах, – это работа макросов.

Мантисса – составляющая часть числа, выраженного в экспоненциальной форме (в виде последовательности: мантисса – символ E – порядок числа, где E означает 10). Например, для числа 12,3E-4 мантисса равна 12,3, порядок числа равен -4, а само число $12,3 \cdot 10^{-4} = 0,00123$.

Массив – упорядоченная последовательность значений (элементов), хранящаяся в одной переменной. Массивы могут иметь несколько измерений. Обращение к элементу массива выполняется путем ссылки на имя переменной с указанием в скобках его индексов в каждом измерении, например: *strValue=strArray(3,2)*.

Машинный язык – набор команд, которые понимаются и могут быть выполнены аппаратными средствами компьютера и операционной системой (система команд компьютера).

Меню – набор команд, из которых пользователь может сделать выбор. Вид меню в Windows описывается в спецификации *Common User Access (CUA)*, разработанной компанией IBM.

Многозадачность – способность компьютера с одним процессором выполнять обработку данных нескольких задач с помощью системы прерываний, когда одна программа прерывает работу другой, выставляя процессору команду прерывания для выполнения каких-то действий в своих интересах. Чаще всего прерывания процессору выставляет операционная система.

Многопоточность – способность приложений допускать разбивку на ряд логически связанных задач, или потоки, которые выполняются независимо.

Модем (модулятор-демодулятор) – устройство, преобразующее информацию, представленную в компьютере в цифровом коде, в аналоговый (непрерывный) сигнал для передачи по каналу связи. На приемной стороне этот сигнал снова преобразуется в цифровой для ввода в принимающее устройство.

Мой компьютер – значок на Рабочем столе компьютера, при щелчке левой клавишей мыши по которому запускается специальная программа операционной системы, позволяющая просматривать файловую систему, управлять папками и файлами, перемещать, копировать, переименовывать и удалять отдельные элементы. При щелчке правой клавишей появляется возможность управлять компьютером, дисками, выполнять дефрагментацию дисков, их форматирование, просматривать события, журналы и оповещения производительности, диспетчер устройств и т. д.

Монитор с сенсорным экраном – монитор, на экране которого выбор объектов производится не указателем мыши, а пальцем руки, как человек поступает в обыденной жизни.

Музыкальный инструментальный цифровой интерфейс (MIDI-musical instruments digital interface) – формат звуковых файлов для воспроизведения на устройствах MIDI. MIDI-файлы – не просто запись звука на диске. Это инструкции о нотах, длительности их звучания, инструментах для их воспроизведения и т. д.

Мультимедиа – сложная сумма информационных технологий, позволяющих вводить в компьютер, обрабатывать, хранить, передавать по каналам связи и выводить на экран одновременно такие типы информации, как текст, графика, анимация, звук, речь и т. д.

Мультипроцессорная обработка – способность компьютера с двумя или более процессорами распределять задачи (потоки команд) между процессорами. Симметричная мультипроцессорная обработка (SMP), реализованная в Windows, позволяет выполнять решение задачи на любом свободном процессоре, при этом приложения должны поддерживать многопоточность.

Наследование – возможность использования ранее созданного класса (класс-потомок), который будет наследовать все свойства ранее созданного класса с добавлением новых свойств при объектно – ориентированном программировании.

Немодальное окно – окно, которое пользователь может закрыть или свернуть, не выполняя в нем никаких действий.

Носитель информации – какой-то материальный объект, хранящий или передающий информацию. В качестве носителей информации могут быть объекты любой физической природы (электрические заряды, напряжение, ток, механическое движение, звуковые волны, намагниченные поверхности и т. д.).

Операторы – зарезервированные символы (или слова), которые выполняют действия над переменными и константами для получения результата. Обычные математические знаки: сложить +, вычесть –, умножить ×, разделить $:(/)$ – являются *арифметическими операторами*. Знаки >, >=, <, <=, < >, = являются *операторами сравнения* – сравнивают значения двух операндов и возвращают значения *True* (истина) или *False* (ложь) в зависимости от сравниваемых значений. Оператор может выполнять действия над одной переменной, тогда он имеет *унарную* форму (оператор с одним операндом), например $Var = - Var$. В данном случае унарный минус меняет знак переменной *Var* с «плюс» на «минус» или с «минус» на «плюс», т. е. инвертирует знак переменной. Операторы могут быть и *логическими* And (логическое умножение), Or (логическое сложение) или Not (отрицание). В программировании часто операторами называют команды программного кода.

