

ГЛАВА 6. СЕТЕВЫЕ ОПЕРАЦИОННЫЕ СИСТЕМЫ

6.1. Классификация операционных систем

Сетевые операционные системы (ОС) созданы для клиент-серверных вычислений. В общих чертах это означает подсоединение однопользовательской рабочей станции общего назначения (клиента) к многопользовательским серверам, и распределение нагрузки между ними. Сетевая операционная система необходима для управления потоками сообщений между рабочими станциями и серверами. Она может позволить любой рабочей станции работать с разделяемым сетевым диском или принтером, которые физически не подключены к этой станции. По запросу клиента сервер предоставляет ему различные сервисные функции. Кроме этого, сетевые ОС обеспечивают совместное использование в сети файлов и принтеров — эти возможности встроены в саму ОС. В результате подобная интегрированная сетевая поддержка позволяет компьютеру, например с сетевой операционной системой Windows NT одновременно взаимодействовать со следующими сетевыми средами:

- с сетями Microsoft, включая Windows NT, Windows 95, Microsoft Windows for Workgroups и Microsoft LAN Manager;
- с сетями на базе Transmission Control Protocol/Internet Protocol (TCP/IP), включая UNIX-хосты;
- с системами удаленного доступа; сетями на основе AppleTalk (при использовании Windows NT Server Services for the Macintosh);
- с сетями Novell Netware 3.x и 4.x.

Подобные сетевые возможности отличают Windows NT от других ОС, таких, как Microsoft MS-DOS и Microsoft Windows, в которых сетевые возможности устанавливаются отдельно от самой ОС.

Операционные системы могут различаться особенностями реализации внутренних алгоритмов управления основными ресурсами компьютера (процессорами, памятью, устройствами), особенностями использованных методов проектирования, типами аппаратных платформ, областями использования и многими другими свойствами. По этим признакам проведена классификация ОС, показанная на рис. 6.1.

Алгоритмы управления ресурсами определяют эффективность сетевой операционной системы. Среди них важнейшими являются следующие алгоритмы:

Поддержка **многозадачности** определяется по числу одновременно выполняемых задач: однозадачные (MS-DOS) и многозадачные (OS/2, UNIX, Windows 95, 2000, XP и др.). Однозадачные ОС включают средства управления периферийными устройствами, средства управления файлами, средства общения с пользователем. Многозадачные ОС, кроме перечисленных функций однозадачных ОС, управляют разделением совместно используемых ресурсов, таких как память, таких память, оперативная память, файлы и внешние устройства и др.

Поддержка **многопользовательского режима** определяется по числу одновременно работающих пользователей и подразделяется на: однопользовательские (MS-DOS, Windows 3.x и др.) и многопользовательские (UNIX, Windows NT). Многопользовательские системы, в отличие от однопользовательских, обладают более развитой системой защиты информации каждого пользователя от несанкционированного доступа других пользователей, а также совместного доступа к разделяемым между ними ресурсам.

Поддержка **вытесняющей и невытесняющей многозадачности** определяется по способу распределения процессорного времени между несколькими одновременно существующими в системе процессами. Основное различие между вытесняющей и невытесняющей

многозадачностью является степень централизации механизма планирования процессов. У невытесняющей многозадачности механизм планирования процессов сосредоточен в операционной системе, а у вытесняющей распределен между ОС и прикладными программами. При невытесняющей многозадачности активный процесс выполняется до тех пор, пока он сам не отдаст управление операционной системе для того, чтобы она сама выбрала из очереди другой готовый к выполнению процесс. При вытесняющей многозадачности решение о переключении процесса с одного процесса на другой принимается операционной системой, а не самим активным процессом.

Поддержка *многопроцессорной обработки* определяется числом процессоров, задействованных на обработку активных процессов. При многопроцессорной обработке все алгоритмы управления усложняются на порядок, данный режим обработки также называют *мультипроцессированием*. Многопроцессорные ОС могут классифицироваться по способу организации вычислительного процесса на асимметричные и симметричные. Асимметричные ОС выполняются целиком только на одном из процессоров системы, распределяя прикладные задачи по остальным процессорам. Симметричная ОС полностью децентрализована и использует весь пул процессоров, разделяя их между системными и прикладными задачами.

Особенности построения *аппаратных платформ* базируются на свойствах операционной системы, ориентированных на аппаратные средства, на которых она реализуется. По типу аппаратуры различают ОС персональных компьютеров, мейнфреймов, кластеров и сетей ЭВМ. Среди перечисленных типов компьютеров могут встречаться как однопроцессорные варианты, так и многопроцессорные. Для больших компьютеров, например многопроцессорных серверов, функции планирования потока выполнения задач реализуются путем использования сложных приоритетных заданий и

требуют большей вычислительной мощности, чем в ОС персональных компьютеров, в связи с чем ОС больших машин являются более сложными и функциональными. Сетевые ОС имеют в своем составе средства передачи сообщений между компьютерами по линиям связи. На основе этих сообщений сетевая ОС поддерживает разделение ресурсов компьютера между удаленными пользователями, подключенными к сети. Для реализации этих функций сетевые ОС поддерживают специальные программные компоненты, реализующие коммуникационные протоколы, рассмотренные в предыдущей главе.

Другие требования предъявляются к операционным системам кластеров. **Кластер** – слабо связанная совокупность нескольких вычислительных систем, работающих совместно для выполнения общих приложений, и предоставляющих пользователю единой системой. Наряду со специальной аппаратурой для функционирования кластерных систем необходима программная поддержка со стороны ОС, которая сводится к синхронизации доступа к разделяемым ресурсам, обнаружению отказов и динамической конфигурации системы. Кроме того, существуют ОС, специально разработанные таким образом, чтобы при необходимости их можно было перенести с одного компьютера на другой. Такие ОС называют **мобильными**.

Особенности областей использования для многозадачных ОС подразделяются на три типа в соответствии с использованными при их разработке критериями эффективности:

- системы пакетной обработки (например, ОС EC)
- системы разделения времени (UNIX, VMS)
- системы реального времени (QNX, RT/11)

Системы **пакетной обработки** предназначались для решения задач в основном вычислительного характера, не требующих быстрого получения результатов. Главной целью и критерием эффективности систем пакетной

обработки является максимальная пропускная способность, то есть решение максимального числа задач в единицу времени.

Для достижения этой цели в системах пакетной обработки используется следующая схема функционирования: в начале работы формируется пакет задания, каждое задание содержит требование к системным ресурсам; из этого пакета заданий формируется мультипрограммная смесь, то есть множество одновременно выполняемых задач. Для одновременного выполнения выбираются задачи, предъявляющие отличающиеся требования к ресурсам, так, чтобы обеспечивалась сбалансированная загрузка всех устройств вычислительной машины; так, например, в мультипрограммной смеси желательно одновременное присутствие вычислительных задач с интенсивным вводом-выводом. Выбор нового задания из пакета заданий зависит от внутренней ситуации, складывающейся в системе, то есть выбирается «выгодное» задание. Следовательно, в таких ОС не возможно гарантировать того или иного задания в течение определенного периода времени. В системах пакетной обработки переключение процессора с выполнения одной задачи на выполнение другой происходит только в случае, если активная задача сама отказывается от процессора, например, из-за необходимости выполнить операцию ввода-вывода. Поэтому одна задача может надолго занять процессор, что делает невозможным выполнение интерактивных задач.

Таким образом, взаимодействие пользователя с вычислительной машиной, на которой установлена система пакетной обработки, сводится к тому, что он приносит задания, отдает его диспетчеру-оператору, а в конце дня после выполнения всего пакета задания получает результат. Очевидно, что такой порядок снижает эффективность работы пользователя.

Системы *разделения времени* призваны исправить основной недостаток систем пакетной обработки – изоляцию пользователя-программиста от процесса выполнения его задач. Каждому пользователю

системы разделения времени предоставляетя терминал, с которого он может вести диалог со своей программой. Так как в системах разделения времени каждой задаче выделяется только квант процессорного времени, ни одна задача не занимает процессор надолго, и время ответа оказывается приемлемым. Если квант выбран достаточно небольшим, то у всех пользователей, одновременно работающих на одной и той же машине, складывается впечатление, что каждый из них единолично использует машину. Ясно, что системы разделения времени обладают меньшей пропускной способностью, чем системы пакетной обработки, так как на выполнение принимается каждая запущенная пользователем задача, а не та, которая «выгодна» системе, и, кроме того, имеются накладные расходы вычислительной мощности на более частое переключение процессора с задачи на задачу. Критерием рациональности построения систем разделения времени является не максимальная пропускная способность, а удобство и эффективность работы пользователя.

Системы *реального времени* применяются для управления различными техническими объектами, такими, например, как станок, спутник, научно-экспериментальная установка или технологическими процессами, такими, как гальваническая линия, доменный процесс и т.п. Во всех этих случаях существует предельно допустимое время, в течение которого должна быть выполнена та или иная программа, управляющая объектом, в противном случае может произойти авария: спутник выйдет из зоны видимости, экспериментальные данные, поступающие с датчиков, будут потеряны, толщина гальванического покрытия не будет соответствовать норме. Таким образом, критерием эффективности для систем реального времени является их способность выдерживать заранее заданные интервалы времени между запуском программы и получением результата - управляющего воздействия. Это время называется *временем реакции системы*, а соответствующее свойство системы – *реактивностью*. Для этих систем мультипрограммная

смесь представляет собой фиксированный набор заранее разработанных программ, а выбор программы на выполнение осуществляется исходя из текущего состояния объекта или в соответствии с расписанием плановых работ.

Некоторые операционные системы могут совмещать в себе свойства систем разных типов, например, часть задач может выполняться в режиме пакетной обработки, а часть – в режиме реального времени или в режиме разделения времени. В таких случаях режим пакетной обработки часто называют **фоновым** режимом.

Особенности методов построения при описании операционной системы часто указываются характерные черты ее структурной организации и основные концепции, положенные в её основу.

Рассмотрим три базовые концепции::

Во-первых, способы построения ядра системы может монолитное ядро или микроядерный подход. Большинство ОС используют монолитное ядро, которое компонуется как одна программа, работающая в привилегированном режиме и использующая быстрые переходы с одной процедуры на другую, не требующие переключения из привилегированного режима в пользовательский и наоборот. Альтернативой является построение ОС на базе микроядра, работающего также в привилегированном режиме и выполняющего только минимум функций по управлению аппаратурой, в то время как функции ОС более высокого уровня выполняют специализированные компоненты ОС – сервера, работающие в пользовательском режиме. При таком построении ОС работает более медленно, так как часто выполняются переходы между привилегированным режимом и пользовательским, зато система получается более гибкой – ее функции можно наращивать, модифицировать или сужать, добавляя, модифицируя или исключая серверы пользовательского режима. Кроме того,

серверы хорошо защищены друг от друга, как и любые пользовательские процессы.

Во-вторых, построение ОС на базе *объектно-ориентированного* подхода дает возможность использовать все его достоинства, хорошо зарекомендовавшие себя на уровне приложений, внутри операционной системы, а именно: аккумуляцию удачных решений в форме стандартных объектов, возможность создания новых объектов на базе имеющихся с помощью механизма наследования, хорошую защиту данных за счет внедрения во внутренние структуры объекта, что делает данные недоступными для несанкционированного использования извне, структурированность системы, состоящей из набора хорошо определенных объектов.

В-третьих, наличие нескольких *прикладных сред* дает возможность в рамках одной ОС одновременно выполнять приложения, разработанные для нескольких ОС. Многие современные операционные системы поддерживают одновременно прикладные среды MS-DOS, Windows, UNIX (POSIX), OS/2 или хотя бы некоторого подмножества из этого популярного набора. Концепция множественных прикладных сред наиболее просто реализуется в ОС на базе микроядра, над которым работают различные серверы, часть которых реализуют прикладную среду той или иной операционной системы.

Распределенная организация операционной системы позволяет упростить работу пользователя и программистов в сетевых средах. В распределенной ОС реализованы механизмы, которые дают возможность пользователю представлять и воспринимать сеть в виде традиционного однопроцессорного компьютера. Характерными признаками распределенной организации ОС являются: наличие единой справочной службы разделяемых ресурсов, единой службы времени, использование механизма вызова удаленных процедур RPC (Remote Procedure Call) для прозрачного распределения программных процедур по машинам, многонитевой

обработки, позволяющей распараллеливать вычисления в рамках одной задачи и выполнять эту задач сразу на нескольких компьютерах сети, а также наличие других распределенных служб.

6.2. Обобщенная структура операционных систем

Системы должны быть гибкими с точки зрения бизнес-компонентов, открытыми на уровне технологий объектного взаимодействия, и, что очень важно для будущего — обладать высокой степенью стандартизованности выбранных базовых технологий. Чем больше производителей вычислительных систем поддерживают стандарт, тем ниже вероятность больших расходов при интеграции как программных, так и аппаратных комплексов.

Сетевая операционная система составляет основу любой вычислительной сети. Под сетевой операционной системой в широком смысле понимается совокупность операционных систем отдельных компьютеров, взаимодействующих с целью обмена сообщениями и разделения ресурсов по единым правилам - протоколам. В узком смысле сетевая ОС - это операционная система отдельного компьютера, обеспечивающая ему возможность работать в сети.

В сетевой операционной системе отдельной машины можно выделить несколько частей (рис. 6.2).

Средства *управления локальными ресурсами* компьютера выполняют функции распределения оперативной памяти между процессами, планирования и диспетчеризации процессов, управления процессорами в мультипроцессорных машинах, управления периферийными устройствами и другие функции управления ресурсами локальных ОС.

Средства *предоставления собственных ресурсов* и услуг в общее пользование - серверная часть ОС (сервер). Эти средства обеспечивают, например, блокировку файлов и записей, что необходимо для их совместного

использования; ведение справочников имен сетевых ресурсов; обработку запросов удаленного доступа к собственной файловой системе и базе данных; управление очередями запросов удаленных пользователей к своим периферийным устройствам.

Средства запроса доступа к удаленным ресурсам и услугам и их использования - клиентская часть ОС (редиректор). Эта часть выполняет распознавание и перенаправление в сеть запросов к удаленным ресурсам от приложений и пользователей, при этом запрос поступает от приложения в локальной форме, а передается в сеть в другой форме, соответствующей требованиям сервера. Клиентская часть также осуществляет прием ответов от серверов и преобразование их в локальный формат, так что для приложения выполнение локальных и удаленных запросов неразличимо.

Коммуникационные средства ОС, с помощью которых происходит обмен сообщениями в сети. Эта часть обеспечивает адресацию и буферизацию сообщений, выбор маршрута передачи сообщения по сети, надежность передачи и т.п., то есть является средством транспортировки сообщений.

В зависимости от функций, возлагаемых на конкретный компьютер, в его операционной системе может отсутствовать либо клиентская, либо серверная части.

На рисунке 6.3 показано взаимодействие сетевых компонентов. Здесь ЭВМ 1 выполняет роль клиента, а ЭВМ 2 - роль сервера, соответственно на первой машине отсутствует серверная часть, а на второй - клиентская. На рисунке отдельно показан компонент клиентской части - редиректор. Именно редиректор перехватывает все запросы, поступающие от приложений, и анализирует их. Если выдан запрос к ресурсу данного компьютера, например **HDD** (Hard Disk Drive), то он переадресовывается соответствующей подсистеме локальной ОС, если же это запрос к удаленному ресурсу, он переправляется в сеть. При этом клиентская часть преобразует запрос из

локальной формы в сетевой формат и передает его транспортной подсистеме, которая отвечает за доставку сообщений указанному серверу. Серверная часть операционной системы ЭВМ 2 принимает запрос, преобразует его и передает для выполнения своей локальной ОС. После того, как результат получен, сервер обращается к транспортной подсистеме и направляет ответ клиенту, выдавшему запрос. Клиентская часть преобразует результат в соответствующий формат и адресует его тому приложению, которое выдало запрос.

Существует два подхода к построению сетевых ОС (рис. 6.4).

Первые сетевые ОС представляли собой совокупность существующей локальной ОС и надстроенной над ней *сетевой оболочки*. При этом в локальную ОС встраивался минимум сетевых функций, необходимых для работы сетевой оболочки, которая выполняла основные сетевые функции. Примером такого подхода является использование на каждой машине сети операционной системы MS DOS (у которой начиная с ее третьей версии появились такие встроенные функции, как блокировка файлов и записей, необходимые для совместного доступа к файлам). Принцип построения сетевых ОС в виде сетевой оболочки над локальной ОС используется и в современных ОС, таких, например, как LANtastic или Personal Ware.

Однако более эффективным представляется путь разработки операционных систем, изначально предназначенных для работы в сети. Сетевые функции у ОС такого типа глубоко встроены в основные модули системы, что обеспечивает их логическую стройность, простоту эксплуатации и модификации, а также высокую производительность. Примером такой ОС является система Windows NT фирмы Microsoft, которая за счет встроенностя сетевых средств обеспечивает более высокие показатели производительности и защищенности информации по сравнению с сетевой ОС LAN Manager той же фирмы (совместная разработка с IBM), являющейся надстройкой над локальной операционной системой OS/2.

Компоненты сетевой операционной системы на каждой рабочей станции и файловом сервере взаимодействуют друг с другом посредством языка, называемым протоколом. Одним из общих протоколов является протокол фирмы IBM NetBIOS (Network Basic Input Output System -Сетевая операционная система ввода-вывода). Другим распространенным протоколом является IPX (Internet-work Packet Exchange - Межсетевой обмен пакетами) фирмы Novell.x, рассмотренные в предыдущей главе. Ниже представлены ОС различных разработчиков:

Таблица. Типы ОС

Nпп	Операционная система	Производитель
1	Apple Talk	Apple
2	LANtastic	Artisoft
3	NetWare	Novell
4	NetWare Lite	Novell
5	Personal NetWare	Novell
6	NFS	Sun Microsystems
7	OS/2 LAN Manager	Microsoft
8	OS/2 LAN Server	IBM
9	Windows NT Advanced Server	Microsoft
10	POWERfusion	Performance Technology
11	POWERLan	Performance Technology
12	Vines	Ba

6.3. Модель клиент-сервер и модель ОС на базе микроядра

6.3.1. Модель клиент-сервер

Модель клиент-сервер - это еще один подход к структурированию ОС. В широком смысле модель клиент-сервер предполагает наличие программного компонента - потребителя какого-либо сервиса - клиента, и

программного компонента - поставщика этого сервиса - сервера. Взаимодействие между клиентом и сервером стандартизуется, так что сервер может обслуживать клиентов, реализованных различными способами и, может быть, разными производителями. При этом главным требованием является то, чтобы они запрашивали услуги сервера понятным ему способом. Инициатором обмена обычно является клиент, который посыпает запрос на обслуживание серверу, находящемуся в состоянии ожидания запроса (рис. 6.5). Один и тот же программный компонент может быть клиентом по отношению к одному виду услуг, и сервером для другого вида услуг. Модель клиент-сервер является скорее удобным средством ясного представления функций того или иного программного элемента в той или иной ситуации, нежели технологией. Эта модель успешно применяется не только при построении ОС, но и на всех уровнях программного обеспечения, и имеет в некоторых случаях более узкий, специфический смысл, сохраняя, естественно, при этом все свои общие черты.

В целях обеспечения эффективности и целостности работы, ОС реализуется двумя режимами работы: **режим пользователя** (user mode) и **режим ядра** (kernel mode).

Применительно к структурированию ОС идея состоит в разбиении ее на несколько процессов - подсистем, каждая из которых выполняет отдельный набор сервисных функций - например, управление памятью, создание или планирование процессов. Каждая подсистема выполняется в пользовательском режиме. Клиент, которым может быть либо другой компонент ОС, либо прикладная программа, запрашивает сервис, посыпая сообщение на сервер. Ядро ОС (называемое здесь микроядром), работая в привилегированном режиме, доставляет сообщение нужному серверу, сервер выполняет операцию, после чего ядро возвращает результаты клиенту с помощью другого сообщения (рис. 6.5).

6.3.2. Режим пользователя

Режим пользователя — менее привилегированный по сравнению с режимом ядра режим работы процессора. Он не имеет прямого доступа к аппаратуре. Выполняющийся в этом режиме код непосредственно имеет дело лишь с объектами своего адресного пространства (рис. 6.6).

Системные службы он вызывает через интерфейсы прикладных программ (Application Program Interface, API). Поддерживающие их приложения и подсистемы работают в режиме пользователя. При запуске приложения создается *процесс* (*process*), реализованный в виде *объекта* (*object*). Объект состоит из исполняемой программы, пространства адресов виртуальной памяти и одного или нескольких потоков.

Особенности процесса пользовательского режима таковы:

не имеет прямого доступа к оборудованию. Это сделано в целях защиты от неверно работающих приложений или от несанкционированного доступа. Запросы на использование аппаратных ресурсов должны быть разрешены компонентом режима ядра;

ограничен размерами выделенного адресного пространства. Ограничение размера памяти, используемой процессом, позволяет обеспечить дополнительную защиту ОС. Это ограничение устанавливается путем выделения процессу диапазона фиксированных адресов;

может быть выгружен из физической памяти в виртуальную память на жестком диске. *Виртуальная память* (virtual memory, VRAM) использует пространство жесткого диска как дополнительную оперативную память. В результате процесс режима пользователя получает доступ к памяти, размер которой превышает объем ОЗУ;

приоритет процесса данного типа ниже, чем у процессов режима ядра. Поэтому в сравнении с последними ему, как правило, предоставляется меньше процессорного времени. Это предохраняет ОС от снижения

производительности или возникновения задержек, связанных с ожиданием завершения работы приложений.

Подход с использованием ядра заменил вертикальное распределение функций операционной системы на горизонтальное. Компоненты, лежащие выше микроядра, хотя и используют сообщения, пересылаемые через микроядро, взаимодействуют друг с другом непосредственно. Микроядро играет роль регулировщика. Оно проверяет сообщения, пересылает их между серверами и клиентами, и предоставляет доступ к аппаратуре.

6.3.3. Режим ядра

Режим ядра — это привилегированный режим работы, в котором код имеет прямой доступ ко всем аппаратным ресурсам и всей памяти, включая адресные пространства всех процессов режима пользователя (рис. 6.7).

Ниже перечислены функциональные возможности компонентов режима ядра, которые имеют:

- прямой доступ к оборудованию;
- прямой доступ ко всем видам памяти компьютера;
- более высокий приоритет исполнения, чем процессы режима пользователя.

Кроме того, компоненты не выгружаются на жесткий диск в файл подкачки виртуальной памяти; Функционирование режима ядра обеспечивается ***исполнительной системой***, включающей в себя системные службы, микроядро и слой абстрагирования от оборудования (HAL).

Исполнительная система представляет собой обобщенное наименование ряда подсистем и компонентов ОС, работающих в режиме ядра.

Поскольку системные (исполнительные) службы обеспечивают работу всех основных функций ОС, очень важно защитить их от влияния приложений и подсистем пользовательского режима. Такую защиту обеспечивают системные службы, работающие в режиме ядра:

диспетчеры - различные модули, осуществляющие управление вводом/выводом, объектами, безопасностью, процессами, взаимодействием между процессами, виртуальной памятью, окнами и графикой;

драйверы устройств - программные компоненты, управляющие доступом к оборудованию;

Микроядро предоставляет наиболее общие службы ОС, такие, как диспетчеризация потоков, обработка прерываний первого уровня и отложенный вызов процедур. Микроядро расположено между слоем системных служб и HAL.