Операнды – переменные или константы, объединяемые операторами. В приведенном примере: $3 \times 5 - 9 = 6$ числа 3, 5, 9 и 6 являются операндами, а знаки ×, – и = являются операторами.

Операционная система – комплекс программ для управления компьютером. Они запускают в работу пользовательские программы, распределяя им ресурсы компьютера и управляя их работой, обеспечивают защиту доступа к данным, обмен данными между устройствами компьютера, простое и легкое общение пользователя с компьютером (пользовательский интерфейс) и т. д.

Оперативная память – системное (основное) электронное быстродействующее запоминающее устройство, непосредственно с которым работает центральный процессор, поэтому любая программа для выполнения или данные для обработки должны быть загружены (перенесены с внешних устройств) в нее. Эта память энергозависимая – после выключения питания компьютера вся информация в ней теряется.

Открытая архитектура – архитектура компьютера, на которую опубликованы спецификации и стандарты, что позволяет на их основе другим производителям во всем мире разрабатывать разнообразное периферийное оборудование, комплектующие устройства компьютеров, программное обеспечение и т. д.

Панель задач – видимый элемент оболочки Windows, обеспечивающий пользователю доступ к главному меню системы и к работающим в данный момент программам (прямоугольная панель, обычно располагающаяся в нижней части Рабочего стола). На панели задач

находятся кнопки *Пуск (Start)*, кнопки всех выполняющихся программ и открытых документов, а также вспомогательные кнопки, например текущего времени, установленного в данный момент языка и др.

Панель управления – диалоговое окно, открывающееся из меню кнопки Пуск, содержащее группу программ Windows, с помощью которых осуществляется модификация и настройка параметров оборудования, операционной системы, программного обеспечения, установка даты и времени, установка на компьютер нового программного обеспечения и т.д.

Панель инструментов – группа кнопок дублирующих команды меню, обычно горизонтально выстроенных вверху окна приложения, предназначенных для ускорения доступа к командам приложения. Панели инструментов открываются из контекстного меню при щелчке правой кнопкой мыши в любом месте любой панели инструментов, легко перемещаются мышью и могут располагаться в любом удобном месте экрана. Пользователь может создавать свои панели инструментов для удобства и ускорения работы на соответствующей вкладке диалогового окна, открываемого из пункта меню *Сервис* по команде *Настройка* путем перетаскивания нужных кнопок с вкладки *Команды* на созданную панель.

Папка (каталог) – поименованная простая таблица или более сложная структура (контейнер) для регистрации и хранения сведений о расположении файлов и других папок на дисках (аналогично обычной папке для хранения различных документов). Папка создается из контекстного меню при щелчке правой клавишей мыши по команде *Создать папку*. Корневой каталог создается при форматировании диска, имеет имя в виде \ (косой черты), которое невозможно изменить, и ограничен по количеству регистрируемых файлов и папок.

Параметр сигнала – характеристика сигнала, служащая для представления информации. Например, в радиопередачах в диапазоне средних волн сообщения передаются путем изменения амплитуды колебаний, а в диапазоне ультракоротких волн – путем изменения частоты колебаний. В первом случае на СВ параметром сигнала является амплитуда, а во втором случае на УКВ – частота колебаний.

Пароль – уникальный набор символов, который необходимо набрать при доступе в сеть, к электронной службе, защищенной папке или файлу и др. Пароли используются для защиты доступа к ресурсам сети или локального компьютера.

Периферийное устройство – внешнее устройство, подключаемое к компьютеру для выполнения определенных функций, например, принтер для распечатки выводимой из компьютера информации, звуковые колонки для воспроизведения музыкальных файлов, клавиатура для ввода символьной информации, накопитель на гибких магнитных дисках для записи информации на дискету и т. д.

Плоская база данных – простая база данных, состоящая из одной таблицы, например в Excel.

Подсистема информационной системы – часть информационной системы, выделенная по какому-либо признаку. Различают функциональные и обеспечивающие подсистемы. Функциональные подсистемы предназначены для решения определенных задач, например, подсистема бухгалтерского учета или подсистема сбыта продукции. Обеспечивающие подсистемы независимо от сферы применения обеспечивают нормальную работу информационной системы. К ним относятся подсистемы информационного, технического, математического, программного, организационного и правового обеспечения.

Полиморфизм – задает правила взаимодействия со средой разработки и другими компонентами при объектно – ориентированном программировании.

Портал – Web-сайт, предназначенный для специфической аудитории, который обеспечивает объединение информационного наполнения и доставку важной для данной аудитории информации, совместную работу и коллективные услуги, доступ к услугам и приложениям для избранной аудитории, предоставляемый на основе строгой персонализации.

Потерянный кластер – кластер, помеченный как фрагмент файла, но не связанный ни с одним файлом на диске. Он занимает место на диске, но его нельзя найти в списках

файлов в окне Проводника или в окне Мой компьютер. Такая ситуация возможна при сбое записи информации на диск или сбое питания при записи и т. д.

Преобразователь аналого-цифровой и цифро-аналоговый – устройства, преобразующие аналоговые сигналы в дискретные (АЦП) и наоборот (ЦАП). Учитывая преимущества дискретных сигналов, с помощью АЦП любой физический процесс может быть представлен посредством дискретных сигналов либо смоделирован с их помощью. Эти преобразователи используются на звуковой карте для преобразования формы звука.

Приложение – комплекс программ, который предоставляет пользователю на компьютере удобную, «дружественную» среду для создания документа определенного типа (например, *Word, Excel, AutoCAD* и др.).

Проводник Windows – программа операционной системы, позволяющая открывать и просматривать диски, папки, файлы, а также осуществлять их копирование, перемещение, удаление, запуск программ и т.д. Проводник проще всего запускать из контекстного меню при щелчке правой клавишей мыши на кнопке *Пуск*.

Протокол – правила передачи данных и взаимодействия компьютеров в сети. Как правило, протоколы состоят из нескольких уровней и взаимодействуют на строго иерархической основе по принципу «снизу вверх».

Путь – расположение файла или папки в файловой системе компьютера. Путь указывает имя диска, через косую (\) перечисление подчиненных папок и имя файла. Путь к файлу на другом компьютере сети включает в себя также и имя этого компьютера.

Рабочая станция – любой компьютер, включенный в компьютерную сеть, за исключением серверов.

Рабочий стол – программа операционной системы, после своей загрузки выводящая на экран монитора окно, в котором могут отображаться Панель задач, ярлыки или значки различных объектов для быстрого доступа к ним, фоновый рисунок и т. д. Если после загрузки ОС не запущена ни одна программа, то окно Рабочего стола выводится постоянно в цикле. Пользователь может Рабочий стол настроить по своему желанию.

Разрешение экрана – разрешающая способность монитора. Это размер экрана монитора по ширине и высоте в пикселях, например 1024 × 768.

Режим предварительного просмотра – режим просмотра документа на экране монитора в том виде, в каком он будет напечатан.

Резервное копирование – создание архивных копий информации с жестких несъемных дисков на устройства с магнитной лентой (стримеры) или съемные жесткие магнитные диски. Резервное копирование желательно выполнять всегда для сохранности информации.

Реляционная база данных – сложная база данных, включающая несколько связанных между собой таблиц, причем данные могут извлекаться из нескольких таблиц одновременно.

Ресурсы, или службы, Интернет – совокупность информационных технологий и баз данных, доступных при помощи этих технологий, использующих различные протоколы, построенные на базовых протоколах *TCP/IP* (электронная почта, телеконференции, передача файлов по протоколу *FTP*, система чатов, мировая виртуальная файловая система – *WWW* и др.).

Световое перо – светочувствительное устройство снятия координат точек экрана, используемое для ввода данных. Не требует специального экрана. В наконечнике светового пера установлен фотоэлемент, который реагирует на световой сигнал, передаваемый экраном в точке прикосновения пера. Сопоставление времени появления сигнала с синхросигналом развертки изображения позволяет определить положение светового пера на экране.