Слой абстрагирования от оборудования (HAL) представляет собой библиотеку режима ядра, включающую процедуры управления оборудованием. Этот программный слой позволяет скрыть особенности аппаратных платформ, предоставив ОС стандартные точки входа в процедуры, благодаря чему для нее исчезают различия между платформами и архитектурами. Поэтому ОС может функционировать на разных платформах с разными процессорами. Сетевая операционная система способна работать на одно- и многопроцессорных компьютерах и позволяет высокоуровневым драйверам графических адаптеров форматировать данные для мониторов разных типов.

Сетевые ОС обеспечивает работу с приложениями с помощью **подсистем среды**. Подсистема среды предоставляет API приложениям, разработанным под конкретную среду или ОС. Рассмотрим функционирование на примере широко распространенной подсистемы Win32.

Подсистемы среды являются промежуточным звеном между приложением, спроектированным для работы в конкретной операционной среде, и службами исполнительной системы. Подсистема среды транслирует инструкции, специфичные для рабочей среды приложения, в команды, которые могут быть выполнены службами исполнительной системы. Работа

приложений, созданных для других ОС, поддерживается двумя подсистемами среды Windows NT: POSIX и OS/2. Эти подсистемы обрабатывают все функциональные запросы от поддерживаемых приложений. Подсистема либо самостоятельно обрабатывает запрос, либо передает его службам исполнительной системы СОС.

На **подсистему Win32** иногда ссылаются как на подсистему клиент/сервер, подсистему CSR (Client/SeRver) или CSRSS (Client/SeRver SubSystem). Подсистема поддерживает Win32- (приложения, позволяющие организовать многопоточность выполнения задач, а также метод передачи и совместного использования информации между приложениями), MS-DOS-, Windows 3.x (приложения, содержащие инструкции (команд) платформы Intel x86)- приложения и остальные подсистемы среды. Подсистема Win32 также поддерживает консольные приложения, завершение работы приложений и функции обработки ошибок.

6.3.4. Взаимодействие подсистем с исполнительной системой

Службы исполнительной системы СОС обеспечивают выполнение основных функций ОС для всех подсистем. Службы работают в режиме ядра, что позволяет гарантировать устойчивость ОС, поскольку прямого доступа к ним не имеет ни одно приложение. В результате неправильно работающий компонент режима пользователя (приложение) не сможет случайно остановить функционирование компонента режима ядра.

Подсистемы, построенные на базе исполнительной системы, формируют операционные среды, соответствующие требованиям конкретных клиентских приложений. Благодаря этому общие функции ОС единственный раз реализуются в исполнительной системе, не повторяясь в каждой из подсистем. Это упрощает разработку новых подсистем и облегчает их поддержку.

Эта теоретическая модель является идеализированным описанием системы клиент-сервер, в которой ядро состоит только из средств передачи сообщений. В действительности различные варианты реализации модели клиент-сервер в структуре ОС могут существенно различаться по объему работ, выполняемых в режиме ядра.

На одном краю этого спектра находится разрабатываемая фирмой IBM на основе микроядра Mach операционная система Workplace OS, придерживающаяся чистой микроядерной доктрины, состоящей в том, что все несущественные функции ОС должны выполняться не в режиме ядра, а в непrivилегированном (пользовательском) режиме. На другом - Windows NT, в составе которой имеется исполнительная система (NT executive), работающая в режиме ядра и выполняющая функции обеспечения безопасности, ввода-вывода и другие.

Микроядро реализует функции, лежащие в основе операционной системы. Это основа для менее существенных системных служб и приложений. В общем случае, подсистемы, бывшие традиционно неотъемлемыми частями операционной системы - файловые системы, управление окнами и обеспечение безопасности - становятся периферийными модулями, взаимодействующими с ядром и друг с другом.

Главный принцип разделения работы между микроядром и окружающими его модулями - включать в микроядро только те функции, которым абсолютно необходимо исполняться в режиме супервизора и в привилегированном пространстве. Под этим обычно подразумеваются машинозависимые программы (включая поддержку нескольких процессоров), некоторые функции управления процессами, обработка прерываний, поддержка пересылки сообщений, некоторые функции управления устройствами ввода-вывода, связанные с загрузкой команд в регистры устройств.

Есть два пути построения подсистем. Один путь - разместить

несколько таких, чувствительных к режиму работы процессора, серверов, в пространстве ядра, что обеспечит им полный доступ к аппаратуре и, в то же время, связь с другими процессами с помощью обычного механизма сообщений. Такой подход был использован, например, при разработке Windows NT: кроме микроядра, в привилегированном режиме работает часть Windows NT, называемая executive управляющей программой. Она включает ряд компонентов, которые управляют виртуальной памятью, объектами, вводом-выводом и файловой системой (включая сетевые драйверы), взаимодействием процессов, и частично системой безопасности.

Другой путь заключается в том, чтобы оставить в ядре только небольшую часть сервера, представляющую собой механизм реализации решения, а часть, отвечающую за принятие решения, переместить в пользовательскую область. В соответствии с этим подходом, например, в микроядре Mach, на базе которого разработана Workplace OS, размещается только часть системы управления процессами (и нитями), реализующая **диспетчеризацию** (то есть непосредственно переключение с процесса на процесс), а все функции, связанные с анализом приоритетов, выбором очередного процесса для активизации, принятием решения о переключении на новый процесс и другие аналогичные функции выполняются вне микроядра. Этот подход требует тесного взаимодействия между внешним **планировщиком** и резидентным диспетчером.

Здесь важно отметить различие - запуск процесса или нити требует доступа к аппаратуре, так что по логике - это функция ядра. Но ядру все равно, какую из нитей запускать, поэтому решения о приоритетах нитей и дисциплине постановки в очередь может принимать работающий вне ядра планировщик.

Перемещение планировщика на пользовательский уровень может понадобиться и для чисто коммерческих целей. Некоторые производители ОС (например, IBM и OSF со своими вариантами микроядра Mach)

планируют лицензировать свое микроядро другим поставщикам, которым может потребоваться заменить исходный планировщик на другой, поддерживающий, например, планирование в задачах реального времени или реализующий какой-то специальный алгоритм планирования.

6.4. Топологии распределенных вычислений

С ростом числа предприятий, применяющих в своих сетях технологию клиентсервер, стандартизированная распределенная обработка становится ключевым фактором эффективности этого подхода. Компьютер под управлением сетевой ОС может разделять *приложения* на две группы: *интерфейсную* (front-end), работающую на клиенте, и *прикладную* (back-end), выполняющуюся на сервере. Такое распределение позволяет приложению лучше использовать преимущества имеющихся аппаратных ресурсов, таких, как несколько процессоров или большие объемы оперативной памяти. Для создания клиент-серверных соединений, поддерживающих распределенную обработку, применяет механизмы взаимодействия процессов IPC (interprocess communication).

В типичном распределенном приложении вычислительная задача делится на два процесса: интерфейсный, работающий на рабочей станции и требующий минимума ресурсов, и прикладной — выполняемый на сервере и нуждающийся в больших объемах данных, интенсивных вычислениях, общих правилах обработки или специализированной аппаратуре (рис 6.8). Сервер предоставляет свои вычислительные мощности в распоряжение задач клиентов.

При распределенной обработке между клиентской и серверной частью приложения должно существовать сетевое соединение, обеспечивающее двусторонний обмен данными. Опишем основные механизмы функционирования IPC, используемые для создания таких соединений - именованные каналы и почтовые ящики

Именованный канал - построенный двунаправленный канал связи между клиентом и сервером. Именованные каналы предоставляют распределенным приложениям средства обмена сообщениями с гарантированной доставкой. После открытия канала клиент и сервер могут читать из канала и записывать в него. Пример процесса, использующего именованные каналы, — WinLogon.

Почтовые ящики - построенные односторонние каналы связи между клиентом и сервером. Почтовые ящики обеспечивают средства обмена сообщениями без гарантии доставки. Они могут использоваться для идентификации в сети компьютеров и служб.

Windows Sockets (WinSock) - предоставление распределенным приложениям доступа к транспортным протоколам, таким, как TCP/IP и IPX. WinSock можно использовать для построения между клиентом и сервером каналов связи с гарантированной доставкой.

Удаленный вызов процедур RPC - предоставление распределенным приложениям возможности вызывать процедуры, доступные на разных компьютерах в сети.

Сетевой динамический обмен данными NetDDE (Network Dynamic Data Exchange)- совместное использование информации между приложениями. Для взаимодействия с сетевыми компонентами нижнего уровня NetDDE применяет NetBIOS API.) NetDDE использует, к примеру, программа Chat.

Распределенная модель многокомпонентных объектов DCOM (Distributed Component Object Model) Распределение применяющих RPC процессов по нескольким компьютерам, чтобы клиентская и серверная части приложения могли быть размещены в оптимальных участках сети. DCOM — это ActiveX-технология Microsoft, которую могут использовать приложения JavaTM и компоненты ActiveX при помощи модели многокомпонентных объектов (COM).

6.5. Администрирование сети

Как уже указывалось в предыдущих разделах главы операционная система, управляющая обработкой, управлением и передачей информации подразделяется на два основных типа: пользовательская и сервера. Соответственно назначение, функции и управление таких операционных систем различно.

Рабочая станция под управлением пользовательской операционной системы, как правило, может поддерживать: выполнение нескольких процессов, создавать, хранить и обновлять список конфигурации компьютера, средства доступа в Интернет, службу сообщений, службу локальной безопасности и защиты файлов, папок и других локальных ресурсов компьютера, надежность функционирования приложений в операционной системе (каждое приложение выполняется в отдельном адресном пространстве).

Серверная операционная система, например Windows NT Server, оптимизирована для работы в качестве сервера файлов, печати, а также для приложений с широким спектром применений: от администрирования нескольких рабочих групп до корпоративных сетей. Основными функциями операционной системы сервера являются: поддержка многопроцессорной обработки задач, управление и администрирование сервера и сети, отслеживание входящего и исходящего трафика сервера, поддержку Web-сервера, интеграцию с клиентами других фирм производителей, например Macintosh и др.

6.5.1. Модели администрирования и регистрации в сети

Сети, работающие под управлением Microsoft Windows NT, могут быть организованы на основе доменной модели или модели рабочей группы.

Доменная модель характеризуется наличием в сети минимум одного компьютера, работающего под управлением Windows NT Server и

выполняющего роль контроллера домена (domain controller). Домен – группа компьютеров, объединенных общей базой учетных записей пользователей и единой политикой защиты.

Модель рабочей группы позволяет организовать сеть на основе Windows NT без контроллера домена. Компьютеры при такой организации обладают равными правами на совместно используемые ресурсы. Главным недостатком построения таких сетей является отсутствие централизованного управления и администрирования учетных записей пользователей и защиты ресурсов, которые создаются на каждом компьютере, где пользователь будет регистрироваться.

Чтобы получить доступ к ресурсам, пользователям необходимо прежде всего зарегистрироваться – идентифицировать себя в домене или компьютере, при этом ему необходимо ввести имя пользователя, пароль, а также название домена, в котором зарегистрирована учетная запись или название компьютера. Окно, в котором происходит регистрация пользователя раскрывается при загрузке операционной системы или при нажатии кнопок Ctrl-Alt-Delete и выборе пункта «Завершение работы» – далее «Завершение сеанса...», представлено на рисунке 6.9.

Учетная запись пользователя – информация о пользователе системы, включающая в себя имя пользователя и пароль, необходимые для регистрации, информации о принадлежности к той или иной рабочей группе или домену, права и привилегии.

Учетные записи бывают двух типов: глобальные и локальные. Глобальная учетная запись содержит информацию о пользователе домена. Она позволяет пользователю зарегистрироваться в домене с любого компьютера сети и работать с доступными для него ресурсами. В Windows NT глобальную запись можно создать средствами User Manager for Domain (Диспетчер пользователей доменов). Она размещается в основной базе данных каталогов на главном контроллере домена PDC (Primary domain

controller). Копии базы данных хранятся на всех резервных контроллерах домена BDC (Backup domain controller), которые с интервалом в пять минут обновляются с основного контроллера домена. Пример построения такой сети представлен на рис. 6.10.

Локальная учетная запись содержит информацию о пользователе данного компьютера. С ее помощью пользователь может зарегистрироваться в системе и получить доступ к ресурсам компьютера. Чтобы иметь право обратиться к ресурсам другого компьютера, надо и на нем завести локальную учетную запись пользователя.

6.5.2. Основные правила конфигурирования компьютеров, подключенных к сети

После того, как было установлено физическое соединение сети (установлено и подключено сетевое оборудование) необходимо соответствующим образом сконфигурировать, то есть программно настроить компьютеры, находящиеся в сети. Для этого необходимо произвести настройку сети. Это можно сделать только в том случае, если пользователь обладает соответствующими правами на конфигурирование системы. Такими правами, как правило, обладает пользователь из группы «Администратор». Настроить сетевые установки можно путем нажатия правой кнопки мыши на значке «Мое сетевое окружение», которое, как правило, располагается на Рабочем столе операционной системы и выбрать пункт меню «Свойства». При этом откроется окно «Сеть и удаленный доступ к сети».

Для того, чтобы раскрыть окно «Подключения по локальной сети - свойства» (рис. 6.11), в котором и настраиваются параметры подключения, необходимо правой кнопкой мыши нажать на значке «Подключение по локальной сети».

В этом окне необходимо установить протокол передачи данных, службу доступа к информации по сети, а также указать, клиентом каких сетей Вы являетесь. Для выбора протокола передачи данных по сети

необходимо в открывшемся окне нажать на кнопку «Установить», а затем в новом окно выбрать «Протокол», нажать «Добавить» (рис. 6.12). Раскроется список доступных для установки протоколов. Выберем, например, протокол передачи данных TCP/IP, для функционирования которого необходимо установить в свойствах данного протокола уникальный для каждого компьютера сети IP-адрес (например, 192.168.0.33) и маску подсети (например, 255.255.0.0).

Кроме того, чтобы получить возможность передавать данные по сети, а также иметь доступ к ресурсам другого компьютера необходимо также установить, что пользователь является клиентом сети Microsoft, а также службу доступа к файлам и принтерам сетей Microsoft. Для этого необходимо в окне «Подключения по локальной сети - свойства» выбрать «Установить», затем, в открывшемся окне выбрать «Клиент», а затем из списка выбрать «Клиент для сетей Microsoft». Служба доступа к файлам и принтерам сетей Microsoft устанавливается аналогичным образом, только в окне «Выбор типа сетевого компонента» выбрать «Служба» и далее в открывшемся окне выбрать «Служба доступа к файлам и принтерам сетей Microsoft».

После выполнения вышеописанных действий, дважды щелкнув левой кнопкой мыши на значке «Мое сетевое окружение» Вы должны увидеть список подключенных в данный момент и настроенных компьютеров в сети, у которых хотя бы один локальный ресурс имеет общий доступ.

По умолчанию все ресурсы компьютера – папки, принтеры и др. не имеют общего доступа. Для того, чтобы разрешить общий доступ к ресурсам своего компьютера необходимо сначала выделить данный объект, затем, нажав правой кнопкой мыши на этом объекте, из раскрывшегося контекстного меню выбрать «Доступ». В открывшемся окне установить «Открыть общий доступ к этой папке» и при необходимости в строке «Сетевое имя» ввести имя, под которым другие компьютеры будут видеть данный ресурс.

6.5.3. Общие сведения об администрировании пользователей и рабочих групп

В сетевой операционной системе Windows NT Server присутствует специальный инструмент, предназначенный для администрирования глобальных учетных записей пользователей и групп на основном контроллере домена, а также локальные учетные записи на любом компьютере домена - User Manager for Domains.

Окно утилиты User Manager for domains представлено на рисунке 6.13.

Для того, чтобы создать учетную запись нового пользователя в домене, необходимо в меню User выбрать "New User...". При этом раскроется окно "New User" (рис 6.14).

Здесь необходимо ввести Имя пользователя (Username) под которым он будет регистрироваться в домене, полное имя пользователя (Full Name), описание, которое может отождествлять пользователя (Description), пароль для регистрации в домене (Password) и подтверждение пароля (Confirm Password). Кроме того, в этом окне можно задать смену пароля при первой регистрации пользователя (User Must Change Password at Next Logon), запретить смену пользователем пароля (User Cannot Change Password), ограничение действия пароля (Password Never Expires), отключить учетную запись (Account Disabled).

Существуют также и другие ОС, например, Linux, администрирование которых отличается, в основном, только инструментом управления учетными записями и настройки служб, протоколов и клиентов.

Контрольные вопросы к главе 6

1. Для чего нужны сетевые операционные системы?
2. По каким основным признакам можно классифицировать ОС?
3. Что такое кластер и какие ОС называют мобильными?

4. Дайте определения понятиям: время реакции системы и реактивность:
5. Опишите два основных подхода к построению ОС.
6. Для чего необходима виртуальная память в компьютере:
7. Каким образом обеспечивается взаимодействие подсистем с исполнительной системой?
8. Опишите основные принципы построения подсистем.
9. В чем основное отличие одноранговых и двухранговых классов сетей?
10. Перечислите известные Вам ОС.
11. На какие группы приложений ОС может разделить компьютер на работающей с сетевой операционной системой.
12. В чем заключается основной принцип организации распределенных вычислений?
13. Для чего необходима служба удаленного вызова процедур и сетевой динамический обмен данными?

ГЛАВА 7. СТРУКТУРА И ИНФОРМАЦИОННЫЕ УСЛУГИ ТЕРРИТОРИАЛЬНЫХ СЕТЕЙ

7.1. Структура территориальных сетей

Глобальная сеть Internet - самая крупная и единственная в своем роде сеть в мире. Среди глобальных сетей она занимает уникальное положение. Правильнее ее рассматривать как некоторую надсеть - объединение многих сетей, сохраняющих самостоятельное значение. Действительно, Internet не имеет ни четко выраженного владельца, ни национальной принадлежности. Любая сеть может иметь связь с Internet и, следовательно, рассматриваться как ее часть, если в ней используются принятые для Internet протоколы TCP/IP или имеются конверторы в протоколы TCP/IP. Практически все сети национального и регионального масштабов имеют выход в Internet.

Типичная территориальная (национальная) сеть имеет иерархическую структуру.

Верхний уровень - федеральные узлы, связанные между собой магистральными каналами связи. Магистральные каналы физически организуются на ВОЛС или на спутниковых каналах связи. Средний уровень - региональные узлы, образующие региональные сети. Они связаны с федеральными узлами и, возможно, между собой выделенными высокоскоростными каналами, такими, как каналы T1, E1, B-ISDN или радиорелейные линии. Нижний уровень - местные узлы (серверы доступа), связанные с региональными узлами преимущественно коммутируемыми или выделенными телефонными каналами связи, хотя заметна тенденция к переходу к высокоскоростным каналам. Именно к местным узлам подключаются локальные сети малых и средних предприятий, а также компьютеры отдельных пользователей. Корпоративные сети крупных

предприятий соединяются с региональными узлами выделенными высокими или среднескоростными каналами.

Иерархическая архитектура Internet может быть представлена так, как на рис. 7.1.

Внутри каждой автономной системы (AS) используется некоторый единый внутренний протокол маршрутизации, например IGP. Между AS маршрутизация подчиняется внешним протоколам, например EGP.

7.2. Протоколы файлового обмена, электронной почты, дистанционного управления

Основные услуги телекоммуникационных технологий:

- передача файлов;
- электронная почта;
- телеконференции;
- справочные службы (доски объявлений);
- видеоконференции;
- доступ к информационным ресурсам (информационным базам) сетевых серверов;
- мобильная сотовая связь;
- компьютерная телефония.

Файловый обмен - это доступ к файлам, распределенным по различным компьютерам. В сети Internet на прикладном уровне используется протокол FTP. Доступ возможен в режимах off-line и on-line. В режиме off-line посыпается запрос к FTP-серверу, сервер формирует и посыпает ответ на запрос. В режиме on-line осуществляется интерактивный просмотр каталогов FTP-сервера, выбор и передача нужных файлов. Для осуществления указанных операций на ЭВМ пользователя должно быть установлено программное обеспечение FTP-клиент. При запросе файла по протоколу FTP пользователь должен знать, где находится нужный ему файл. Для этого

удобно воспользоваться другой информационной системой сети Internet, называемой Archie. Обращаясь к клиенту Archie по команде

archie <имя файла>,

пользователь получает в ответ адрес сервера, имя директории и размер файла. Далее можно обращаться к FTP-серверу с помощью команды:

ftp[<параметры>][<имя сервера>].

Квадратные скобки в записи команд означают необязательные части. Параметры используются только при отладке FTP. В качестве имени сервера указывается IP-имя или IP-адрес удаленного компьютера.

В большинстве серверов Internet для входа по FTP-команде нужны предварительная регистрация пользователя и указание пароля. Однако это не требуется при обращениях к общедоступным (анонимным) серверам. Такие серверы создают и обслуживают организации, заинтересованные в распространении информации определенного вида.

После выполнения команды обращения к серверу FTP-клиент переходит в командный режим. Ниже приведены примеры команд, которые могут выполняться в командном режиме (где S - удаленный компьютер, T - локальный компьютер):

open [<имя S>] - устанавливает связь с удаленным компьютером;

close [<имя S>] - разрывает связь с удаленным компьютером, оставаясь в командном режиме;

quit - то же, что и close, но с выходом из командного режима (из ftp);

cd [<имя каталога в S>] - выбор каталога на сервере;

get [<имя файла в S> [<имя файла в T>]] - перепись файла с S на T;

mget [<имена файлов в S>] - то же, что и get, но нескольких файлов;

put [<имя файла в T> [<имя файла в S>]] - обратная перепись (допускается не во всех случаях);

mput <имена файлов в S> - то же, что и put, но более одного файла;

user <имя/пароль> - идентификация пользователя на сервере.

Пример последовательности команд при работе по протоколу FTP:

ftp> cd techno - переход в каталог techno;

ftp> ascii - установка передачи текста в коде ASCII (если указать "binary", то будут передаваться двоичные данные);

ftp> get test test.txt - перепись файла test в компьютер пользователя под именем test.txt;

ftp> quit - конец.

Во время сеанса связи инициируется управляющий (командный) процесс, который осуществляется через протокол Telnet и существует во время всего сеанса связи. Процесс передачи файла существует только на время передачи.

Протокол эмуляции терминала Telnet. С помощью этого протокола пользователь сети Internet может работать на удаленном компьютере. Связь устанавливается при обращении к Telnet-программе командой

telnet: <имя базы данных или системы каталогов> или <имя удаленного компьютера S> .

После установления связи все, что пользователь набирает на клавиатуре своего компьютера, передается на удаленный компьютер S, а содержимое экрана удаленного компьютера S отображается на экране пользователя. Для возвращения в свой компьютер (т.е. в командный режим клиентской программы Telnet) нужно нажать соответствующую клавишу (Ctrl-). Примерами команд в клиентской программе могут служить: установление связи (open), возвращение в командный режим (close), завершение работы (quit). Передача сообщений при работе с Telnet осуществляется с помощью средств FTP.

Протокол Telnet должен иметь возможность работать в условиях разных аппаратных платформ клиента и сервера, что достигается через промежуточный виртуальный терминал.

Электронная почта (E-mail) - это средство обмена сообщениями по электронным коммуникациям (в режиме off-line). По электронной почте

можно пересылать текстовые сообщения и архивированные файлы. В архивированных файлах могут содержаться данные в различных форматах.

Разработан ряд протоколов электронной почты для прикладного уровня. Наиболее популярны среди них протоколы SMTP в стеке протоколов TCP/IP и X.400 в модели ISO. Расширение числа возможных кодировок и форматов данных по сравнению с SMTP сделано в протоколе MIME (Multipurpose Internet Mail Extensions). На их базе разработано программное обеспечение E-mail, способное работать в обоих протоколах. Оно включает программы почтовых серверов и клиентов. Применение MIME упрощает пересылку графических и звуковых файлов, реализацию шифрования и электронной подписи.

На ЭВМ пользователя должна быть установлена программа-клиент, поддерживающая функции создания, передачи и приема сообщений. На почтовом сервере, выделяемом в корпоративной или локальной сети, организуется промежуточное хранение поступающих сообщений. Связь индивидуальных пользователей с почтовым сервером осуществляется по протоколам IMAP или POP3. Для индивидуального пользователя, общающегося с другими абонентами по телефонной сети общего пользования, такое промежуточное хранение возможно на собственном компьютере, но тогда требуется либо круглосуточное включение компьютера, либо предварительная договоренность о времени связи.

В территориальных сетях почтовые сообщения проходят через ряд промежуточных федеральных или региональных узлов. В таких узлах устанавливается программное обеспечение (так называемый агент передачи сообщений), выполняющее функции сортировки и маршрутизации сообщений.

Примерами программных систем электронной почты, выполняющих все отмеченные функции E-mail, могут служить Microsoft Mail, Outlook Express или Microsoft Outlook. Они позволяют адресовать и

переадресовывать сообщения индивидуальному пользователю и/или группе пользователей, использовать доску объявлений, осуществлять поиск сообщений, пришедших в почтовый сервер, по контексту, адресу, времени отправки.

В настоящее время при разработке многих программных систем предусматривается интерфейс со средствами электронной почты. Клиентские программы E-mail стараются включать в Web-браузеры сети Internet, а также в такие прикладные программные системы, как АСУ, САПР, системы документооборота.

Письма в E-mail состоят из заголовка и тела (текста). В заголовке указывается кому предназначено письмо, от кого оно поступило, кому посланы копии, дата отправки, указатель ключа, по которому пользователь может определить ключ для декодирования текста. В протоколе IMAP (Internet Message Access Protocol) сначала клиенту передается заголовок, а текст остается на сервере, затем пользователь при желании может получить и весь текст. В протоколе POP3 при обращении к почтовому серверу на клиентский узел переписывается все сообщение.

Вспомогательные системы Archie и Whois в Internet.

Вспомогательные средства облегчают поиск в разветвленных сетях. В Internet к ним относится Archie - информационная система для просмотра содержимого FTP-серверов. Вместо утомительной навигации вручную по каталогам система позволяет искать данные по ключевым словам или по образцу. Другая вспомогательная система в Internet - система Whois - справочник по абонентам электронной почты.

7.3. Виды конференц-связи

Телеконференции - доступ к информации, выделенной для группового использования в отдельных конференциях (newsgroups).

Возможны глобальные и локальные телеконференции. Основные функции программного обеспечения телеконференций: включение материалов в телеконференцию, рассылка извещений о новых поступивших материалах, выполнение заказов. Возможны режимы E-mail и on-line.

Самая крупная система телеконференций - USENET. В USENET информация организована иерархически. Сообщения рассылаются или лавинообразно, или через списки рассылки. В режиме on-line можно прочитать список сообщений, а затем и выбранное сообщение. В режиме off-line из списка выбирается сообщение и на него посыпается заказ.

Существуют также средства аудиоконференций (голосовых телеконференций). Вызов, соединение, разговор происходят для пользователя как в обычном телефоне, но связь идет через Internet.

Электронная "доска объявлений" BBS (Bulletin Board System) - технология, близкая по функциональному назначению к телеконференции, позволяет централизованно и оперативно направлять сообщения для многих пользователей. Программное обеспечение BBS сочетает в себе средства электронной почты, телеконференций и обмена файлами. Примеры программ, в которых имеются средства BBS, - Lotus Notes, World-group.

В настоящее время интенсивно развиваются технологии настольной конференц-связи в реальном масштабе времени. В зависимости от вида разделяемой пользователями информации возможны несколько уровней настольной конференц-связи:

- простая E-mail сессия,
- совместная работа над документом без голосовой связи (shared whiteboard - разделяемая "доска"),
- совместная работа над документом с голосовой связью (разновидность аудиоконференций),
- видеоконференция.

По мере повышения уровня возрастают требования к пропускной способности используемых каналов передачи данных. Для простых видов конференц-связи, а также и для аудиоконференций при применении современных эффективных способов сжатия информации можно использовать даже обычные телефонные линии, способные передавать информацию со скоростью от 8-10 кбит/с.

В зависимости от числа участников и способа интерактивной связи между ними различают двухточечную (unicast), широковещательную (broadcast) и многоточечную (multicast) конференции. Если в широковещательной конференции информация от центрального узла доставляется всем участникам, то в многоточечной конференции она рассыпается избирательно, т.е. одновременно может идти обмен разной информацией внутри нескольких подгрупп одной группы пользователей.

Наиболее очевидными областями применения настольной конференц-связи являются дистанционное обучение, медицинские консультации, различные бизнес-приложения.

Программное обеспечение телеконференций включает серверную и клиентскую части. В клиентской программе должны быть, как минимум, средства E-mail, многооконный текстовый редактор (так как принимаемый и отправляемый партнеру тексты помещаются в разные окна, отдельное окно может быть выделено для видео в случае видеоконференций), средства файлового обмена.

Серверная часть (MCU - Multipoint Control Unit) служит для распределения потока данных между пользователями с согласованием форматов окон с видеоинформацией, способов сжатия данных, скоростей потоков, идущих от разных сетей (пользователей).

Видеоконференция - способ связи, включающий передачу видеоизображений по телекоммуникационным каналам связи с возможностями интерактивного общения (в режиме on-line). Очевидно, что

требования к пропускной способности каналов передачи данных в видеоконференциях существенно выше, чем в обычных телеконференциях. Видеоконференции стали доступными после развития высокоскоростных каналов связи и эффективных алгоритмов сжатия данных при их передаче.

Система видеоконференции включает дистанционно управляемую видеокамеру, монитор, микрофоны, динамики, устройство для считывания графических документов, кодеки (кодек – специальное устройство для сжатия информации, само слово образовано первыми слогами слов кодирование и декодирование).

При использовании в системе видеоконференции аналогового телевидения достигается самое высокое качество передачи динамических изображений, однако для этого требуется полоса около 5 МГц, что при кодово-импульсной модуляции и кодировании отсчетов восьмибитовыми комбинациями эквивалентно пропускной способности каналов 80 Мбит/с.

Цифровые видеосистемы также используют видеокамеру, монитор, микрофон, динамик, кодек. Связь чаще всего организуется по цифровым каналам (ISDN). Качество передачи изображения не так высоко, поэтому этот способ обходится значительно дешевле аналогового телевидения.

Для организации конференц-связи имеется группа стандартов серии T.120, разработанных ITU. Стандарты T.122/125 относятся к службе многоточечных соединений, T.126 - к whiteboard технологии, T.127 - к передаче файлов при многоточечной связи. Стандарт T.123 содержит описание транспортных протоколов, которые могут использоваться в системах конференц-связи. В стандарте T.124 разработан соответствующий язык диаграмм для пользователей с недостатками слуха или речи.

Другая группа стандартов конференц-связи H.32x посвящена реализации мультимедийных приложений в различных типах сетей. Стандарты H.320, H.321, H.322, H.323 и H.324 ориентированы соответственно на каналы N-ISDN (узкополосные), B-ISDN

(широкополосные), локальные сети с гарантированной пропускной способностью, локальные сети без гарантированной полосы пропускания и телефонные линии с коммутацией каналов. Стандарт H.310 относится к мультимедийным приложениям с высоким разрешением. В этих стандартах устанавливаются требования к сжатию информации, к протоколу передачи, к синхронизации видео и звука.

7.4. Web-технологии

В сети Internet имеется уникальная информационная система WWW (World Wide Web - всемирная паутина). Другое ее краткое название - Web. Она представляет собой распределенное хранилище информации, а также серверное и клиентское программное обеспечение для обслуживания этой информации и доступа к ней.

Система WWW использует **гипертекст** - структурированный текст с введением в него перекрестных ссылок, отражающих смысловые связи частей текста. Слова-ссылки выделяются цветом и/или подчеркиванием. Выбор ссылки вызывает на экран связанный со словом-ссылкой текст или рисунок. Можно искать нужный материал по ключевым словам.

Информация, доступная по Web-технологии, хранится на Web-серверах. Сервер имеет специальную программу, постоянно отслеживающую приход на определенный порт (обычно это порт 80) запросов от клиентов. Сервер удовлетворяет запросы, посылая клиенту содержимое запрошенных Web-страниц или результаты выполнения запрошенных процедур.

Клиентские программы WWW называют **браузерами** (browsers). Имеются текстовые (например, Lynx) и графические (наиболее известны Netscape Navigator и MS Explorer) браузеры. В браузерах имеются команды листания, перехода к предыдущему или последующему документу, печати, перехода по гипертекстовой ссылке и т.п. Из браузеров доступны различные сервисы - FTP, Gopher, USENET, E-mail. Для подготовки материалов для их

включения в базу WWW разработаны специальный язык HTML (HyperText Markup Language) и реализующие его программные редакторы, например Internet Assistant в составе редактора Word. Подготовка документов предусмотрена и в составе большинства браузеров.

Для связи Web-серверов и клиентов разработан протокол HTTP, работающий на базе TCP/IP. Web-сервер получает запрос от браузера, находит соответствующий запросу файл и передает его для просмотра в браузер. Популярными серверами являются Apache, Netscape Enterprise Server и Microsoft Internet Information Server (IIS), которые могут работать как в Unix, так и в Windows NT. Все три сервера поддерживают язык CGI, имеют встроенный HTML-редактор. Кроме того, в первых двух из них поддерживается стандарт шифрования SSL (Secure Sockets Layer) для защиты передаваемых по сети данных от несанкционированного доступа. Опыт показывает, что для крупных серверов предпочтительнее платформа Unix, тогда как для серверов с малым числом транзакций лучше подходит ОС Windows NT.

В настоящее время для облегчения поиска информации в Internet применяют информационно-поисковые системы (ИПС), располагаемые на доступных пользователям Internet-серверах. В этих системах собирается, индексируется и регистрируется информация о документах, имеющихся в обслуживаемой группе Web-серверов. Индексируются или все значащие слова, имеющиеся в документах, или только слова из заголовков. Пользователю предоставляется возможность обращаться к серверу с запросами на естественном языке, со сложными запросами, включающими логические связки. Примером таких ИПС может служить AltaVista, Rambler. Например, для функционирования AltaVista фирма DEC выделила 6 компьютеров, самый мощный из них - 10-процессорная ЭВМ Alpha-8400 с базой данных объемом более 45 Гбайт.

7.5. Языки и средства создания Web-приложений

Бурное развитие глобальной сети Internet оказывает огромное влияние на все сферы деятельности человека. Internet вызвал революционные изменения в индустрии программного обеспечения. Появилась новая категория приложений, специально разработанных для Интернета и учитывающих особенность серверов Web. Поэтому программы для Интернета часто называют приложениями Web. Например, организация образования через Интернет требует специальной организации учебных пособий, которые могут быть подготовлены в формате HTML, рассчитанном на просмотр учебника в одном из браузеров (Internet Explorer, Netscape Navigator). Для создания документов в формате HTML существуют различные программные средства. Например, текстовый редактор Word позволяет сохранять документ и отдельные его части в формате HTML и даже организовывать гиперсвязи между HTML-файлами. Для получения более сложного HTML-документа требуется навыки программирования на языке HTML.

Язык HTML - гипертекстовый язык, описывающий структуру документа, вид которого на экране определяется браузером.

Описание на HTML - это текст в формате ASCII и последовательность включенных в него команд (управляющих кодов, называемых также *дескрипторами*, или *тегами*). Эти команды расставляются в нужных местах текста, определяя шрифты, переносы, появление графических изображений, ссылки и т.п.

Команды имеют форму < >, где между скобками записывается имя команды.

Не вдаваясь в детали языка HTML, которые легко могут быть найдены в соответствующих книгах, приведем только необходимые сведения о нем.

Если открыть программу Блокнот, написать в нем следующие строчки
<HTML>

```
<HEAD>
<TITLE>Информационный раздел 1</TITLE>
</HEAD>
<BODY>
    <FONT FACE="Times New Roman">
        <P>Информационный раздел 1.</P>
        <P>Текст</P>
    </FONT>
</BODY>
</HTML>
```

и сохранить их в виде файла с расширением .html, то открытие этого файла с помощью браузера Internet Explorer приведет к появлению в окне браузера простейшего HTML-документа (рис. 7.2).

Теги `<HTML>`, `</HTML>` определяют начало и окончание HTML-документа.

Теги `<HEAD>`, `</HEAD>` определяют информацию, относящуюся к разделу заголовка HTML-документа, в частности, шрифт и текст самого заголовка:

```
<TITLE>Информационный раздел 1</TITLE>
```

Теги `<BODY>`, `</BODY>` определяют содержание HTML-документа и свойства этого содержания:

- тип шрифта

```
<FONT FACE = "Times New Roman">
</FONT>
```

- текстовое содержание

```
<P>Информационный раздел 1.</P>
<P>Текст</P>
```

Команды форматирования текста (дескрипторы компоновки):

Теги <P>, </P> определяют начало и окончание текстового фрагмента, начинающегося с новой строки;

 - перевод строки;

<HR> - перевод строки с печатью горизонтальной линии, разделяющей части текста.

Команды форматирования заголовков (дескрипторы стиля):

<H1> Текст </H1> - текст печатается наиболее крупным шрифтом, используется для заголовков верхнего уровня;

<H2> Текст </H2> - для следующего уровня и т.д. вплоть до команды <H6>.

Команды форматирования символов представлены парными символами B, I, U. Текст между открывающей и закрывающей командами будет выделен полужирно, курсивом, подчеркиванием соответственно.

Дескрипторы связи

В командах вставки графики и гипертекстовых ссылок используются адреса вставляемого или ссылочного материала, называемые URL (Uniform Resource Locator). Ссылаться можно как на нужные места в том же документе, в котором поставлена ссылка, так и на другие файлы, находящиеся в любом месте сети. URL может представлять собой имя файла в данном узле сети или IP-имя другого узла с указанием местоположения файла в этом узле и, возможно, также метки внутри этого файла.

Команда гипертекстовой ссылки:

Текст .

Текст в окне будет выделен цветом или подчеркиванием. Можно ссылаться на определенное место в документе. Тогда

 Текст .

Сама метка в документе имеет вид:

 Текст .

Ссылки на фрагменты данного документа можно упростить:

Текст .

Для того чтобы встроить растровое изображение в документ HTML, необходимо использовать тег . Общий вид этого тега показан ниже:

<IMG SRC="Адрес_файла_изображения"

NAME="Имя_изображения"

...

WIDTH="Ширина" HEIGHT="Высота">

Здесь указаны только три параметра. Полный список параметров тега с кратким их описанием находится в таблице 7.1.

**Таблица 7.1. Параметры тега **

Параметр	Описание
SRC	Адрес URL файла с растровым графическим изображением
NAME	Имя объекта, соответствующего растровому графическому изображению. Это имя может быть использовано для ссылки на объект в клиентском сценарии
ALT	Текстовая строка, которая отображается в тех случаях, когда браузер не может показывать графические изображения или когда такая возможность отключена
ALIGN	Выравнивание текста относительно графического изображения: LEFT - по левой границе; RIGHT - по правой границе; TOP - по верхней границе; MIDDLE - по центру изображения; BOTTOM - по нижней границе; TEXTTOP - выравнивание по верхней границе относительно самых высоких символов в текстовой строке; ABSMIDDLE - выравнивание середины текстовой строки относительно середины изображения; BASELINE – выравнивание нижней рамки изображения относительно базовой линии текстовой строки; ABSBOTTOM - выравнивание нижней границы изображения относительно нижней границы текущей строки
HEIGHT	Высота изображения в пикселях
WIDTH	Ширина изображения в пикселях
BORDER	Ширина рамки (в пикселях) вокруг изображения (используется только браузером Netscape Navigator)
HSPACE	Ширина (в пикселях) свободного пространства, отделяющего изображение от текста по горизонтали
VSPACE	Ширина (в пикселях) свободного пространства, отделяющего изображение от текста по вертикали
USEMAP	Адрес URL файла, содержащего так называемую карту изображения, которая

	используется для сегментированной графики
ISMAP	Этот параметр указывает, что данное изображение является сегментированным

Параметры тега определяют адрес файла с изображением, выравнивание текста, расположенного возле изображения, и т. д. С помощью параметров HEIGHT и WIDTH выполняется масштабирование графических изображений. Значение этих параметров указано в процентах от ширины окна просмотра.

Масштабирование позволяет подготовить графический файл весьма небольшого размера: он занимает значительную площадь в окне браузера, но быстро передается через Интернет. Однако, масштабирование сегментированных графических и фоновых изображений невозможно.

Если в документе HTML размещено несколько растровых изображений, то можно адресовать соответствующие объекты как элементы массива document.images. Например, первое изображение адресуется следующим образом: document.images[0]. Однако в некоторых случаях удобнее пользоваться именами изображений, определенными параметром NAME оператора . Объект-изображение имеет свойство src, соответствующее параметру SRC оператора . Адресуясь к этому свойству, можно не только определять текущий адрес URL изображения, но и задавать новый.

Рассмотрим фреймовую структуру организации HTML-документа, когда окно просмотрщика (браузера) разделено на несколько частей, в каждую из которых выводится свой HTML-документ. Такая организация наиболее удобна для организации сайта или компьютерного учебника, т.к. позволяет совмещать удобную навигацию в пространстве сайта или учебника с удобным представлением его информации. Например, удобно разделять окно браузера на три части (три фрейма): в левой части расположить оглавление сайта (учебника) с гиперссылками на соответствующие информационные разделы, в правой части выводить содержание

информационного раздела, к которому произведено обращение из фрейма оглавления, а в верхней части выводить название соответствующего информационного раздела (рис. 7.3).

Для того чтобы объединить несколько страниц HTML при помощи фреймов, нужно подготовить специальный документ HTML, в котором описаны такие параметры фреймов, как их размер и расположение.

Особенность такого документа - отсутствие на своем обычном месте области тела документа, выделенного тегами <BODY> и </BODY>. Вместо этого в файле описания фреймов присутствуют теги <FRAMESET>, </FRAMESET>, <NOFRAME> и </NOFRAME>:

```
<html>
<head>
...
</head>
<frameset rows="Высота_строки" cols="Ширина_колонки">
    <frame src="Адрес_URL" name="Имя_фрейма">
    ...
    <frame src="Адрес_URL" name="Имя_фрейма">
<noframe>
    <body>
    ...
    </body>
</noframe>
</frameset>
</html>
```

Параметры rows и cols тега <PRAMESET> определяют размеры фреймов и задаются в виде списка значений, разделенных запятой.

Для тех браузеров, которые не могут работать с фреймами, необходимо

подготовить документ HTML, расположив его тело между операторами <NOFRAMES> и </NOFRAME>. В этот документ стоит поместить сообщение о том, что для просмотра данной страницы Web необходимо применять более современный браузер.

Параметры тега <FRAMESET>

Рассмотрим подробнее параметры оператора <FRAMESET>, предназначенного для определения набора фреймов. Эти параметры описаны в таблице 7.2.

Таблица 7.2. Параметры тега <FRAMESET>

Параметр	Описание
COLS	Ширина колонки в процентах, пикселях или ее относительный размер
ROWS	Высота строки в процентах, пикселях или ее относительный размер
FRAMEBORDER	Если значение этого параметра равно 1, фреймы будут ограничены трехмерной рамкой, ширина которой задается в пикселях. В том случае когда указано значение 0, рамка не создается
BORDER	Используется только браузером Netscape Navigator. Задает толщину рамки фрейма в пикселях
FRAMESPACING	С помощью этого параметра задается дополнительное расстояние между фреймами в пикселях

Параметры COLS и ROWS нужны в том случае, когда фреймы, определенные в наборе, располагаются в виде таблицы. Первый из этих параметров указывает ширину колонки, а второй - высоту строки. Если фреймы располагаются в одном столбце, параметр COLS указывать не надо. Аналогично, если фреймы занимают только одну строку, не нужно указывать параметр ROWS.

Можно задать значения для параметров COLS и ROWS либо в процентном отношении соответственно к ширине и высоте окна браузера, либо в пикселях. Если вместо значения указан символ “*”, колонка или строка занимают всю оставшуюся часть окна.

Например, в следующей строке задана высота первого фрейма, равная 80 пикселям, а второй фрейм занимает всю нижнюю часть окна браузера:

```
<FRAMESET ROWS="90,*">
```

В следующем примере два фрейма, расположенные рядом, занимают соответственно 20% и 80% ширины окна браузера.

```
<FRAMESET COLS="20%, 80%">
```

Параметры оператора <FRAME>

Между тегами <FRAMESET> и </FRAMESET> располагаются теги <FRAME>, определяющие параметры отдельных фреймов. Это параметры SRC и NAME. Первый задает адрес URL документа HTML, который будет загружен в данный фрейм, а второй - имя фрейма, которое можно использовать в клиентском сценарии для адресации объектов, расположенных во фрейме. Параметры тега <FRAME> приведены в таблице 3.2.

Таблица 3.2. Параметры тега <FRAME>

Параметр	Описание
MARGINHEIGHT	Используется только для "плавающих" фреймов в браузере Microsoft Internet Explorer. Задает выравнивание фрейма или текста, расположенного рядом с фреймом. Этот параметр может принимать следующие значения: LEFT, CENTER, RIGHT, TOP, BOTTOM
MARGINWIDTH	Размер отступа (в пикселях) по вертикали от границ фрейма
FRAMEBORDER	Размер отступа (в пикселях) по горизонтали от границ фрейма. Если значение этого параметра равно 1, фреймы ограничены трехмерной рамкой, ширина которой задается в пикселях. В том случае когда указано значение 0, рамка не создается
NAME	Этот параметр задает имя фрейма, которое используется в теге ссылки <A> для указания, в какой фрейм нужно загрузить новый документ
NORESIZE	Если указан этот параметр, пользователь не сможет изменять размеры фрейма, передвигая его границы мышью
SCROLLING	Параметр SCROLLING определяет, нужно ли создавать полосы просмотра для пролистывания содержимого фрейма. Для этого параметра можно указывать следующие значения: YES - полосы просмотра создаются всегда; NO - полосы просмотра не создаются; AUTO - полосы просмотра создаются только при необходимости, когда документ HTML не помещается полностью в окне фрейма
SRC	Адрес URL файла с документом HTML, который загружается в окно фрейма

Взаимодействие между фреймами

Средства клиентских сценариев, составленных на языках программирования, позволяют наделить фреймы возможностями, недостижимыми при использовании одного лишь языка разметки гипертекста HTML. Например, один из фреймов может содержать ссылки на документы, которые при активизации этих ссылок загружаются в окно другого фрейма. Клиентский сценарий позволит таким образом загружать не один документ, а одновременно несколько документов в разные фреймы.