Сектор – часть дорожки диска, создаваемая при его форматировании и предназначенная для записи информации. Это наименьшая единица обмена информацией с диском, поддерживаемая контроллером. Контроллер накопителя может поддерживать длину сектора 128, 256, 512, 1024, 2048, 4096, 8192 или 16384 байта. Стандартно операционные системы используют для данных только сектора размером 512 байт, но для защиты данных диск можно отформатировать нестандартной программой форматирования на сектора различной длины.

Сенсорный экран – экран, совмещенный с сенсорными устройствами. Позволяет вводить в компьютер информацию прикосновением пальца руки. Сенсорными экранами оборудуют рабочие места операторов и диспетчеров, их используют в информационно-справочных системах и т. д.

Сервер – централизованный информационно-вычислительный ресурс, предназначенный для коллективного использования и обеспечивающий возможности совместной работы пользователей сети. Это может быть центральный компьютер, управляющий сетью, сетевой принтер, файл-сервер (сервер баз данных), WEB-сервер, телекоммуникационный сервер или сервер-приложение и т. д.

Сжатие данных – преобразование информации по определенному алгоритму в более компактную форму для уменьшения места при хранении на диске, для транспортировки или более эффективной работы.

Сигнал – материальное воплощение сообщения в виде, используемом в информационных системах при передаче, обработке и хранении информации. Это средство перенесения информации в пространстве и времени.

Сигнал аналоговый – непрерывный сигнал, воспроизводящий все мгновенные значения физической величины на отрезке времени его существования. Практически характеристики всех явлений реального мира описываются с помощью аналоговых сигналов (звук, световое излучение, давление, температура, электромагнитные волны и т. д.).

Сигнал дискретный (импульсный или цифровой) – сигнал, параметр которого может принимать лишь конечное число значений и существовать лишь в конечном значении моментов времени. Эти сигналы в настоящее время находят все большее применение в компьютерной технике, в радио и телефонной связи, в радиолокации, радиоизмерениях, в импульсных системах автоматического регулирования и управления, телевидении, радиотелеметрии, и т. д.

Символ – знак в алфавите. В экспертных системах – цепочка знаков, представляющая некоторое понятие реального мира.

Система счисления – совокупность приемов и способов обозначения (записи) чисел. Система счисления называется позиционной, если значение каждой цифры в числе зависит от ее позиции (разряда). Если такой зависимости нет, то система счисления называется непозиционной, например римская, в которой число 30 записывается в виде XXX – трех десятков, не зависящих от расположения в числе. В информационной технике используется позиционная двоичная система счисления с двумя цифрами – 0 и 1.

Слайд – один из сменяющихся кадров презентации. Является основной единицей презентации.

Слот – в устройствах компьютера – разъем. В экспертных системах – атрибут, связанный с узлом в системе, основанной на фреймах. Узел может означать объект, понятие или событие. Например, узел представляющий объект «*служащий*», может иметь слот с атрибутом *имя* и слот с атрибутом *адрес*. В этих слотах должны быть записаны действительные имя и адрес служащего.

Стандарт Plug and Play (*вставляй и работай*) – стандарт для внешних устройств, который не требует настройки их адаптеров. Операционная система сама определяет подключенное устройство и автоматически настраивает его. Кроме того, для многих устройств этот стандарт позволяет «горячее» подключение и отключение, т. е. без выключения питания компьютера.

Стример – накопитель на магнитной ленте (кассете), предназначенный в основном для хранения больших объемов информации, например резервного копирования жестких дисков.

Структура компьютера – совокупность функциональных элементов компьютера и связей между ними. Графически структура компьютера представляется в виде схем, с помощью которых можно получить описание компьютера на любом уровне детализации.

Таблица размещения файлов (FAT–File Allocation Table) – таблица, содержащая номера всех кластеров диска, в которой отмечаются кластеры, занятые тем или иным файлом (адрес файла на диске). Файл на диске может быть фрагментированным, т. е. «разбросанным» по поверхности диска. Именно FAT указывает, в каких кластерах диска расположен конкретный файл.

Твип – самая маленькая единица измерения длины в Windows, по умолчанию единица измерения в VBA. Твип составляет $\frac{1}{20}$ пункта (напомним, что пункт равен 0,353 мм или $\frac{1}{72}$ дюйма).

Тип данных – в базах данных параметр, определяющий данные, которые могут быть введены в поле таблицы: число, дата/время, денежный, логический, счетчик, текст и т. д.

Тонкопленочный транзистор – технология изготовления жидкокристаллических мониторов, в которых для каждого пикселя используются отдельные транзисторы.

Топология сети – структура построения вычислительной сети. Топология во многом определяется структурой *сети связи*, т. е. способом соединения объектов сети друг с другом через коммутирующие устройства.

Транзакция – 1. В диалоговых системах это прием порции данных (сообщения или запроса) от пользователя, ее обработка и выдача ответного сообщения. 2. В базах данных и файловых системах это выполнение элементарной целостной операции над данными (например, удаление, модификация или добавление записи), в течение которой база данных или файловая система находятся в некорректном состоянии.

Тэги HTML – операторы языка *HTML*, используемого для создания Web-страниц, определяющие отображение информации на странице, ссылки на другие страницы, позволяющие создавать на странице различные формы документов, пересылаемых для заполнения по указанным адресам и выполняющие другие функции страницы.

Файл – выделенная на диске или другом носителе область для хранения какой-либо информации, имеющая имя, которое задается пользователем при записи информации на диск.

Файловая система – адресная структура хранения папок и файлов на диске, позволяющая быстро по заданному адресу (пути) находить нужную информацию. Файловая система создается при форматировании диска и зависит от установленной операционной системы.

Физический уровень работы с компьютером – управление устройствами компьютера с помощью программ, написанных в машинных кодах. Такие программы всегда строго специализированны и жестко привязаны к аппаратуре, к конкретному оборудованию с учетом специфики его работы.

Фишинг – в Интернете это способ обманым путем получить с компьютеров пользователей личную или финансовую информацию через мошенническое сообщение электронной почты или веб-узел. Обычно фишинг в Интернете начинается с электронного сообщения, похожего на официальное уведомление из надежного источника: банка, компании кредитных карточек или от известного получателя интерактивного платежа. В электронном сообщении содержится ссылка на поддельный веб-узел, где требуется ввести личную информацию, например номер банковского счета или пароль. Эта информация впоследствии используется с целью кражи.

Фрейм (frame – рамка, окно) – на Web-странице выделенный самостоятельный раздел. В экспертных системах – метод представления знаний, который связывает свойства с узлами, представляющими понятия и объекты. Свойства описываются атрибутами (слотами) и их значениями.

Шаг монитора – расстояние между центрами пикселей на экране монитора. Чем меньше шаг монитора, тем выше четкость изображения.

Шина системная – общий канал для пересылки информации между устройствами в компьютере. По линиям системной шины передаются данные (шина данных), адреса ячеек памяти или устройств (шина адреса), по другим линиям шины подводится питание к устройствам, специальные сигналы управления ими и т. д.

Шина локальная – кроме системной шины в современных компьютерах используются различные локальные шины (*USB, SCSI, IEEE 1394* и др.) для подключения к компьютеру самых разнообразных устройств. Локальные шины расширяют возможности системной шины.

Шифрование – преобразование информации методами криптографии по специальным алгоритмам из общепринятой формы в кодированную (шифрованную) с целью ее сохранности и безопасности.

Эвристика – эмпирическое правило, упрощающее или ограничивающее поиск решения в предметной области, которая является сложной или недоступной ясному пониманию.

Экспертная система – компьютерная программа, использующая знания и процедуры вывода из цепочек рассуждений для высокоэффективного решения задач в некоторой узкой предметной области. Такие программы, как правило, представляют знания символически, исследуют и объясняют свои процессы рассуждения и предназначены для тех предметных областей, в которых людям для достижения мастерства необходимы годы специального обучения и практики.

Электронный документ – документ, в котором информация представлена в электронно-цифровой форме.

Элемент управления – объект формы: надпись, поле для ввода, кнопка, переключатель для выбора чего-то, флажок и т. д.

Язык – множество всех слов и других конструкций, записываемых в данном алфавите согласно правилам. Язык – это средство отражения и познания окружающего нас мира. Всякий информационный процесс может осуществляться только при наличии языка, описывающего объекты и связи между ними.