Большую известность приобрели технология и язык программирования сетевых приложений Java, разработанные фирмой Sun Microsystems для систем распределенных вычислений.

Язык Java объектно-ориентированный, прототипом является C++, но Java более прост в использовании (так, например, убраны указатели), в нем введены многопотковость и дополнительная защита от вирусов.

Для пользователей важны также следующие черты языка:

аппаратная независимость (мобильность) за счет создания приложений в виде байт-кодов для некоторой виртуальной машины - каждая аппаратная платформа интерпретирует эти байт-коды; благодаря введению компиляции потеря эффективности, присущая интерпретации, здесь менее значительна;

интеграция с браузерами;

используемые программные объекты могут находиться в разных узлах, интерпретатор находит их и загружает в компьютер пользователя.

Другими словами, в узле-клиенте достаточно иметь лишь браузер, все остальное можно получить по сети. Однако при этом обостряется проблема информационной безопасности. В связи с этим, загружаемым по сети программам (они называются *апплетами*) обычно запрещается обновлять и читать файлы, кроме тех, которые находятся на компьютере самого аплета.

Java-апплеты доступны из HTML-документов (обращение к ним производится через тег <applet>), хотя могут использоваться и независимо от

них. При обращении к аплету он компилируется на сервере, а для исполнения передается клиенту вместе с Web-страницей.

Большое распространение получил интерфейс CGI (Common Gateway Interface - общий шлюзовой интерфейс) - программное обеспечение связи HTML браузеров с другими прикладными программами и/или текстами, находящимися на серверной стороне. Программа CGI - посредник между браузером и приложениями. Обычно программа CGI находится на сервере в специальном каталоге `CGI_BIN`, она является обработчиком запросов, идущих от браузера. Обращение к файлу из этого каталога означает запуск соответствующего обработчика. Если браузер обращается к документу не в HTML формате, то CGI преобразует форму документа в HTML и возвращает ее браузеру.

В гипертекстовых документах также широко используется JavaScript - язык и интерпретатор этого языка для генерации и управления просмотром составных гипертекстовых документов. JavaScript более прост, чем Java, и тексты JavaScript исполняются быстрее, чем тексты Java или запросы к CGI, поскольку обработчики событий JavaScript реализованы в браузере, а не на сервере. Тексты на JavaScript записываются непосредственно в HTML документе с помощью специальных тегов и имеют вид

```
<SCRIPT LANGUAGE = "javascript"><!-- . . . //--> </SCRIPT>
```

где `<!-- . . . //-->` - текст в виде комментария. В отличие от Java программы на JavaScript полностью интерпретируются в браузере.

Для разработки приложений в Internet уже созданы специальные языки и средства. Это, кроме упомянутых языков, также язык Visual Basic Script (VBScript).

Microsoft разработала технологию создания и использования интерактивных сетевых приложений, названную ActiveX. Некоторые компоненты ActiveX передаются в составе HTML-документов, другие служат для взаимодействия сервера с приложениями. Microsoft предлагает среду

разработки Web-документов и приложений, включающую ряд продуктов, например:

Internet Assistant - служит для создания HTML-документов, использует возможности редактора Word, взаимно преобразует форматы документов HTML и Word;

FrontPage - применяется Web-мастерами и администраторами для сопровождения гипертекстовой информационной базы;

Internet Studio - помогает художественному оформлению Web-страниц;

Visual J++ в составе компилятора Java, набора JDK, средств взаимодействия Java-апплетов и ActiveX-компонентов, и др.

Internet-функции становятся неотъемлемой частью сетевых операционных систем. Так, в ОС Windows NT, начиная с версии 4.0, входит Internet-сервер IIS (Internet Information Server), реализующий технологии WWW, Gopher, FTP, ISAPI.

В качестве примера рассмотрим методику создания простейшего компьютерного учебника в формате HTML, использующего фреймовую структуру.

Методика создания компьютерного учебника в формате HTML:

1) Подготовить все разделы учебника (оглавление, названия информационных разделов, главы, параграфы, примеры, контрольные вопросы и т.д.) **в текстовом редакторе Word и сохранить из в виде отдельных файлов**, например, oglavlenie.doc, title1.doc, title2.doc, ..., titleN.doc, ch1. doc, 1.1. doc, 1.2. doc, ..., 1.N. doc, ch2. doc, 2.1. doc, 2.2. doc, ..., 2.N. doc, ..., chN. doc, N.1. doc, N.2. doc, ..., N.N. doc.

2) Преобразовать все файлы разделов учебника в формат HTML, для чего использовать опцию меню “Файл\Сохранить в формате HTML”. Например, oglavlenie.html, title1. html, title2. html, ..., titleN. html, ch1.html, 1.1.html, 1.2.html, ..., 1.N.html, ch2.html, 2.1.html, 2.2.html, ..., 2.N.html, ..., chN.html, N.1.html, N.2.html, ..., N.N.html.

3) Организовать основной загрузочный файл Учебника index.html, из которого будет осуществляться управление Учебником.

В нашем случае создается HTML-файл с именем index.html, который является основным (первоначально загружающимся) файлом компьютерного учебника, из которого осуществляется все дальнейшее управление Учебником.

```
<html>
<head>
    <title>Название учебника</title>
</head>
<frameset FRAMEBORDER="1" rows="100,*">
    <frame SCROLLING="no" NAME="title" SRC="title.html" MARGINHEIGHT="10" MARGINWIDTH="10"/>
<frameset FRAMEBORDER="1" cols="300,*">
    <frame SCROLLING="auto" NAME="oglavlenie"
        SRC="oglavlenie.html">
    <frame SCROLLING="auto" NAME="main" SRC="main.html">
</frameset>
<noframes>
<body BGCOLOR="#FFFFFF">
</body>
</noframes>
</frameset>
</html>
```

В нашем примере мы создаем три фрейма с именами oglavlenie, title и main. Результатом открытия этого файла в браузере является появление окна, представленного на рис. 7.4.

4) Организовать гипертекстовую среду Учебника. Последним шагом в разработке компьютерного учебника является реализация гиперссылок из фрейма oglavlenie, загружающих соответствующие HTML-документы во

фреймы title и main. Для этого следует открыть в программе Блокнот файл oglavlenie.html и вставить в него после тега <body> следующую запись

```
<P><SCRIPT LANGUAGE="JavaScript"><!--  
    function loadPage(szNewURL,szTitle)  
    {  
        parent.main.window.location.href=szNewURL;  
        parent.title.window.location.href=szTitle;  
    }  
// --></SCRIPT>
```

В каждую строку оглавления, из которой осуществляется гиперссылка к какому-либо информационному разделу, следует вставить запись, указывающую, какие файлы будут загружаться во фреймы main и title:

```
<A HREF="javascript:loadPage('Имя html-файла, помещаемого  
в фрейм main',' Имя html-файла, помещаемого в фрейм title');">
```

Получаем следующий код для файла oglavlenie.html:

```
<HTML>  
<HEAD>  
<TITLE>ОГЛАВЛЕНИЕ</TITLE>  
</HEAD>  
<BODY>  
<P><SCRIPT LANGUAGE="JavaScript"><!--  
    function loadPage(szNewURL,szTitle)  
    {  
        parent.main.window.location.href=szNewURL;  
        parent.title.window.location.href=szTitle;  
    }  
// --></SCRIPT>  
<FONT FACE="Times New Roman"><P>ОГЛАВЛЕНИЕ</P>
```

```

<P><A HREF="javascript:loadPage('ch1.html','title.html');">Глава1.</P>
</FONT>

<P><A HREF="javascript:loadPage('1.1.html','title.html');">1.1.</P>
<P><A HREF="javascript:loadPage('1.2.html','title.html');">1.2.</P>
<P><A HREF="javascript:loadPage('1.N.html','title.html');">1.N.</P>
<FONT FACE="Times New Roman">
<P><A HREF="javascript:loadPage('ch2.html','title.html');">Глава
2.</P></FONT>

<P><A HREF="javascript:loadPage('2.1.html','title.html');">2.1.</P>
<P><A HREF="javascript:loadPage('2.2.html','title.html');">2.2.</P>
<P><A HREF="javascript:loadPage('2.N.html','title.html');">2.N.</P>
<FONT FACE="Times New Roman">
<P><A HREF="javascript:loadPage('chN.html','title.html');">Глава N.</P>
</FONT>

<P><A HREF="javascript:loadPage('N.1.html','title.html');">N.1.</P>
<P><A HREF="javascript:loadPage('N.2.html','title.html');">N.2.</P>
<P><A HREF="javascript:loadPage('N.N.html','title.html');">N.N.</P>
</BODY>
</HTML>

```

После указанной процедуры открытый в браузере основной управляющий файл Учебника index.html, приобретает вид, представленный на рис. 7.5.

Контрольные вопросы к главе 7

1. Опишите структуру территориальных сетей.
2. Какие типовые информационные услуги предоставляют территориальные сети?
3. Охарактеризуйте протоколы файлового обмена.
4. Охарактеризуйте протоколы электронной почты.

5. Охарактеризуйте протоколы дистанционного управления.
6. Какие виды конференц-связи применяются в современных телекоммуникациях?
7. Охарактеризуйте современные WEB-технологии и области их применения.
8. Расскажите о языках и средствах создания WEB-приложений.
9. Составьте программу на языке HTML для создания простейшего гипертекстового документа.
10. Составьте программу на языке HTML для создания простейшего HTML-документа фреймовой структуры.

ПРИЛОЖЕНИЕ. ПРИМЕР ПОДХОДА К РАЗРАБОТКЕ ЛВС ПРЕДПРИЯТИЯ, ВЫБОРУ ТИПА СЕРВЕРА С ВОЗМОЖНОСТЬЮ РАСШИРЕНИЯ СЕТИ.

Технические требования к разрабатываемой сети:

- число помещений - 7.
- количество персональных компьютеров в помещениях - 18
- построение сети выполнить на линиях связи типа "витая пара".
- управление сервером должно осуществляться операционной системой Windows 2000 Server.

П 1. ФУНКЦИОНАЛЬНАЯ СХЕМА КОРПОРАТИВНОЙ ЛОКАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ “НИККО СТАЙЛ”. ИНФОРМАЦИОННЫЕ ПОТОКИ В ЛВС ПРЕДПРИЯТИЯ

Рассмотрим организационно-штатную структуру подразделения. Во главе подразделения стоит генеральный директор предприятия. В структуру предприятия входят 4 отдела. Каждый отдел имеет в подчинении разное количество сотрудников. На предприятии существует 3 типа потоков информации: распоряжения, доклады, оперативная информация (информация, изменяющаяся во время работы предприятия, т.е. количество заказов, наличие товаров на складе и т.п.).

В структуре предприятия существует 4 отдела (рис. П.1):

Демонстрационный отдел (3 человека). Данный отдел занимается демонстрацией образцов предлагаемой продукции клиентам.

Отдел оформления заказов (8 человек). Данный отдел занимается составлением заказа клиента и выдачей оформленного и

оплаченного заказа, т.е. непосредственно работает с клиентами.

Отдел информационной и технической поддержки (2 человека).

Данный отдел занимается поддержкой работоспособности программно-аппаратного комплекса.

Финансовый отдел (4 человека). Этот отдел состоит из 3 служб (бухгалтерская, кадровая, касса).

Всего в подразделении задействовано 18 человек, каждому из которых предполагается выделить в пользование персональный компьютер.

П 2. РАЗРАБОТКА СТРУКТУРЫ СЕТИ

П 2.1. Выбор структуры управления сетью

Каждая фирма формулирует собственные требования к конфигурации сети, определяемые характером решаемых задач. В первую очередь необходимо определить, сколько человек будут работать в сети. От этого решения, по существу, будут зависеть все последующие этапы создания сети.

Количество рабочих станций напрямую зависит от предполагаемого числа сотрудников. Другим фактором является иерархия компании. Для фирмы с горизонтальной структурой, где все сотрудники должны иметь доступ к данным друг друга, оптимальным решением является простая одноранговая сеть.

Фирме, построенной по принципу вертикальной структуры, в которой точно известно, какой сотрудник и к какой информации должен иметь доступ, следует ориентироваться на более дорогой вариант сети – с выделенным сервером. Только в такой сети существует возможность администрирования прав доступа к определенным ресурсам сразу для группы пользователей.

К примеру, на предприятии имеется 18 рабочих станций, которые и требуется объединить в корпоративную сеть. Причем они должны быть объединены в следующие группы:

директор предприятия – 1 рабочая станция;
демонстрационный отдел - 3 рабочих станции;
отдел оформления заказов – 8 рабочих станций;
отдел информационной и технической поддержки – 2 рабочих станции;
финансовый отдел – 4 рабочих станции.

Так как на предприятии присутствуют несколько отделов, каждый из которых занимается определенной деятельностью и, соответственно, работает с разной информацией, то, следовательно, мы имеем вертикальную структуру предприятия, в которой осуществляется разграниченный доступ к информации.

Одним из главных этапов планирования является создание предварительной схемы. При этом в зависимости от типа сети возникает вопрос об ограничении длины кабельного сегмента. Это может быть несущественно для небольшого офиса, однако, если сеть охватывает несколько этажей здания, проблема предстает в совершенно ином свете. В таком случае необходима установка дополнительных *репитеров* (repeater).

В ситуации с рассматриваемым предприятием вся сеть будет располагаться на одном этаже, и расстояние между сегментами сети не столь велико, чтобы требовалось использование репитеров.

П 2.2 План помещений

План помещения влияет на выбор топологии сети значительно сильнее, чем это может показаться на первый взгляд (рис. П.2).

После определения места установки сервера можно сразу определить, какое количество кабеля потребуется.

П 2.3 Размещение сервера

В отличие от установки одноранговой сети, при построении ЛВС с сервером возникает еще один вопрос - где лучше всего установить сервер.

На выбор места влияет несколько факторов:

необходимость обеспечить постоянный доступ к серверу для технического обслуживания;

по соображениям защиты информации требуется ограничить доступ к серверу посторонних лиц.

Таким образом, выбрано единственное, возможное место установки сервера, не требующее перестройки внутренних помещений. Сервер было решено установить в помещении отдела информационной и технической поддержки, так как только это помещение удовлетворяет требованиям, то есть будет обеспечен постоянный доступ сотрудников данного отдела к серверу, помещение данного отдела в свою очередь изолировано от других, следовательно, доступ к серверу посторонних лиц будет ограничен.

П 3. ВЫБОР И ОБОСНОВАНИЕ СЕТЕВОЙ АРХИТЕКТУРЫ

Выбор топологии определяется, в частности, планировкой помещения, в котором разворачивается ЛВС. Кроме того, большое значение имеют затраты на приобретение и установку сетевого оборудования, что является важным вопросом для фирмы, разброс цен здесь также достаточно велик. В главе 7 мы рассмотрели основные виды топологии сетей, их принципы организации.

П 3.1. Топология ЛВС предприятия.

Для ЛВС была выбрана топология типа «звезда» в связи с тем, что она представляет собой более производительную структуру: каждый компьютер, в том числе и сервер, соединяется отдельным сегментом кабеля с центральным концентратором (hab).

Основным преимуществом такой сети является её устойчивость к сбоям, возникающим вследствие неполадок на отдельных ПК или из-за повреждения сетевого кабеля.

На рисунке П3 изображена топология сети предприятия.

Важнейшей характеристикой обмена информацией в локальных сетях являются так называемые методы доступа (access methods), регламентирующие порядок, в котором рабочая станция получает доступ к сетевым ресурсам и может обмениваться данными.

Так как метод CSMA/CD хорошо зарекомендовал себя именно в малых и средних сетях, для предприятия данный метод удобен. К тому же сетевая архитектура Ethernet, которую и будет использовать сеть предприятия, использует именно этот метод доступа.

Сеть на основе витой пары, в отличие от сети на базе тонкого и толстого коаксиала, строится по топологии «звезда». Чтобы построить сеть по звездообразной топологии, требуется большее количество кабеля (но цена витой пары не велика). Подобная схема имеет и неоценимое преимущество – высокую отказоустойчивость. Выход из строя одной или нескольких рабочих станций не приводит к отказу всей системы. Правда, если из строя выйдет хаб (концентратор), его отказ затронет все подключенные через него устройства.

Еще одним преимуществом данного варианта является простота расширения сети, поскольку при использовании дополнительных хабов (до четырех последовательно) появляется возможность подключения большого количества рабочих станций (до 1024). При применении неэкранированной витой пары (UTP) длина сегмента между концентратором и рабочей станцией не должна превышать 100 метров, чего не наблюдается в предприятии.

П 3.2. Сетевые ресурсы

Следующим важным аспектом планирования сети является совместное использование сетевых ресурсов (принтеров, факсов, модемов и другой периферии).

Перечисленные ресурсы могут использоваться как в одноранговых сетях, так и в сетях с выделенным сервером. Однако в случае одноранговой сети сразу выявляются её недостатки. Чтобы работать с перечисленными компонентами, их нужно установить на рабочую станцию или подключить к ней периферийные устройства. При отключении этой станции все компоненты и соответствующие службы становятся недоступными для коллективного пользования.

В сетях с сервером такой компьютер существует по определению. Сервер никогда не выключается, если не считать коротких остановок для технического обслуживания. Таким образом, обеспечивается круглосуточный доступ рабочих станций к сетевой периферии.

На предприятии имеется четыре принтера. В каждом отделе по одному принтеру.

Теперь рассмотрим вопрос подключения принтера к ЛВС. Для этого существует несколько способов:

1. Подключение к рабочей станции

Принтер подключается к той рабочей станции, которая находится к нему ближе всего, в результате чего данная рабочая станция становится сервером печати. Недостаток такого подключения в том, что при выполнении заданий на печать производительность рабочей станции на некоторое время снижается, что отрицательно скажется на работе прикладных программ при интенсивном использовании принтера. Кроме того, если машина будет выключена, сервер печати станет недоступным для других узлов.

2. Прямое подключение к серверу

Принтер подключается к параллельному порту сервера с помощью специального кабеля. В этом случае он постоянно доступен для всех рабочих станций. Недостаток подобного решения обусловлен ограничением в длине принтерного кабеля, обеспечивающего корректную передачу данных. Хотя кабель можно протянуть на 10 и более метров, его следует прокладывать в коробах или в перекрытиях, что повысит расходы на организацию сети.

3. Подключение к сети через специальный сетевой интерфейс

Принтер оборудуется сетевым интерфейсом и подключается к сети как рабочая станция. Интерфейсная карта работает как сетевой адаптер, а принтер регистрируется на сервере как узел ЛВС. Программное обеспечение сервера осуществляет передачу заданий на печать по сети непосредственно на подключенный сетевой принтер.

4. Подключение к выделенному серверу печати.

Альтернативой третьему варианту является использование специализированных серверов печати. Такой сервер представляет собой сетевой интерфейс, скомпонованный в отдельном корпусе, с одним или несколькими разъемами (портами) для подключения принтеров. Однако в данном случае использование сервера печати является непрактичным.

В нашем случае в связи с тем, что установка отдельного сервера печати увеличивает стоимость создания сети (так же как и покупка принтера с сетевым интерфейсом), то было решено подключать принтеры непосредственно к рабочим станциям в отделах. На это решение так же повлияло то, что принтеры расположены в тех помещениях, где потребность в них наибольшая. Поэтому был выбран первый тип подключения принтера.

П 4. ОРГАНИЗАЦИЯ И ОБОСНОВАНИЕ ВЫБОРА СЕТИ НА ОСНОВЕ WINDOWS 2000

Выпустив операционную систему Windows 2000, компания Microsoft сделала серьезный шаг к тому, чтобы Windows NT (NT – New Technology)

стала корпоративным стандартом проведения вычислений. Windows 2000 – один из самых крупных, когда-либо реализованных проектов создания программного обеспечения; программный код этой операционной системы содержит 40-65 миллионов символов. Над проектом работало свыше 2000 программистов, в Windows 2000 Server включены новые технологии, разработанные 24 компаниями.

Многие комбинации новых технологий обеспечили новые возможности операционной системы, в частности, иерархическую структуру системы хранения данных, при которой часто используемые файлы переносятся на ленточные накопители, откуда их при необходимости извлекают. Эта технология реализована благодаря разработкам компании High Ground. Некоторые возможности – улучшенные варианты технологий, представленных в предыдущих версиях операционной системы или в сервисных пакетах, например, организация многосессионных вычислений в сетевых серверах с использованием архитектуры «тонкого» клиента, основаны на совместных разработках компаний Microsoft и Citrix.

Если спросить у группы системных администраторов, что их больше всего беспокоит и что они хотели бы видеть в следующем поколении серверных операционных систем под названием Windows, ответ на оба вопроса будет одинаковым – стабильность и надежность. Дальше следует простота управления, дополнительные инструменты, которыми новая операционная система от Microsoft обзавелась во время разработки. Но они занимают последние места в списке.

Компания Microsoft действительно уделила повышенное внимание тому, чтобы сделать Windows 2000 более стабильной, менее склонной к зависаниям, легче настраиваемой и требующей перезагрузки в меньшем количестве случаев. Если Windows NT 4.0 требовала перезагрузки в 75 различных ситуациях, то Windows 2000 – только в пяти. Каждый, кто когда-либо сталкивался с установкой и настройкой параметров Windows NT 4.0,

будет только приветствовать эти сведения. Раньше простое изменение параметров протокола TCP/IP требовало обязательной перезагрузки системы. В Windows 2000 Server это не так.

Несмотря на то, что Windows 2000 Server еще отстает от Windows 98 по возможностям автоматической настройки работы различных устройств, она все же намного более совместима со стандартом Plug and Play, чем ее предшественницы. Не следует удивляться тому, что после замены видеоадаптера или звуковой карты система найдет новое аппаратное обеспечение. Поскольку сервер, работающий под управлением Windows 2000 Server, выключается не часто, возможности этой операционной системы по управлению питанием достаточно слабы, несмотря на заявления компании Microsoft о поддержке стандарта ACPI (Advanced Configuration and Power Interface – усовершенствованный интерфейс управления питанием).