Язык обработки текстов – язык программирования, разработанный специально для представления и манипулирования сложными языковыми концепциями. Это, например, символьные языки программирования *LISP, SMALLTALK* или языки инженерии знаний *LOOPS, OPS-5, ПРОЛОГ* и др.

Язык программирования – искусственный язык, разработанный для написания программ работы компьютера.

Ярлык – небольшого размера файл на диске, содержащий путь доступа к объекту, для которого он создан, представляется в виде маленького рисунка (иконки) и создается по команде из контекстного меню объекта. Ярлык размещается пользователем в удобном для него месте (на Рабочем столе, в папке Мои документы или в любой другой, в стартовом меню и т.д.). На двойной щелчок левой клавишей мыши по ярлыку реагирует объект (запускается программа, открывается рисунок, открывается окно приложения или документа и т. д.). В отличие от значка, при удалении ярлыка объект не удаляется.

API (Application Programming Interface) – интерфейс прикладного программирования. Это основа разработки Windows-программ – более 1000 функций, предоставляемых библиотеками DLL, которые позволяют приложениям открывать и закрывать окна, читать то, что вводится с клавиатуры, интерпретировать движения мыши и т. д.

Application program package – пакет прикладных программ, например Microsoft Office, включающий в себя несколько различных приложений для выполнения той или иной работы.

Array – массив данных.

ASCII – американский стандартный код обмена информацией. Представляет собой набор стандартных числовых кодов для печатаемых, управляющих и специальных символов, используемых в компьютерах. Другими распространенными кодировками наборов символов являются ANSI, Unicode и EBCDIC – расширенный код двоично-десятичного обмена, применяемый фирмой IBM в мейнфреймах.

Authorized access – санкционированный доступ.

Binary code (двоичный код) – набор данных, представленных в виде совокупности нулей и единиц.

Burst mode – пакетный режим.

Character – символ, знак.

Compatible information systems – совместимые информационные системы.

Computer literacy – компьютерная грамотность.

Cookies – файл небольшого размера, создаваемый Web-сервером для пометки браузера пользователя, посетившего его, с целью отслеживания пользователей и их поведения.

Data security – безопасность данных.

Data validity – достоверность данных.

DIB (*Device-Independent Bitmap* – растровое аппаратно-независимое изображение) – формат, в котором совместно с графическим изображением сохраняется информация об использованной цветовой палитре и разрешающей способности исходного устройства, на котором был создан растровый рисунок. В этом формате воспроизведение растрового рисунка не зависит от аппаратуры.

DLL (*Dinamic Link Libraray*) – динамически компокуемая библиотека. Модули ОС Windows, декларирующие набор точек входа в модуль, соответствующие внешним ссылкам. Большая часть Windows представляет собой совокупность таких модулей, которые используются для взаимодействия между различными компонентами. Например, для вызова функции *MessageBox()* в выполняемый файл при компоновке программы будет включена ссылка на точку входа номер 1 модуля *User*, являющегося модулем ядра Windows. Функции компонируются с приложениями во время их работы, а не во время компиляции.

DMA (*Direct Memory Access – прямой доступ к памяти*) – в некоторых устройствах компьютера имеется контроллер, позволяющий устройству обмениваться информацией с оперативной памятью напрямую, минуя центральный процессор, не прерывая его работу.

FAT (*File Allocation Table*) – таблица размещения файлов на диске. Создается при форматировании диска и заполняется по мере записи файлов на него.

FDDI (*Fiber Distributed Data Interface*) – волоконно-оптический интерфейс компьютерных сетей.

FTP (*File Transfer Protocol*) – протокол передачи файлов между сетевыми устройствами в Интернете.

GDI – модуль ядра ОС Windows, ответственный за реализацию графических функций, таких, как рисование линий и управление цветом. Это DLL-библиотека, в которую входят все функции Windows, обеспечивающие работу с графическими элементами.

HTML (*Hyper Text Markup Language*) – язык разметки гипертекстов. HTML является фундаментальной технологией, базовым форматом Интернета. Это полнофункциональный язык программирования, обладающий практически всеми чертами других языков программирования. Его метки (теги) являются аналогами операторов, процедур, выражений и т. д. обычных языков программирования. Предназначен для создания файлов, которые можно форматировать и отображать браузерами WWW.