Дополнительные улучшения в операционной системе Windows 2000 связаны с поддержкой аппаратного обеспечения. Хотя Windows NT и поддерживала экзотические типы жестких дисков, только немногих моделей потребительских цветных принтеров или сканеров могли взаимодействовать с этой операционной системой. Windows 2000 поддерживает модель драйверов Windows (WDM – Windows Driver Model), позволяющую разработчикам писать драйверы, которые будут normally взаимодействовать и с Windows 98 и Windows 2000. Драйвер должен быть откомпилирован для каждой операционной системы отдельно, но его исходный код – один и тот же. Таким образом, производители различного периферийного оборудования очень быстро выпустили драйверы для операционной системы Windows 2000 Server. Кроме того, теперь Windows 2000 Server может воздействовать с большим количеством устройств, чем предыдущие версии серверных операционных систем.

Основным элементом централизованного администрирования в Windows 2000 Server является домен. Домен - это группа серверов,

работающих под управлением Windows 2000 Server, которая функционирует, как одна система. Все серверы Windows 2000 в домене используют один и тот же набор учетных карточек пользователя, поэтому достаточно заполнить учетную карточку пользователя только на одном сервере домена, чтобы она распознавалась всеми серверами этого домена.

П 5. РАЗРАБОТКА И ОПИСАНИЕ ЛВС ПРЕДПРИЯТИЯ

П 5.1 Схема построения

ЛВС построена по топологии «звезда», хотя, если быть точнее, представляет собой дерево: все клиенты сети являются ответвлениями центрального «магистрального» канала. Но топологически вся сеть представляет собой «звезду», с центром в виде концентратора в серверной комнате отдела информационной и технической поддержки.

П 5.2 Основные административные блоки

Группирование компьютеров в рабочие группы дает два важных преимущества сетевым администраторам и пользователям. Наиболее важное - серверы домена составляют (формируют) единый административный блок, совместно использующий службу безопасности и информацию учетных карточек пользователя. Каждая рабочая группа имеет одну базу данных, содержащую учетные карточки пользователя и групп, а также установочные параметры политики безопасности.

Второе преимущество доменов сделано для удобства пользователей: когда пользователи просматривают сеть в поисках доступных ресурсов, они видят сеть, сгруппированную в домены, а не разбросанные по всей сети серверы и принтеры.

П 5.3 Конфигурирование сервера

Сетевая операционная система выполняется на сервере. С другой стороны, компьютеры-клиенты могут работать под управлением различных операционных систем. Чтобы операционная система клиента могла использовать сеть, будут установлены специальные драйверы, которые позволяют плате сетевого интерфейса компьютера-клиента связаться с сетью. Эти драйверы работают подобно драйверам принтера, позволяющим прикладным программам посыпать информацию на принтер. Программное обеспечение сетевого драйвера дает возможность программам посыпать и принимать информацию по сети. Каждый компьютер в сети может содержать одну или более плат сетевого интерфейса, которые соединяют компьютер с сетью.

Очевидно, что производительность ЛВС зависит от компьютера, используемого в качестве сервера. При использовании Windows 2000 Server необходимо ориентироваться на наиболее высокоскоростной компьютер. В этом случае, как всегда, существует возможность выбора между готовыми серверами, предлагаемыми производителями и поставщиками компьютерной техники, и серверами самостоятельной сборки. При наличии определенного опыта, самостоятельно собранный под заказ сервер может составить альтернативу готовому продукту. Поэтому следует обратить внимание на следующие моменты:

1. На вопрос об используемой шине ответ однозначен – PCI. Помимо высокой производительности (за счет 64-битной разрядности шины), PCI – компоненты допускают программное конфигурирование. Благодаря последнему обстоятельству, возможные конфликты между подключаемыми аппаратными ресурсами почти всегда предотвращаются автоматически.

2. Windows 2000 Server изначально предъявляет высокие требования к объему оперативной памяти. Поэтому с учетом того, что стоимость оперативной памяти на сегодняшний день не столь велика минимальный

объем ОЗУ не целесообразно делать менее 512 Мб (как с точки зрения цены так и с точки зрения производительности).

В серверах рекомендуется использовать винчестеры Fast SCSI и соответствующий адаптер SCSI (но мы будем использовать обычные винчестеры IDE, т.к. использование SCSI значительно увеличит стоимость сервера). При использовании Fast SCSI скорость передачи данных достигает 10 Мбит/с. Новейшие жесткие диски с интерфейсом Ultra SCSI обладают скоростью передачи до 20 Мбит/с. Если же винчестер должен работать еще быстрее, необходимо установить более дорогой Ultra Wide SCSI диск и соответствующий контроллер. Скорость Ultra Wide SCSI достигает 40 Мбит/с, и он представляет собой идеальное устройство для высокопроизводительного сервера, в том числе и для сетей с интенсивным обменом данными.

Маленький корпус для такого компьютера противопоказан, так как это может привести к перегреву, особенно при использовании высокопроизводительного процессора и нескольких жестких дисков. Идеальным корпусом будет корпус типа Big Tower, кроме всего прочего, обеспечивающий возможность дальнейшего расширения системы. Еще более удобны специальные корпуса для серверов, снабженные мощными блоками питания, дополнительными вентиляторами, съемными заглушками и защитной передней панелью.

Если сервер будет оснащен двумя или более жесткими дисками, необходимо подумать о его дополнительном охлаждении. Для этого устанавливают специальные вентиляторы, которые можно дополнительно установить в системный блок.

Скоростной привод CD-ROM (или CD-RW) сэкономит время при установке ОС и прикладного ПО.

Так как все подключенные к сети рабочие станции будут постоянно обращаться к серверу, одним из его важнейших компонентов является

производительная 32-битная сетевая карта. Она должна эффективно управлять информационным обменом, то есть иметь сопроцессор, принимающий на себя основные функции центрального процессора по обработке поступающих на сервер данных.

Таким образом, была разработана топология ЛВС для небольшого предприятия, обосновано применение конкретной ОС сервера.

СПИСОК ЛИТЕРАТУРЫ

1. Администрирование сети на основе Microsoft Windows 2000. Учебный курс, издательство “Русская редакция”, 2000 г.
2. **Вегешна, Шринивас.** Качество обслуживания в сетях IP.: Пер. с англ. - М.: Издательский дом "Вильямс", 2003. - 368 с.
3. Вычислительные системы, сети и телекоммуникации / В.Л. Бродо - СПб.: Питер, 2002. - 688 с.
4. **К. Закер.** Компьютерные сети. Модернизация и поиск неисправностей. Пер. с англ. - СПб.: БХВ - Петербург, 2002. -1008 с.
5. Компьютерные сети. Принципы, технологии, протоколы / В.Г.Олифер, Н.А.Олифер. - СПб.: Питер, 2001. - 672 с.
6. **Новиков Ю.В., Кондратенко С.В.** - Локальные сети: архитектура, алгоритмы, проектирование. М.: Издательство ЭКОМЮ 2000. - 312 с.
7. **Остерлох, Хизер.** Маршрутизация в IP- сетях. Принципы, протоколы, настройки: Пер. с англ./ Хизер Остерлох - СПб.: ООО "ДиаСофтЮП", 2002. - 512 с.
8. **Пакет Кэтрин, Тир Дайана.** Создание масштабируемых сетей CISCO.: Пер. с англ. - М.: Издательский дом "Вильямс", 2002. - 792 с.
9. **Пескова С.А., Гуров А.И., Кузин А.В.** Центральные и периферийные устройства электронных вычислительных средств/Под ред. О.П.Глудкина. - М.: Радио и связь, 2000. - 496 с.
10. **Ретана Альваро, Сиатс Дон, Уайт, Расс.** Принципы проектирования корпоративных IP сетей.: Издательский дом "Вильямс", 2002. - 368 с.
11. **Суворова Е.А., Шейнис Ю.Е.** Проектирование цифровых систем на VHDL. - СПб.: БХВ - Петербург, 2003. - 576 с.
12. **Таненбаум Э.** Коипьютерные сети.- СПб.: Ппитер, 2002. - 848 с.: ил.

13. Шиндер, Дебра, Литтлджон. Основы компьютерных сетей: Пер. с англ. - М.: Издательский дом "Вильямс", 2002. - 656 с.

Рисунки к главе 1

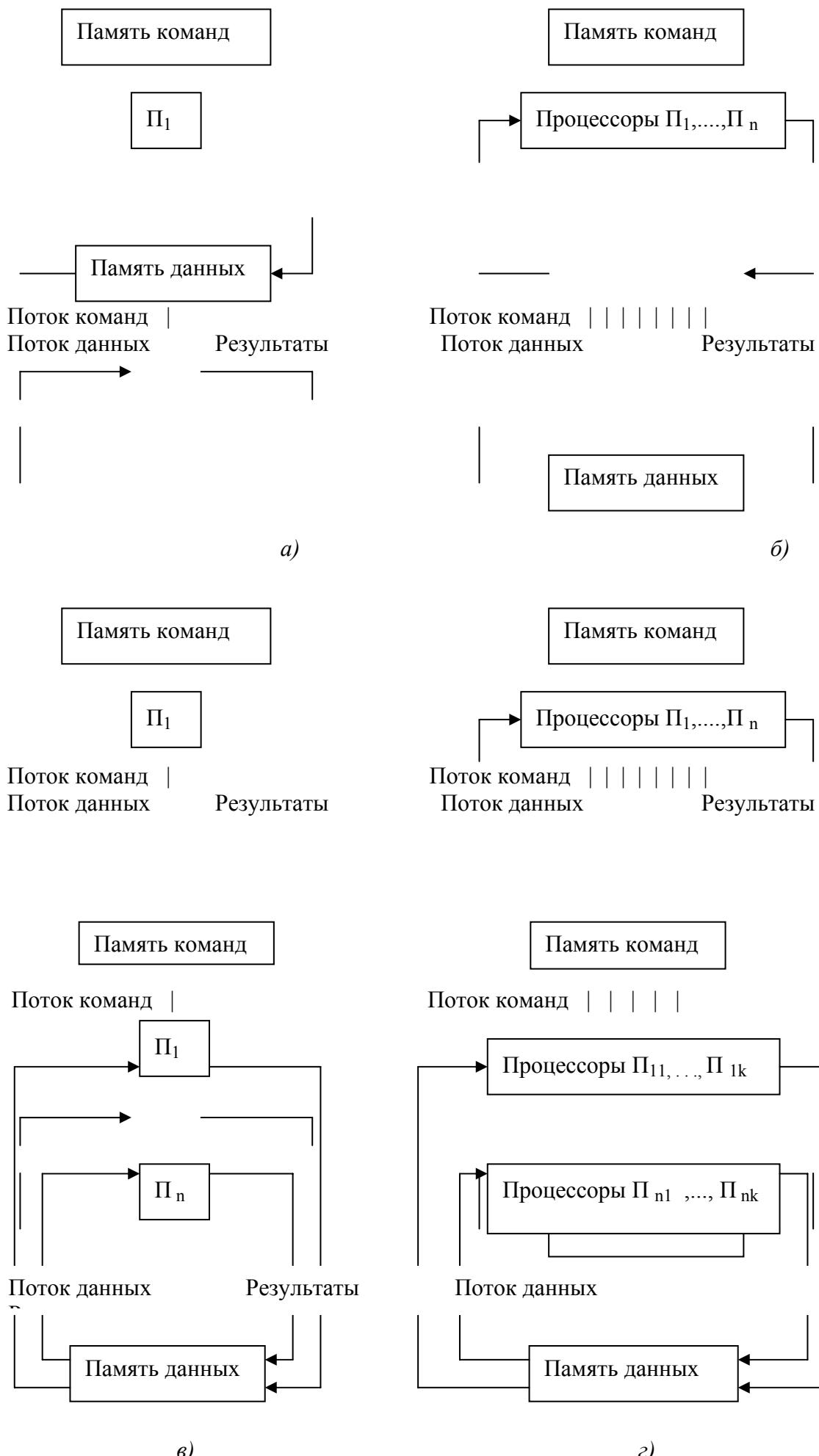




Рис. 1.1. Структуры многопроцессорных ВС: *а*) - структура ОКОД (SISD) - однопроцессорная суперЭВМ; *б*) - структура МКОД (MISD) - конвейерная (магистральная) суперЭВМ; *в*) - структура ОКМД (SIMD) - векторная суперЭВМ; *г*) - структура МКМД (MIMD) - матричная суперЭВМ

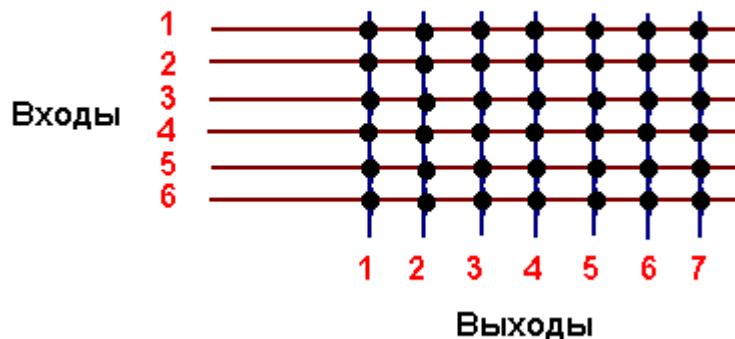


Рис. 1.2. Пространственный коммутатор

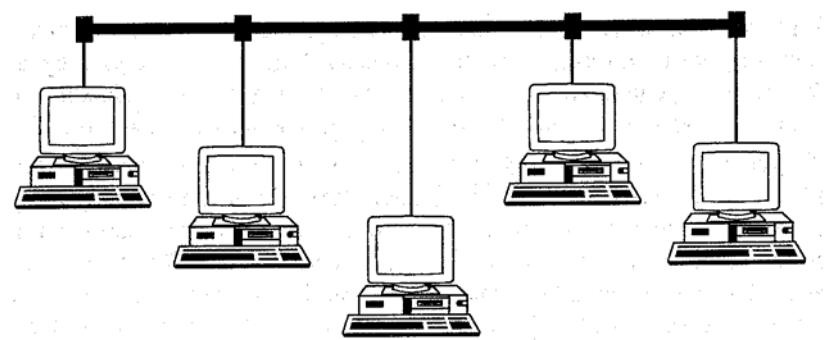


Рис.1.3. Топология "шина"

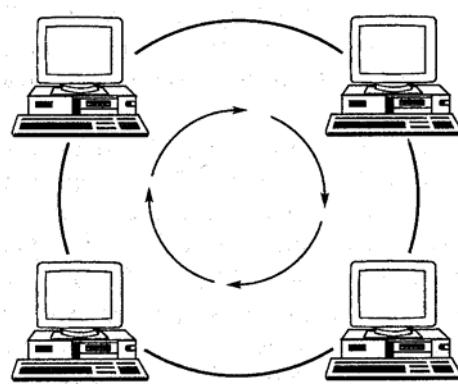


Рис. 1.4. Топология "кольцо"

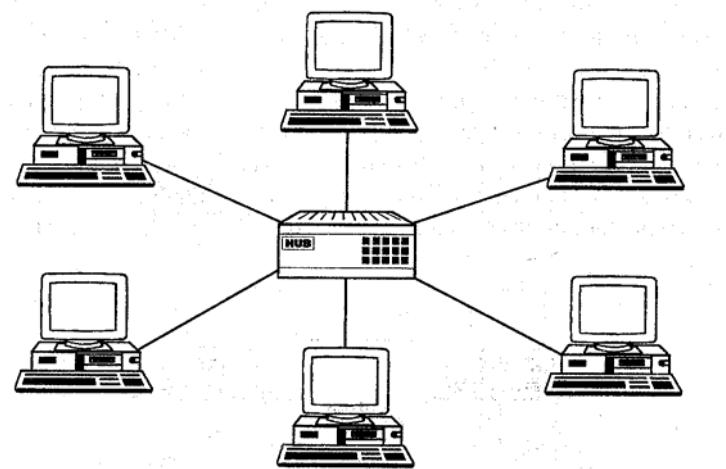


Рис. 1.5. Топология "звезда"

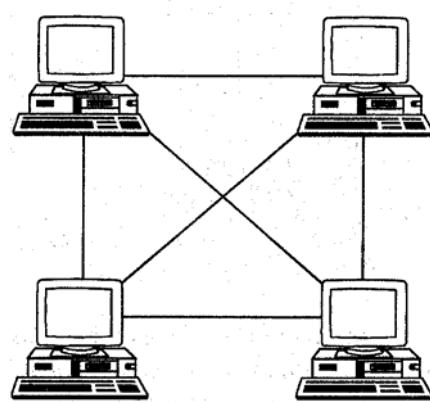


Рис. 1.6. Ячеистая топология

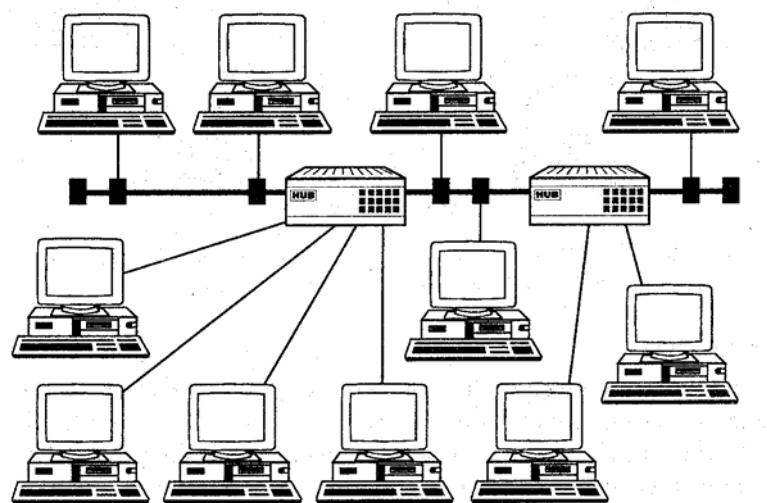


Рис. 1.7. Топология "звезда на шине"

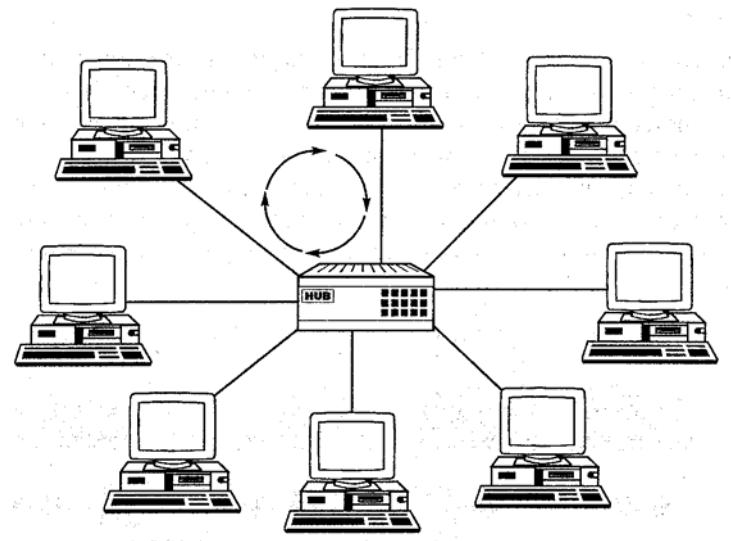


Рис. 1.8. Топология "звезда на кольце"

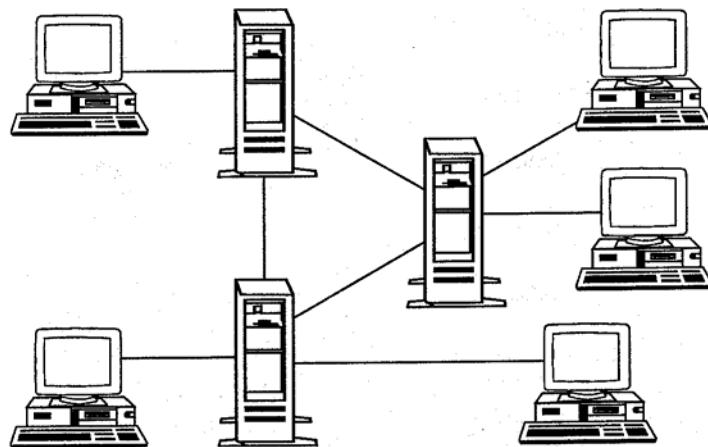


Рис. 1.9. Гибридная ячеистая топология

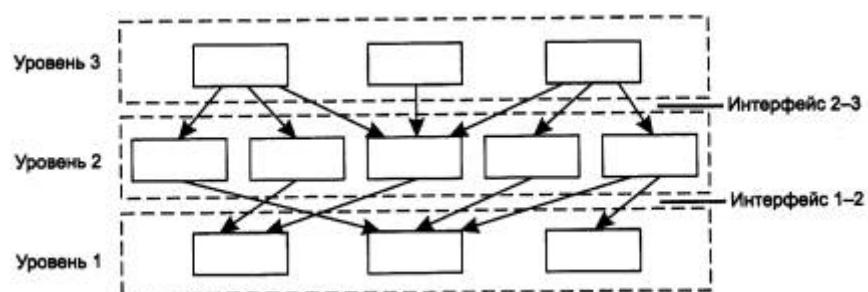


Рис. 1.10. Многоуровневый подход к созданию ИВС

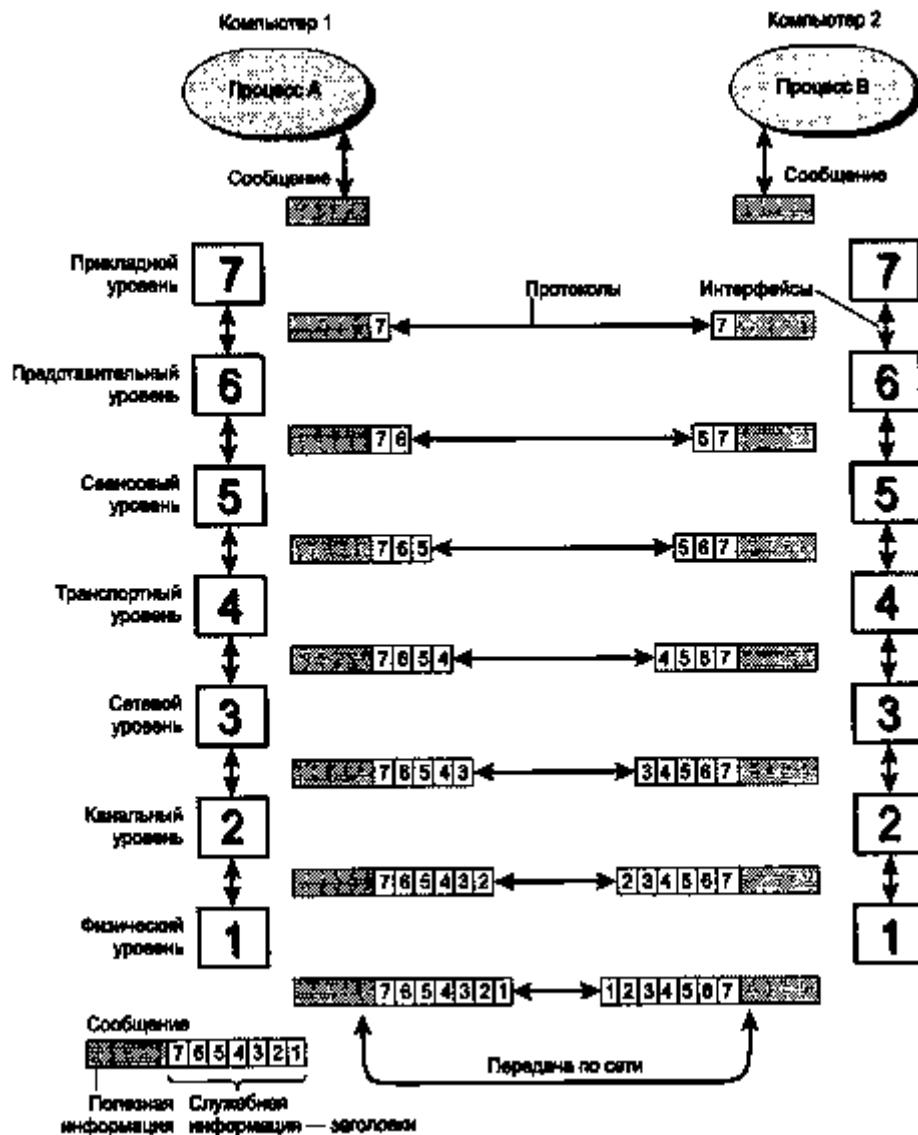


Рис. 1.11. Модель взаимодействия открытых систем ISO/OSI

Рисунки к главе 3



Рис. 3.1. Канал связи



Рис. 3.2. Амплитудно-частотная характеристика

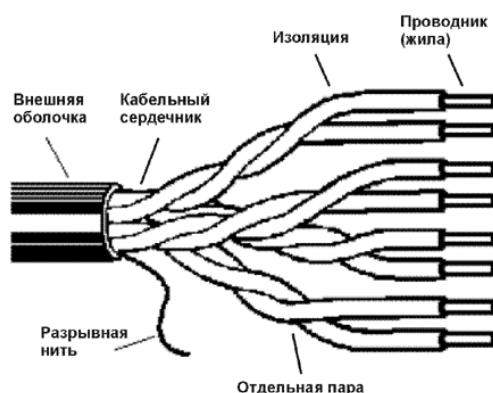


Рис. 3.3. Кабель Витая пара

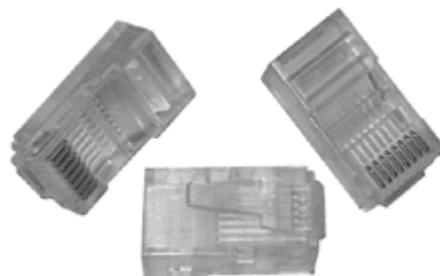


Рис. 3.4. Разъемы RJ-45

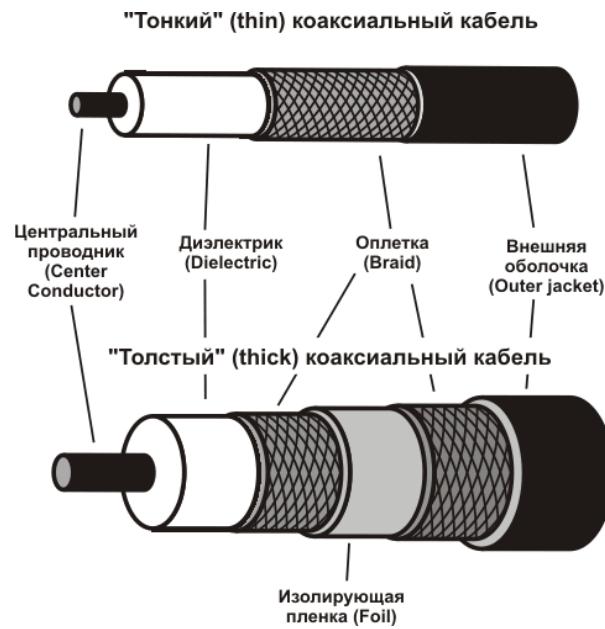


Рис. 3.5. Коаксиальные кабели

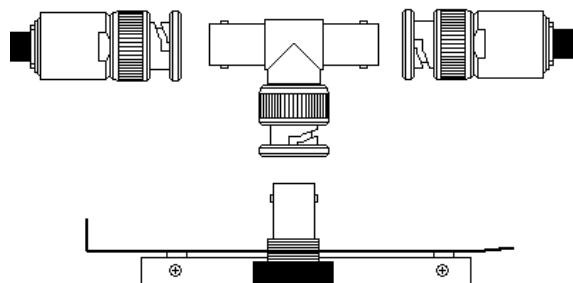


Рис. 3.6. Разъемы BNC

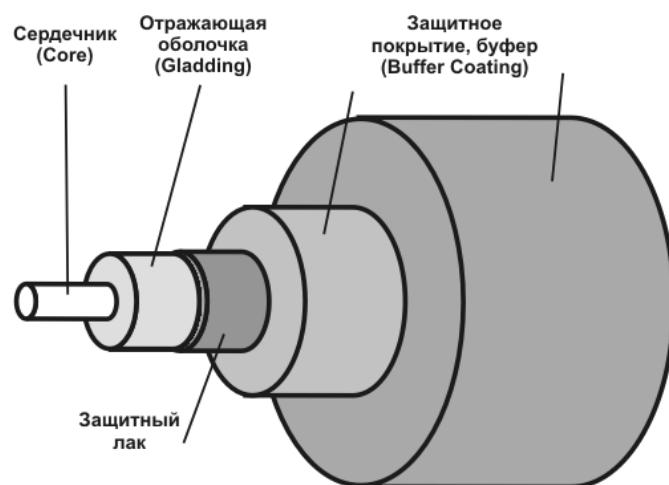


Рис. 3.7. Конструкция оптоволоконного кабеля

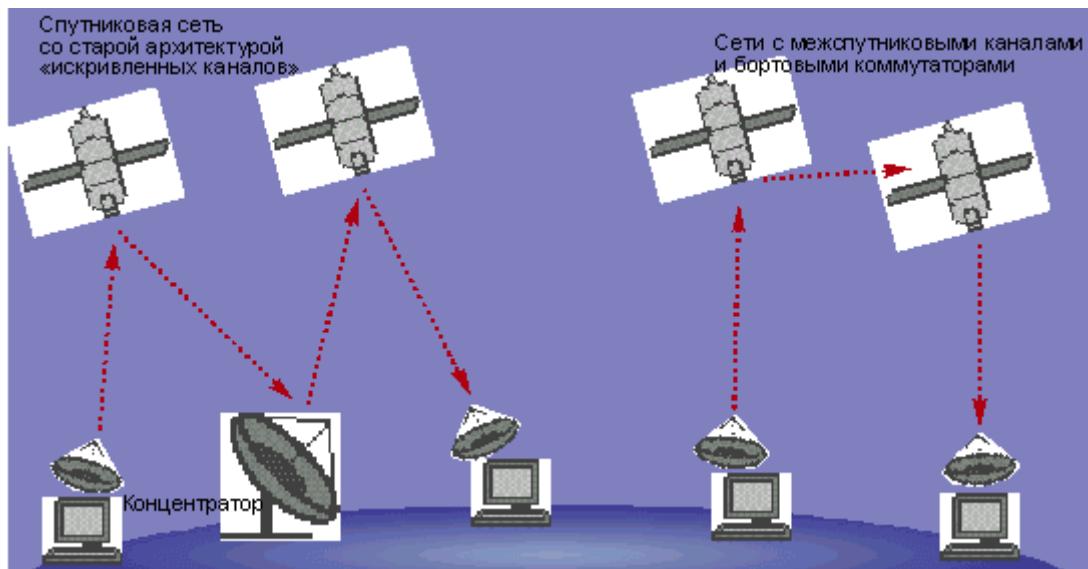


Рис. 3.8. Схема спутниковой связи

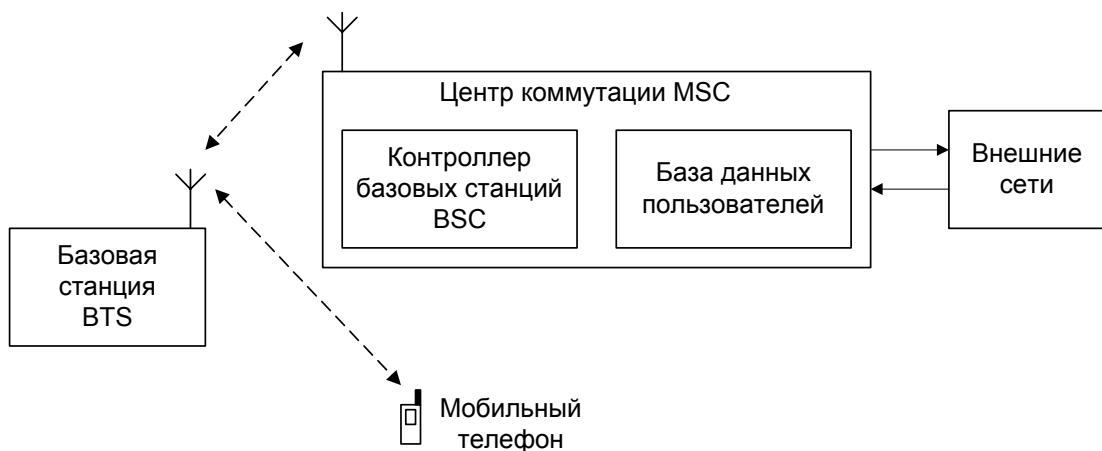


Рис. 3.9. Схема сотовой телефонной связи

Рисунки к главе 4

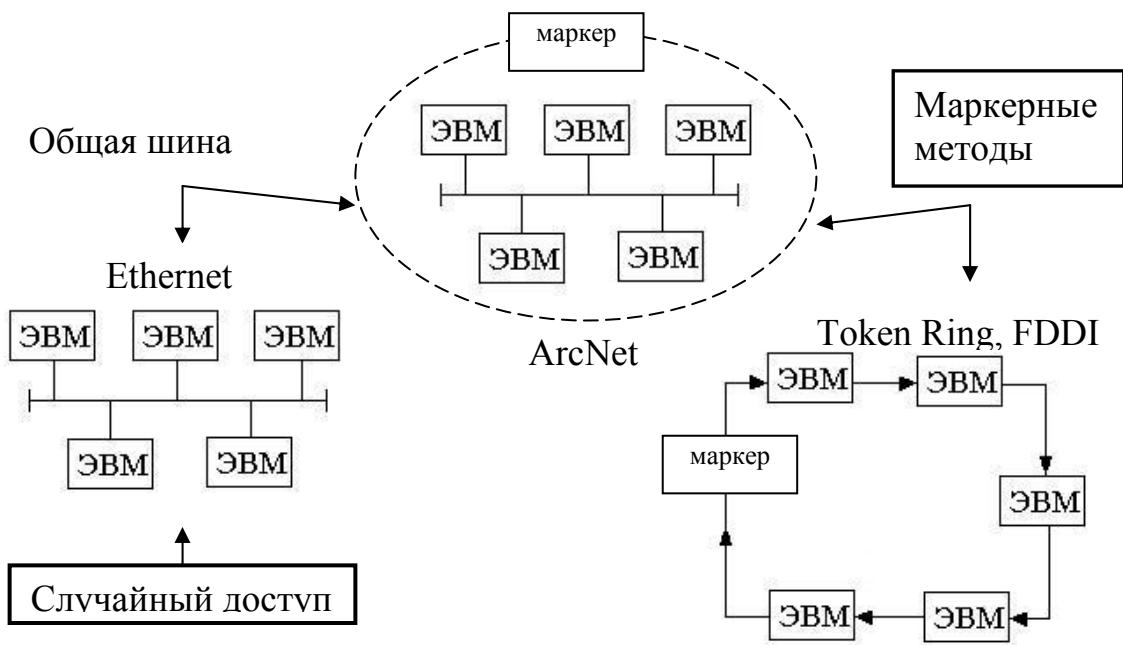


Рис. 4.1. Методы доступа к локальной сети



Рис. 4.2. Формат маркера

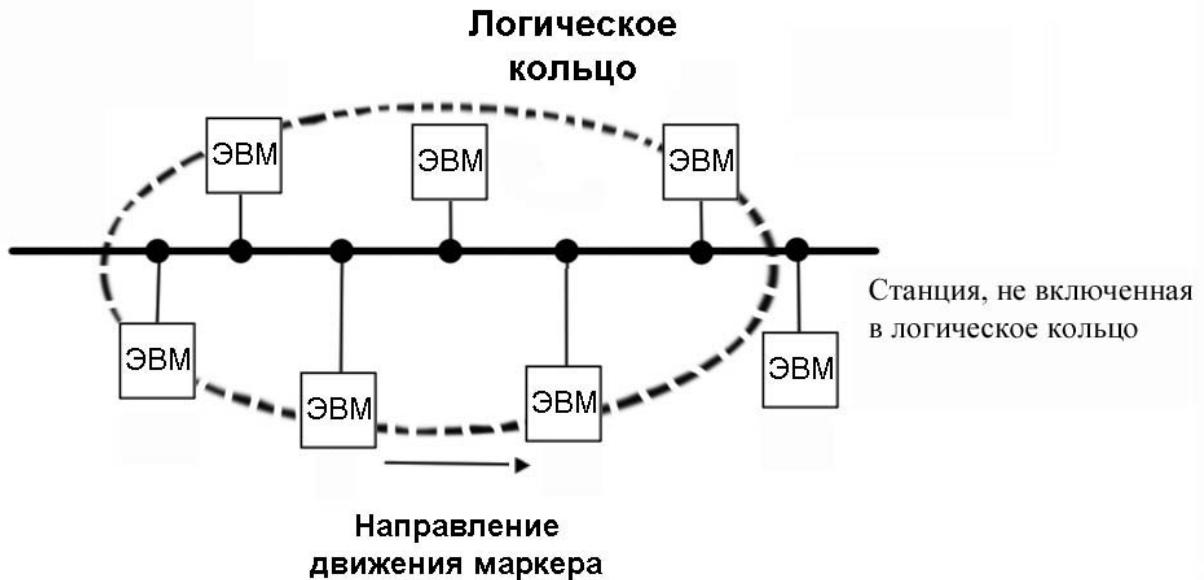


Рис. 4.3. Структура сети на основе маркерной шины

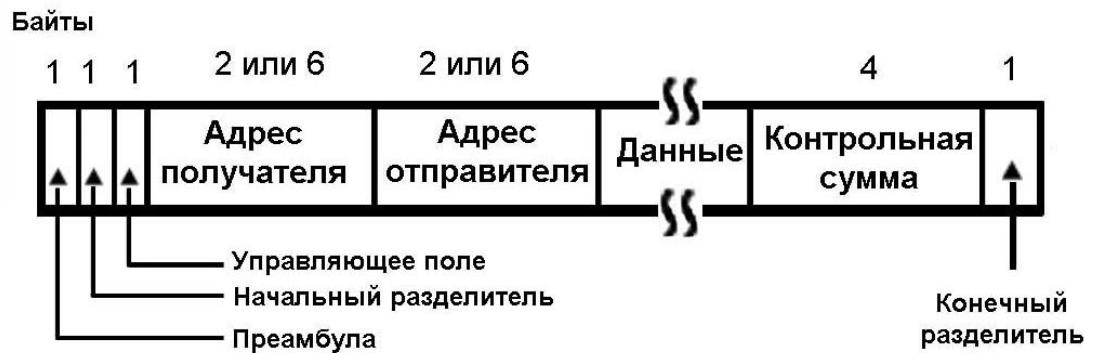


Рис. 4.4. Передаваемый кадр при организации сети маркерной шиной

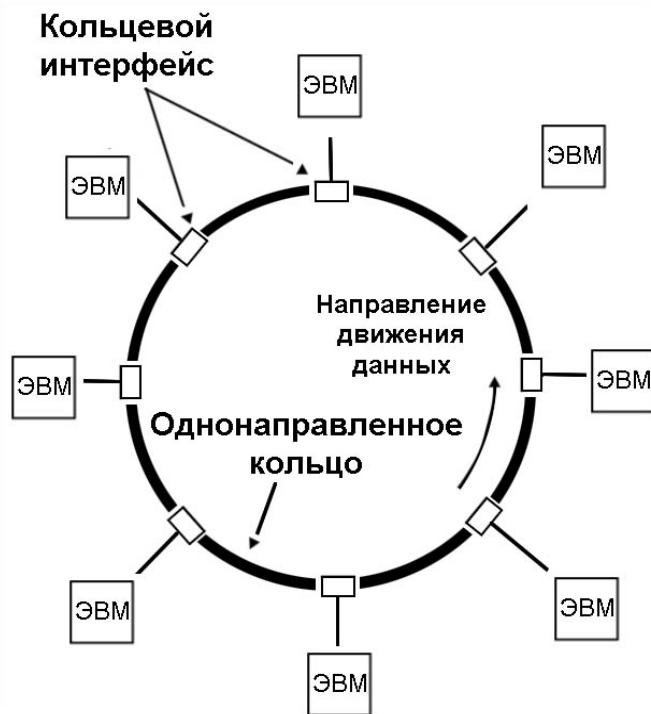