HTTP (*Hyper Text Transfer Protocol*) – объектно-ориентированный протокол уровня приложения, используемый в Интернете с 1990 г. Является протоколом передачи гипертекста. Принцип работы очень похож на работу с энциклопедией: вы читаете статьи, находите интересующие вас ссылки и по ним читаете другие статьи.

IPL (*Initial Program Load*) – программа начальной загрузки операционной системы при включении питания компьютера. Располагается в ПЗУ (BIOS).

IRQ (*Interrupt Request*) – запрос на прерывание работы центрального процессора с целью выполнения им другого вида работы.

ISDN (*Integrated Services Digital Network*) – интегрированная цифровая сеть, в которой все передаваемые сигналы полностью цифровые, что позволяет одновременно передавать по одной двухпроводной линии данные разных типов (речь, видеоизображения, музыку и т. д.).

LCD (*Liquid Crystal Display*) – жидкокристаллический монитор.

MBS (*Master Boot Sector*) – главный загрузочный сектор на диске, в котором размещена главная загрузочная запись операционной системы.

MPEG (*Motion Pictures Experts Group*) – алгоритм сжатия видео и звуковых данных. Для движущихся изображений используется метод межкадровой упаковки.

NDP (*Numeric Data Processor*) – сопроцессор, в настоящее время встраиваемый в микросхему центрального процессора для ускорения обработки вещественных чисел. Ранее располагался на системной плате в виде отдельной микросхемы.

OSI (*Open System Interconnection*) – эталонная модель взаимодействия открытых информационных сетей, принятая в качестве международного стандарта.

ODBC (*Open Data Basic Connectivity*) – стандарт Microsoft, позволяющий приложениям осуществлять доступ к данным различных баз данных посредством единого интерфейса прикладного программирования API Win32, который содержит более 1000 различных функций и макросов.

PNP (*plug and play – вставляй и работай*) – стандарт для внешних устройств, не требующий настройки плат адаптеров.

Router – маршрутизатор. Маршрутизаторы представляют собой программно-аппаратные комплексы (компьютеры), которые физически соединяют компьютерные сети друг с другом и с помощью специального программного обеспечения передают пакеты информации из одной сети в другую.

USB (*Universal Serial Bus*) – универсальная последовательная шина, позволяющая подключать к персональному компьютеру до 127 различных периферийных устройств.

VBA (*Visual Basic for Application*) – визуальный язык программирования высокого уровня, предназначенный для разработки приложений Windows, встроенный в программные продукты корпорации Microsoft, в частности в офисный пакет Microsoft Office. Каждое приложение обладает собственной «разновидностью» VBA со своей иерархией объектов.

VRAM (*Video RAM*) – специализированный тип двухпортовых ОЗУ, предназначенных для использования в видеосистемах. Усовершенствованный вариант – **WRAM**.

WYSIWYG (*What You See Is What You Get*) – «что видите, то и получаете»: принцип построения программного обеспечения, при котором изображение на экране точно соответствует тому, что будет выведено на печать.

WWW, World Wide Web (Всемирная паутина) – служба прямого доступа, требующая полноценного подключения к Интернету и позволяющая интерактивно взаимодействовать с представленным на Web-серверах содержанием: текстом, графическим изображением, видеоизображением, аудиозвуком и т. д. Взаимодействие осуществляется по принципу «клиент – сервер» с использованием протокола передачи гипертекста HTTP.

Web-page (Web-страница) – составная часть Web-сайта. Физически представляет собой HTML-файл. Может содержать тексты, изображения, таблицы, сценарии на языках VBA и JAVA и другие элементы. Страница может генерироваться статическая или динамическая. В случае использования фреймов, каждый фрейм считается отдельной страницей.

Web-Site – Web-сайт – совокупность Web-страниц, объединенных по смыслу и физически находящихся на одном сервере.