Рис 4.5. Структура сети на основе маркерного кольца

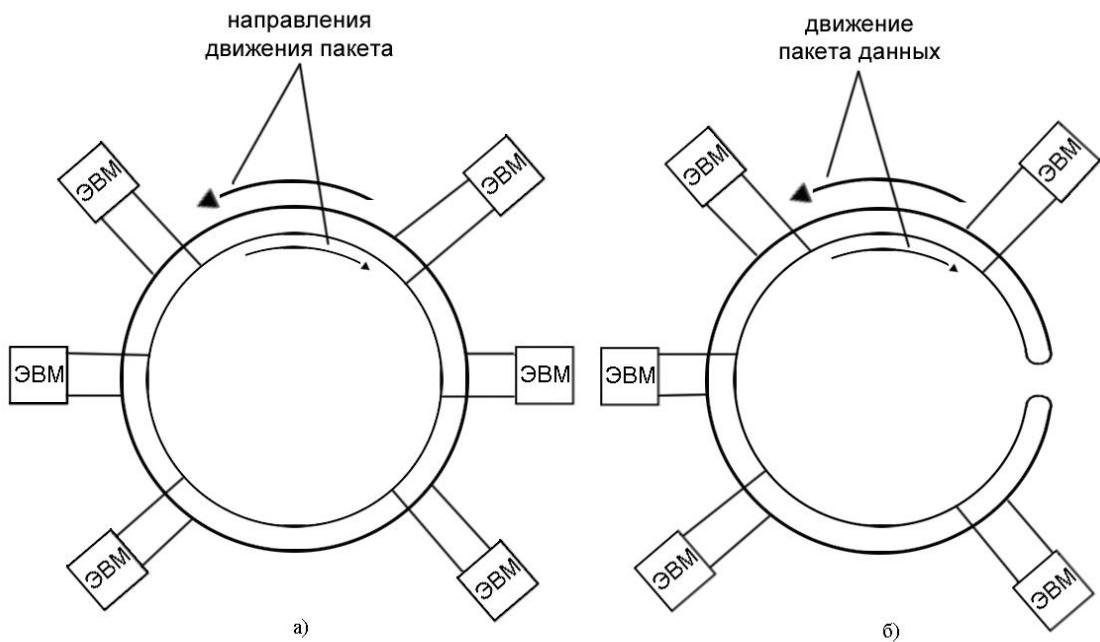


Рис. 4.7. Пример построения структурированной кабельной системы

Рисунки к главе 5

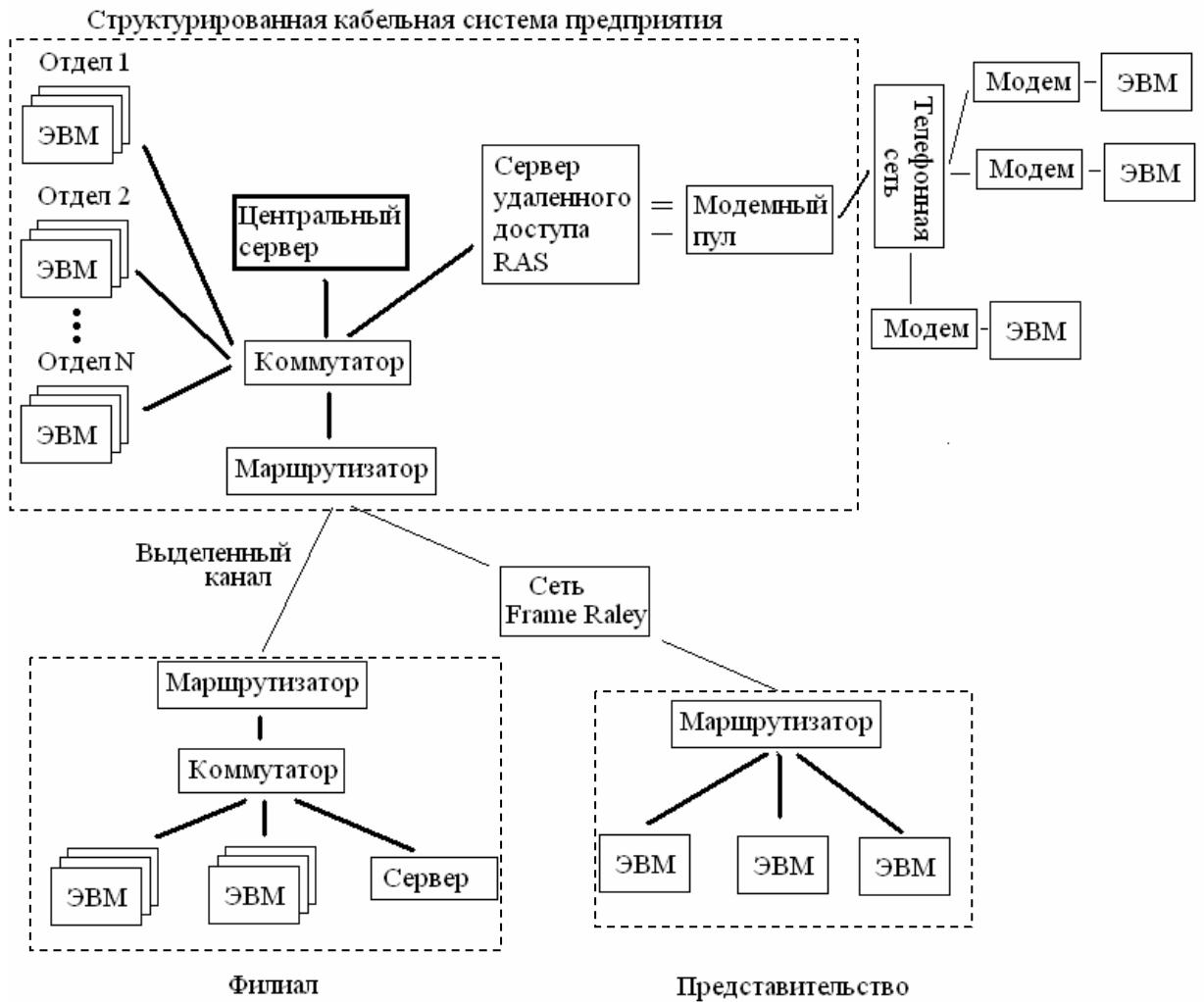


Рис. 5.1. Пример построения корпоративной сети

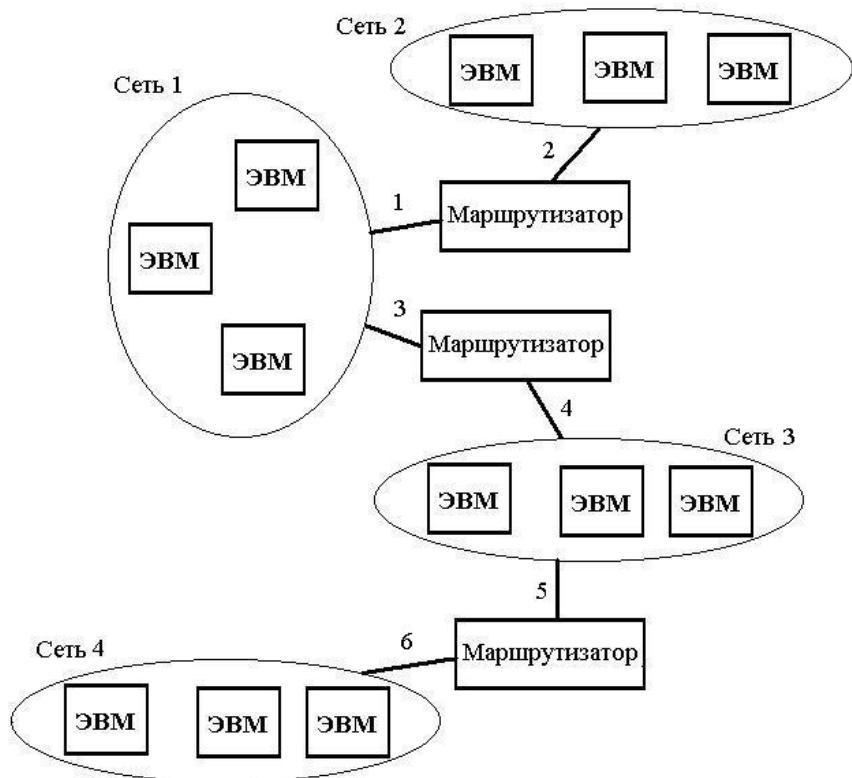


Рис. 5.2. Пример построения корпоративной

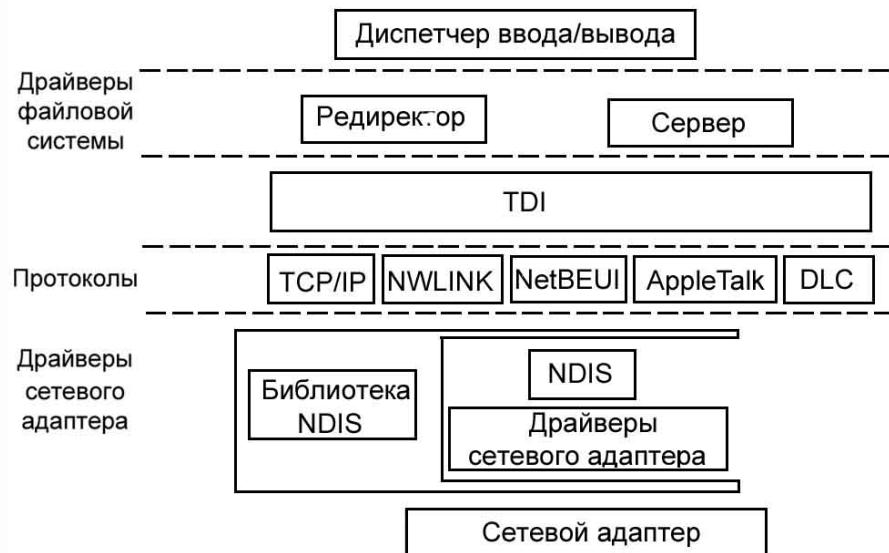


Рис. 5.3 Организация сетевых уровней

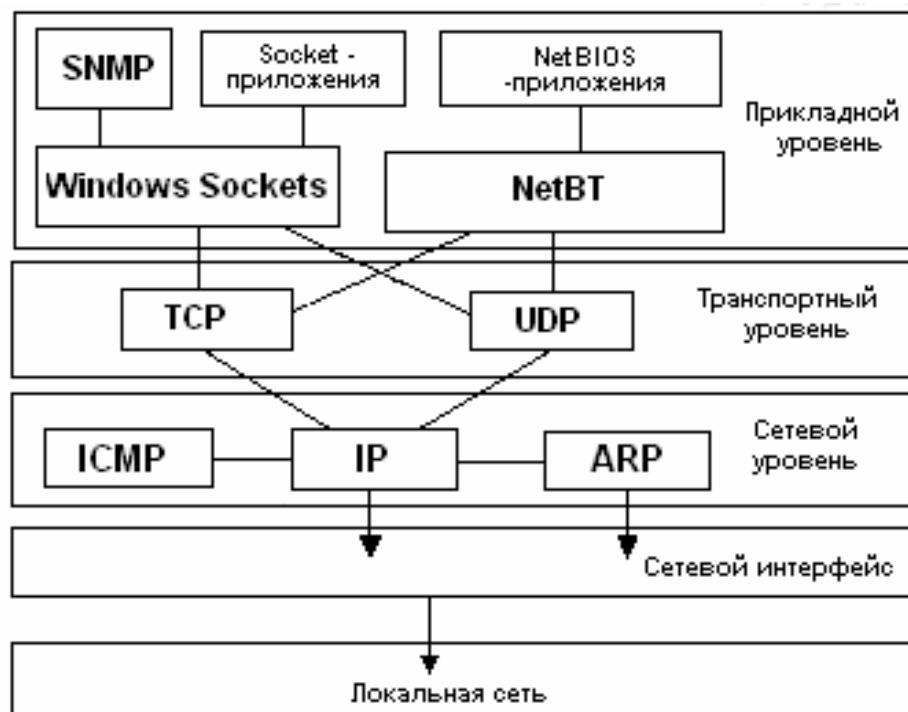


Рис. 5.4 Стек протоколов TCP/IP

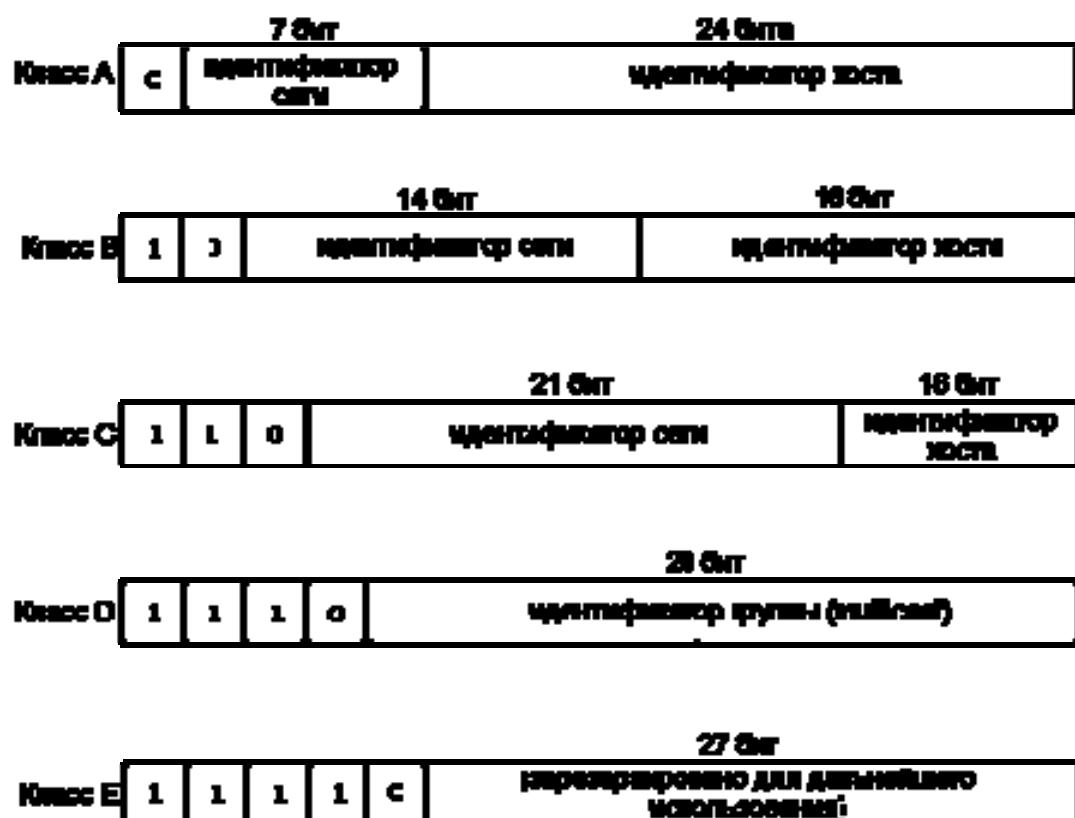


Рисунок 5.5 Классы адресов Internet.

Класс	Диапазон
A	0.0.0.0 - 127.255.255.255
B	128.0.0.0 - 191.255.255.255
C	192.0.0.0 - 223.255.255.255
D	224.0.0.0 - 239.255.255.255
E	240.0.0.0 - 247.255.255.255

Рисунок 5.6. Диапазоны IP адресов в разных классах сетей.

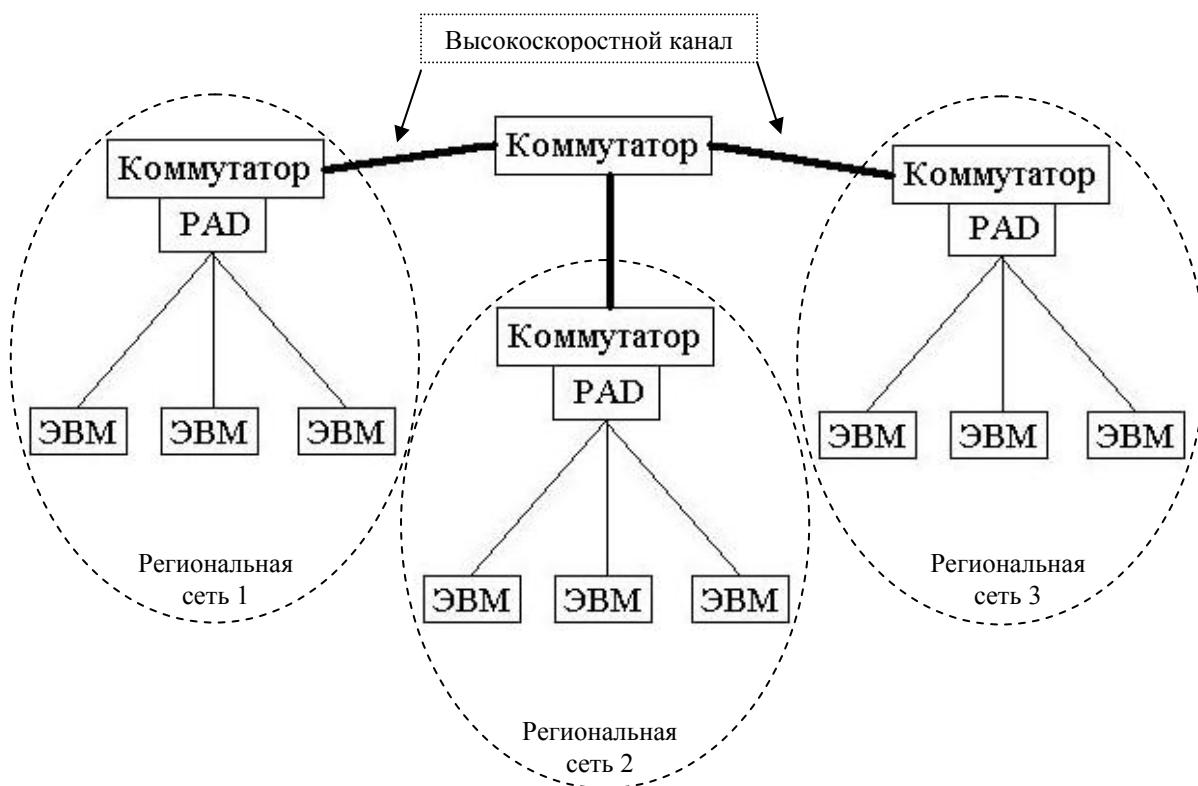


Рис. 5.7. Структура построения сети X.25

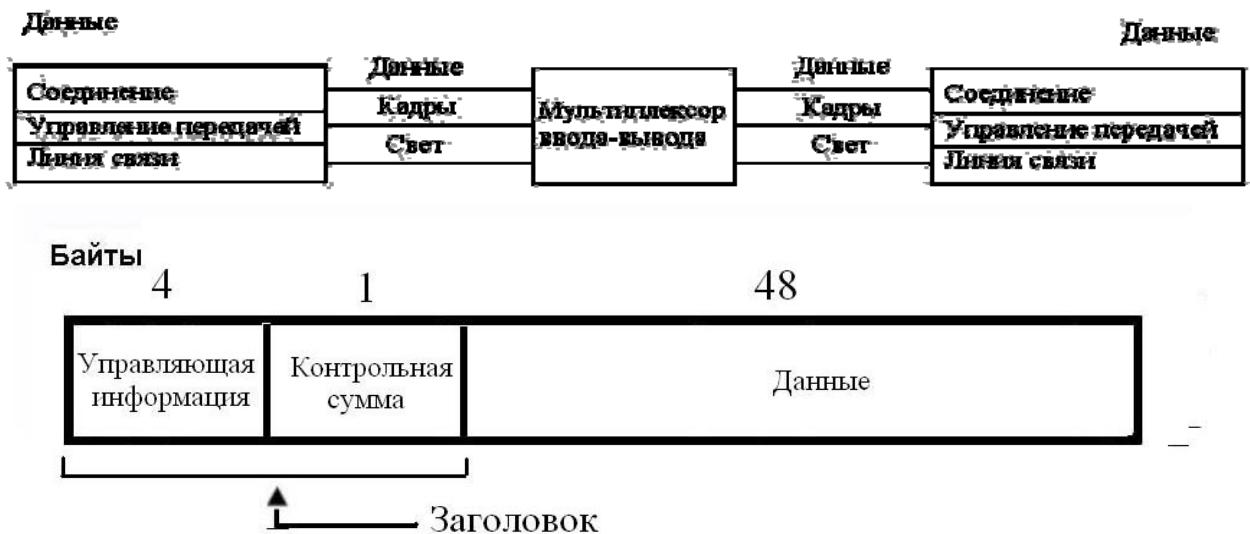


Рис. 5.8. Формат пакета данных АТМ



Рис 5.9. Организация сети SDH

Рисунки к главе 6



Рис. 6.1. Классификация операционных систем

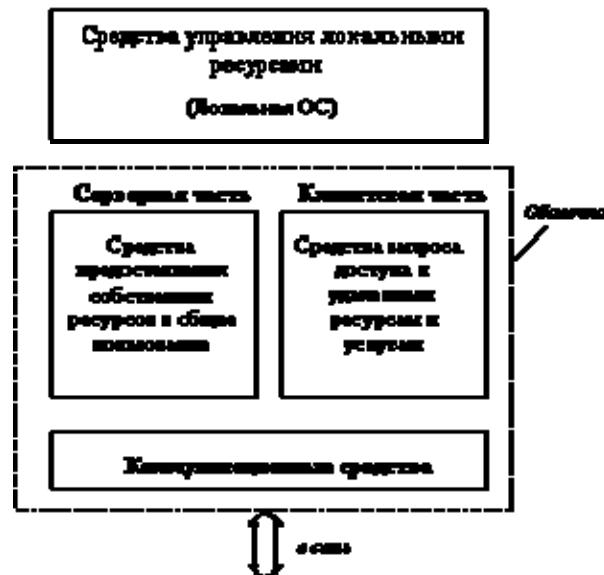


Рис. 6.2 Структура сетевой ОС

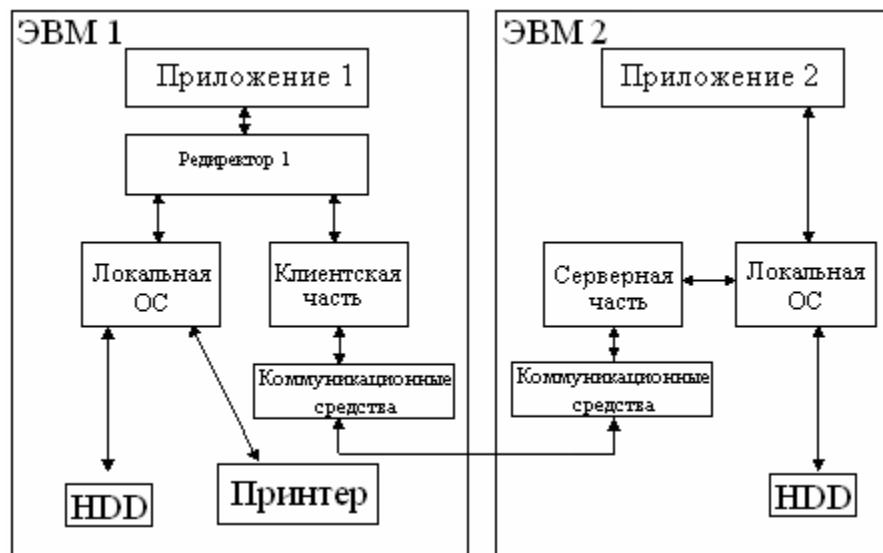


Рис. 6.3. Взаимодействие компонентов операционной системы при взаимодействии компьютеров

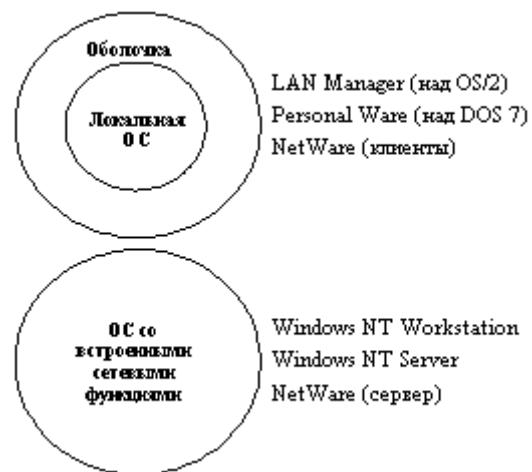


Рис. 6.4. Варианты построения сетевых ОС



Рис. 6.5. Структура ОС клиент-сервер

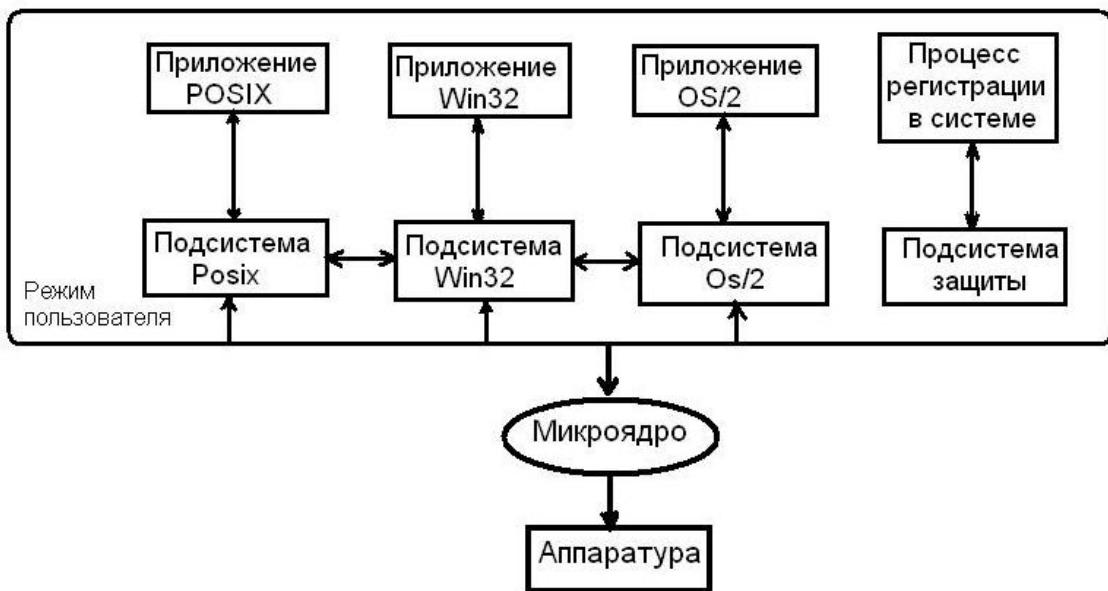


Рис. 6.6. Представление режима пользователья



Рис. 6.7. Представление режима ядра



Рис. 6.8. Топология построения системы распределенных вычислений

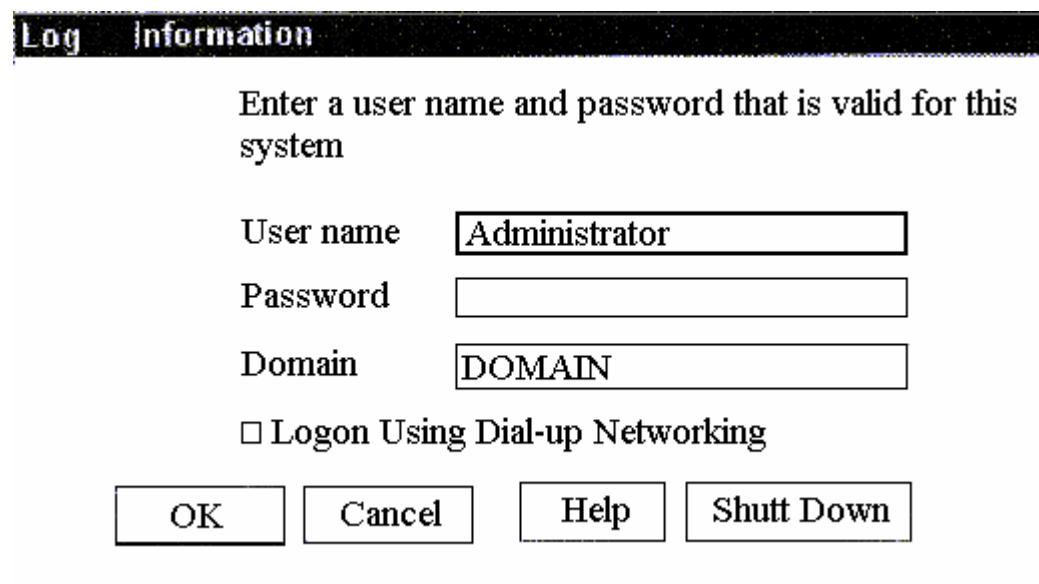


Рис. 6.9. Окно регистрации входа пользователя в систему

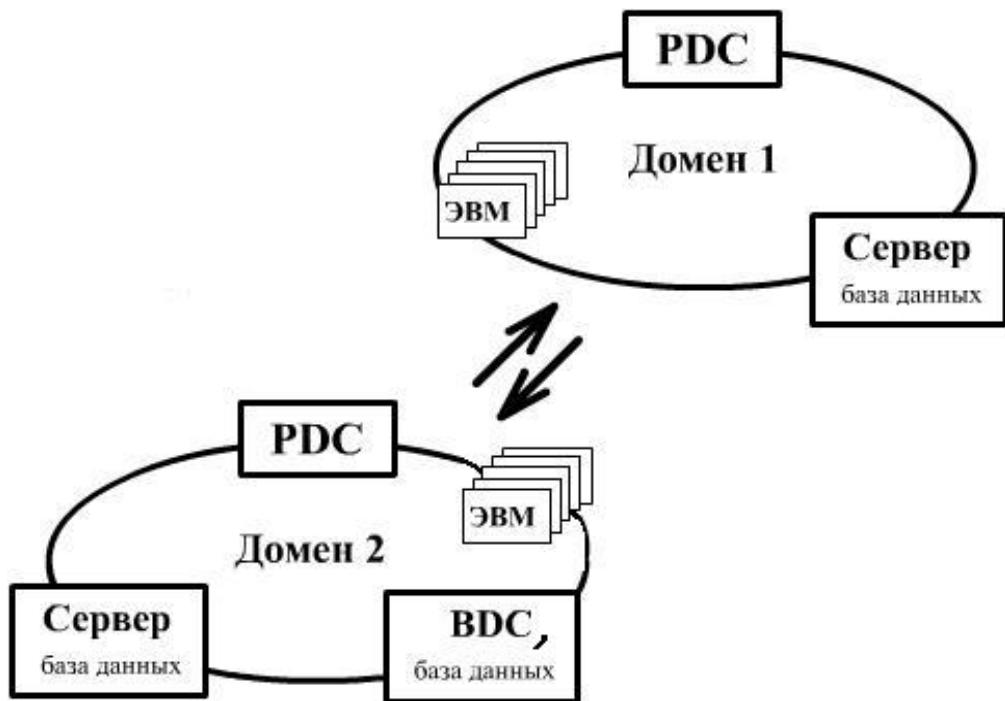


Рис. 6.10. Пример построения доменной модели сети

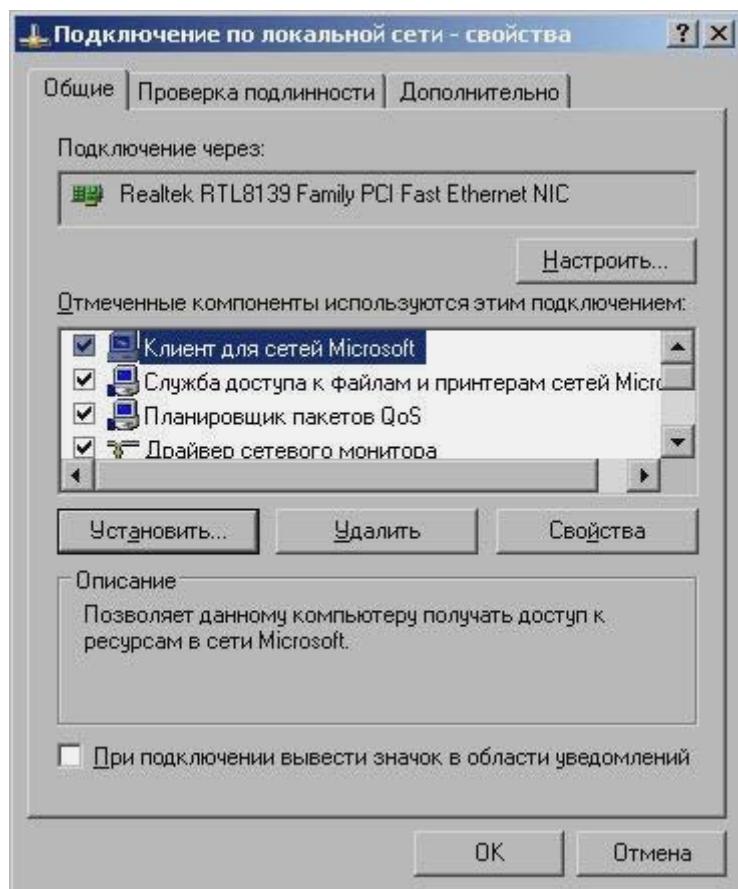


Рис. 6.11. Окно «Подключения по локальной сети - свойства»

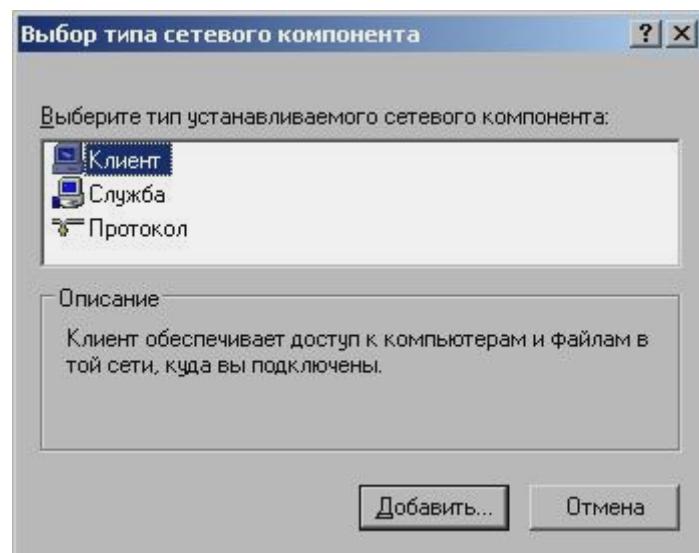


Рис. 6.12. Окно «Выбор типа сетевого компонента»

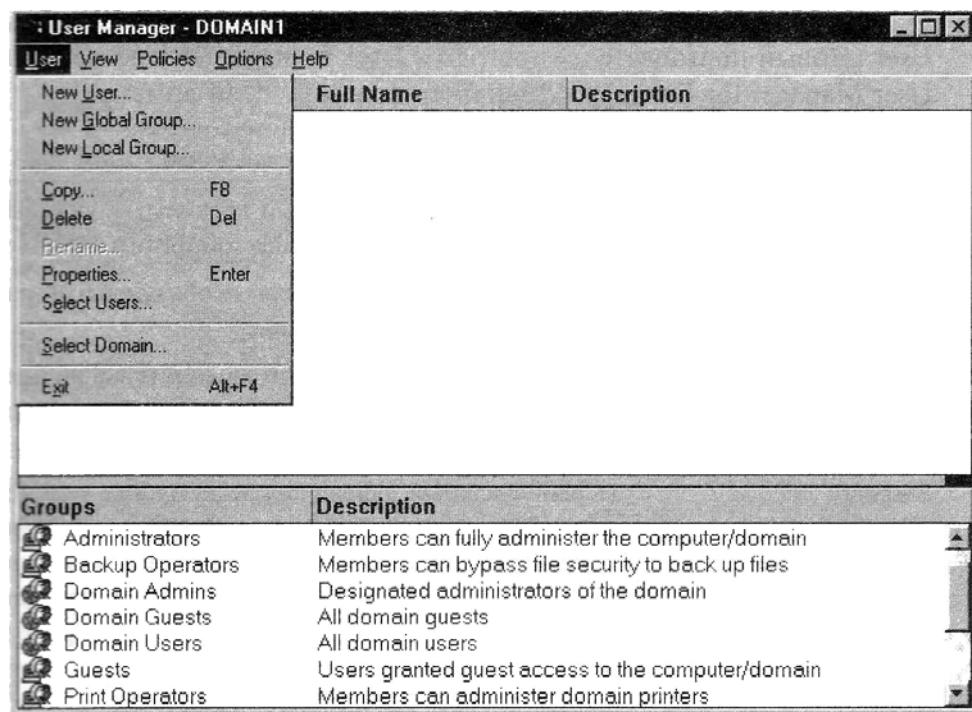


Рис. 6.13. Окно User Manager for Domain

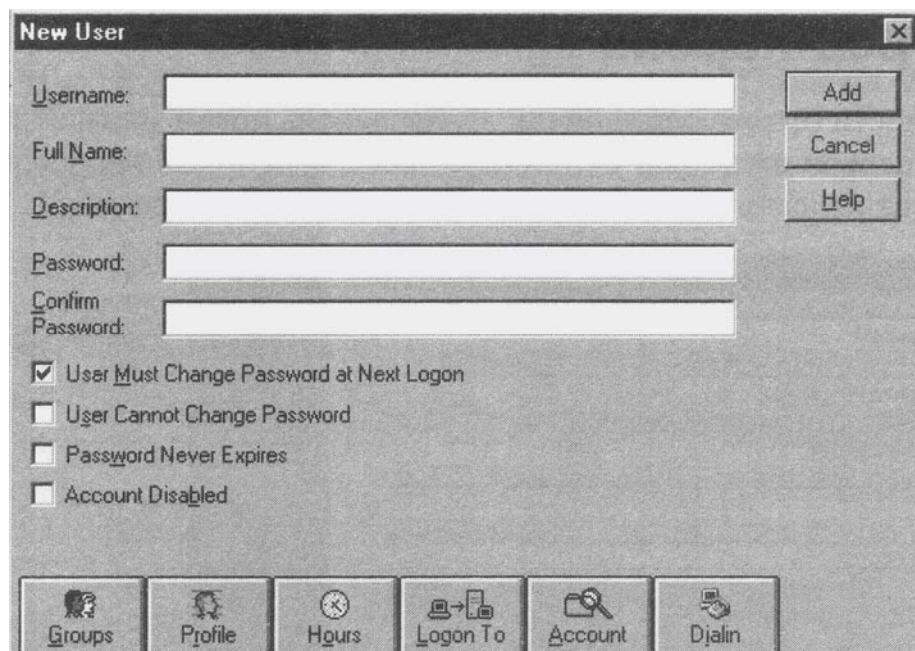


Рис. 6.14. Окно New User

Рисунки к главе 7

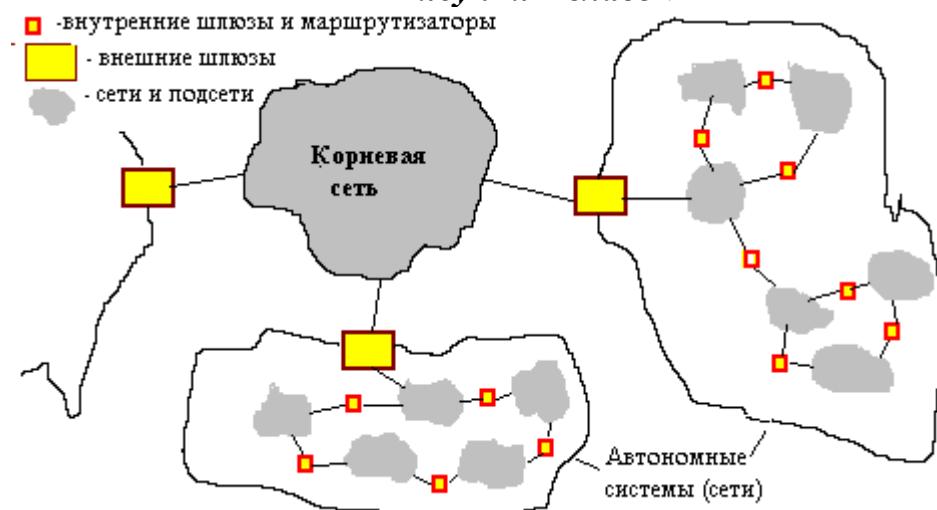


Рис. 7.1. Иерархическая структура территориальной сети

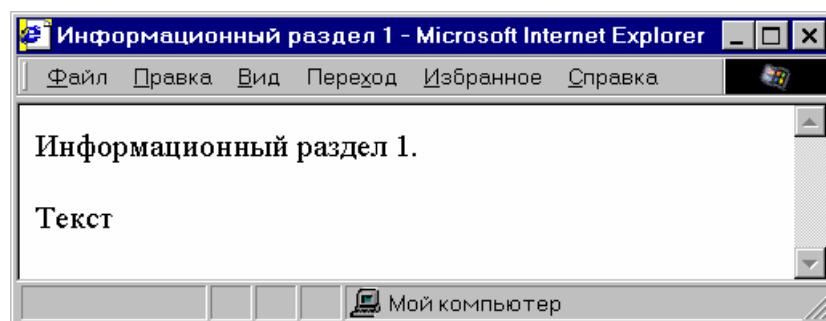


Рис. 7.2. Простейший HTML-документ

Название информационного раздела	
Оглавление	Содержание
Раздел 1	информационного
Раздел 2	раздела
:	
Раздел N	

Рис. 7.3. Организация окна сайта или компьютерного учебника в виде фреймов

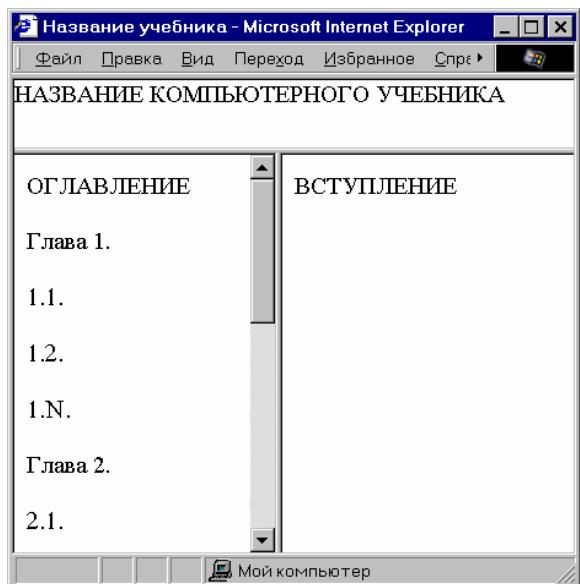


Рис. 7.4. Многофреймовое окно Учебника

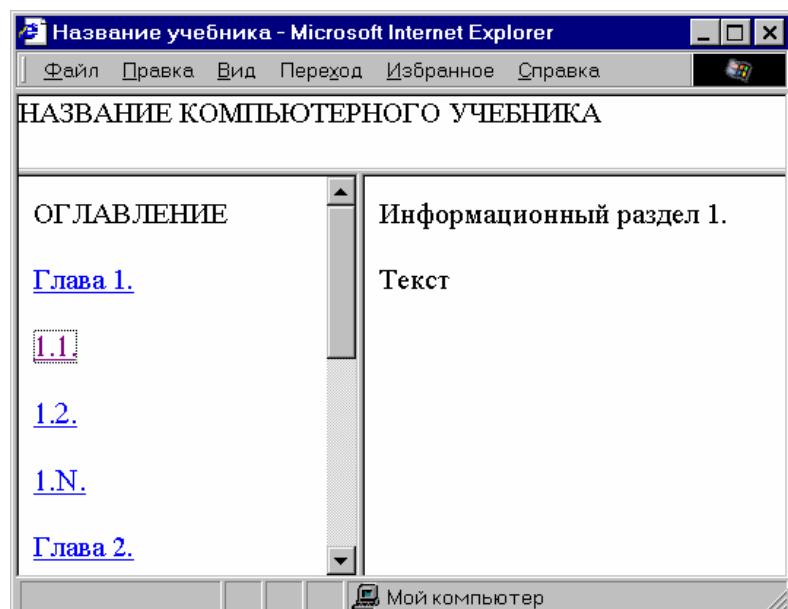


Рис. 7.5. Окно Учебника в формате HTML

Рисунки к приложениям

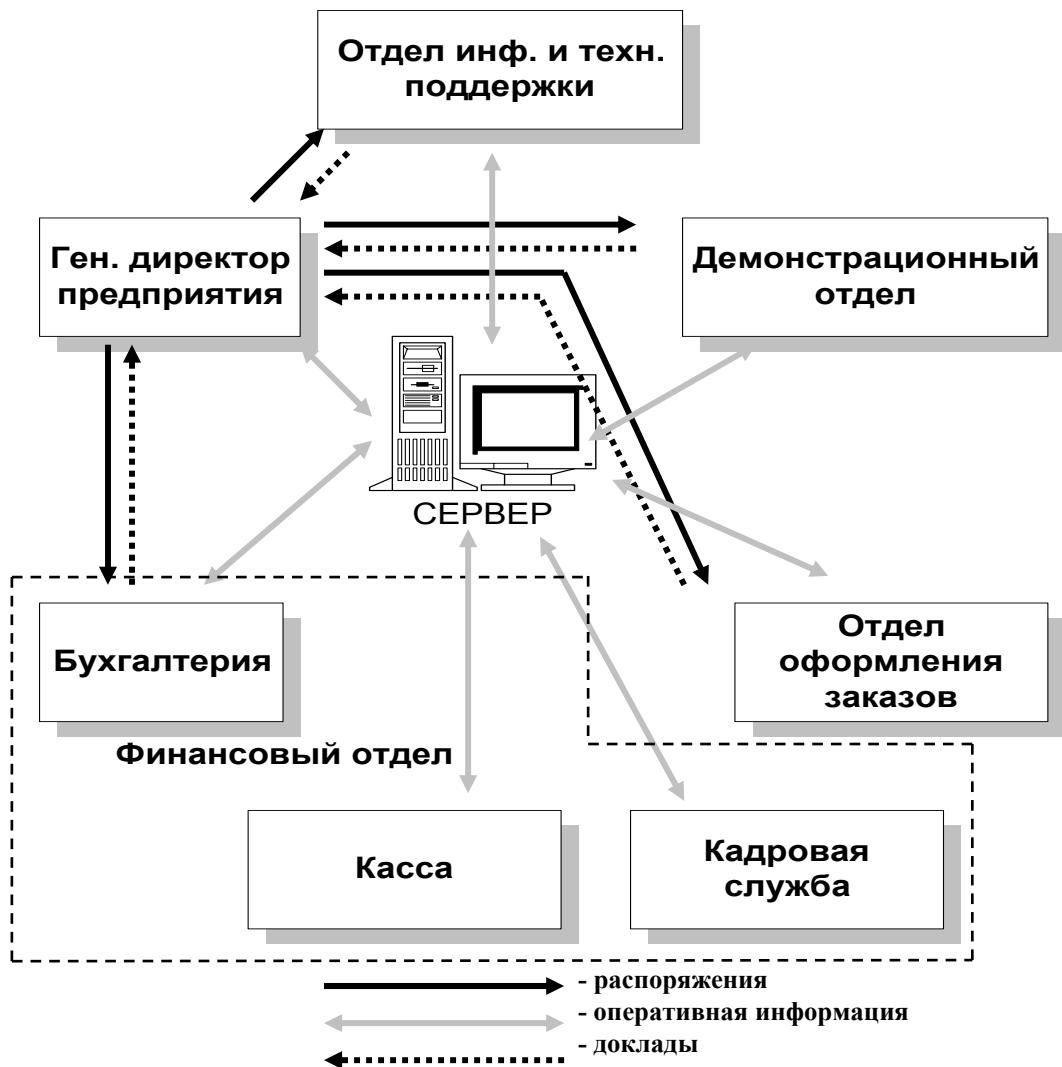


Рис.П1. Организационная структура подразделения и информационные потоки

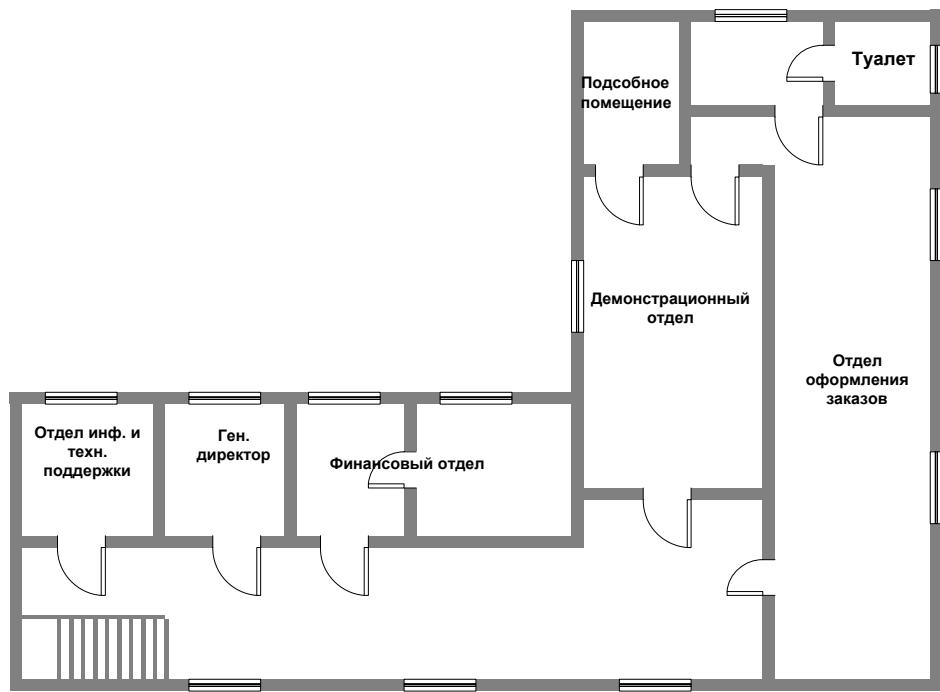


Рис. П2. План помещений предприятия

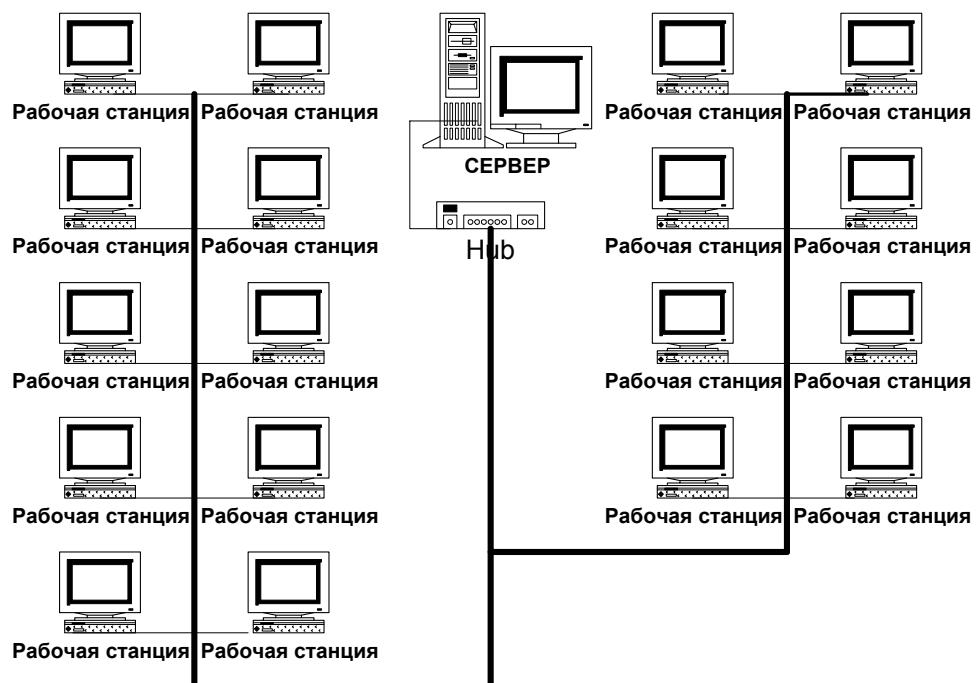


Рис. П3. Топология сети предприятия