XML (*Extended Markup Language*) – усовершенствованный язык, предназначенный для описания структуры Web-документа.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

Основной:

1. *А. В. Могилев и др.* Информатика: Учеб. пособие. – М.: Academia, 2006.
2. *Коноплева И. А.* Информационные технологии: Учеб. пособие /И. А. Коноплева, О. А. Хохлова, А. В. Денисов/. Под ред. И. А. Коноплевой. М.: ТК Вебли: Проспект, 2007. – 304 с.
3. *Угринович Н. Д.* Информатика и информационные технологии. – М.: БИНОМ: Лаборатория знаний, 2003.
4. *Гладких Б. А.* Информатика: Введение в специальность: Учеб. пособие для вузов. – Томск: Изд-во науч.-техн. лит., 2002. – 350 с.
5. *Острейковский В. А.* Информатика: Учеб. для вузов.– М.: Высш. шк., 2001.
6. *Информатика: Учеб. по базовому курсу* /Семакин И., Залогова Л. и др./ – М.: Лаборатория Базовых Знаний, 2001.
7. *Информатика. Базовый курс: Учеб. для вузов.* Под ред. С. В. Симоновича. – СПб.: Питер, 2000.
8. *Леонтьев В. П.* Новейшая энциклопедия персонального компьютера 2002. – М.: ОЛМА-ПРЕСС, 2002.
9. *Шаньгин В. Ф.* Информационная безопасность компьютерных систем и сетей: Учеб. пособие. Инфра – М, 2008. – 416 с.
10. *Об информации, информатизации и защите информации: Федеральный Закон от 20 февраля 1995 г. № 24-ФЗ.*
11. *Указ Президента Российской Федерации «Об основах государственной политики в сфере информатизации» от 20 января 1994 г. № 170.*
12. *Указ Президента Российской Федерации «О совершенствовании информационно-телекоммуникационного обеспечения органов государственной власти и порядке их взаимодействия при реализации государственной политики в сфере информатизации» от 1 июля 1994 г. № 1390.*

Дополнительный:

1. *Информатика в понятиях и терминах: книга для студентов технических вузов* под ред. В. А. Извозчикова. – М.: Просвещение, 2002.
2. *Мальшев В. Н.* Информатика. Вводный курс: Учеб. пособие. – Екатеринбург: УрГУПС, 2001.
3. *Мальшев В. Н.* Информатика: Курс лекций. – Екатеринбург: УрГУПС, 2004.
4. *Журнал «Connect!Мир связи».* – М.: Connect.
5. *Газета «Компьютерная неделя РС WEEK/RE».* – М.: СК Пресс.
6. *Журнал «Информатика».*

7. Сафонов В. О. Экспертные системы – интеллектуальные помощники специалистов. – СПб.: Санкт-Петербургская организация общества «Знание», 1992.
8. Крысин А. В. Информационная безопасность: Практич. руководство. – М.: СПАРК, К.: ВЕК+, 2003.
9. Успенский И. В. Энциклопедия Интернет-бизнеса. – СПб.: Питер, 2001.
10. Малышев В. Н. Текстовый редактор MS WORD: Учеб.- метод. пособие. – Изд. 2-е, перераб. и доп. – Екатеринбург: УрГУПС, 2008.
11. Малышев В. Н. Табличный процессор Microsoft Excel: Учеб.- метод. издание. Изд. 2-е исправл. и доп. – Екатеринбург: УрГУПС, 2006.
12. Информатика: Учеб.: В 3 т. / В. А. Минаев, А. П. Фисун, С. В. Скрыль и др. – М.: Маросейка, 2008.
Том 1. Информатика: Концептуальные основы. – 464 с.
Том 2. Информатика: Средства и системы обработки данных. – 544 с.
Том 3. Информатика: Правовое обеспечение информационной безопасности. – 363 с.

Учебное пособие

Виктор Николаевич Малышев

ИНФОРМАТИКА

курс лекций

для студентов 1 курса технических специальностей
всех форм обучения

Издание 2-е, переработанное и дополненное

Компьютерный оригинал-макет подготовлен автором

Редакторы:

С. В. Пилюгина,

И. М. Леушина,

В. П. Вовчек

Подписано в печать 16. 04.09. Формат 60x84 1/16

Бумага офсетная. Усл. печ. л. 15,2

Тираж 600 экз. Заказ №

Издательство УрГУПС

620034, г. Екатеринбург, Колмогорова, 66