

Тема 5. СЕТЕВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

- 5.1. Локальные сети.
- 5.2. Глобальная сеть Интернет.
- 5.3. Основы проектирования web-страниц.

5.1. Локальные сети

Компьютерная (вычислительная) сеть – это совокупность компьютеров и коммутационных устройств, соединенных с помощью каналов связи в единую систему, удовлетворяющую требованиям распределенной обработки данных.

Распределенная обработка данных – это обработка данных, выполняемая на независимых, но связанных между собой компьютерах, представляющих распределенную систему.

Компьютерные сети являются логическим результатом эволюции компьютерных и телекоммуникационных технологий. Постоянно возрастающие потребности пользователей в вычислительных ресурсах обуславливали попытки специалистов компьютерных технологий объединить в единую систему отдельные компьютеры.

Появление локальных компьютерных сетей внесло много нового в использование вычислительной техники. Появилась возможность быстрого доступа к разделяемым вычислительным ресурсам, к базе данных сразу несколькими пользователями, причем пользователь использовал на своем сетевом компьютере знакомые команды, как и при работе с отдельным компьютером. Задачу обработки этих команд и распределения задач между отдельными компьютерами взяла на себя сетевая операционная система.

В настоящее время разделение компьютерных сетей на глобальные и локальные происходит в первую очередь по признаку их территориального размещения, по механизму установления связей между компьютерами и скорости передачи данных.

К основным характеристикам сетей относятся:

– *пропускная способность* – максимальный объем данных, передаваемых сетью в единицу времени. Пропускная способность измеряется в Мбит/с;

– *время реакция сети* – время, затрачиваемое программным обеспечением и устройствами сети на подготовку к передаче информации по данному каналу. Время реакции сети измеряется в миллисекундах.

Базовые требования к компьютерным сетям:

– *открытость* – возможность включения дополнительных компьютеров, терминалов, узлов и линий связи без изменения технических и программных средств существующих компонентов;

– *надежность* – сохранение работоспособности при изменении структуры;

– *адаптивность* – допустимость изменения типов компьютеров, терминалов, линий связи, операционных систем;

– *эффективность* – обеспечение требуемого качества обслуживания пользователей при минимальных затратах;

– *безопасность* – способность сети обеспечить защиту информации от несанкционированного доступа и сохранить достоверность и целостность передаваемых данных.

Для передачи данных компьютеры используют самые разнообразные физические каналы (коаксиальный кабель, витая пара, оптоволоконный кабель), которые обычно называются *средой передачи*.

Если в сети имеется специальный компьютер, выделенный для совместного использования участниками сети, он называется *файловым сервером*.

Группы сотрудников, работающих над одним проектом в рамках локальной сети, называются *рабочими группами*. В рамках одной локальной сети могут работать несколько рабочих групп. У участников рабочих групп могут быть разные права для доступа к общим ресурсам сети. Совокупность приемов разделения и ограничения прав участников компьютерной сети называется *политикой сети*. Управление сетевыми политиками называется *администрированием сети*. Лицо, управляющее организацией работы участников локальной компьютерной сети, называется *системным администратором*.

По *территориальной распространенности* компьютерные сети подразделяются:

– на локальные сети (LAN, Local Area Network) в пределах предприятия, учреждения, одной организации;

– региональные сети (MAN, Metropolitan Area Network) в пределах города или области;

– глобальные сети (WAN, Wide Area Network) позволяют организовать взаимодействие между компьютерами на больших расстояниях, в пределах территории государства или группы государств.

По *скорости передачи информации* компьютерные сети подразделяются:

- на низкоскоростные сети – до 10 Мбит/с;
- среднескоростные сети – до 100 Мбит/с;
- высокоскоростные сети – свыше 100 Мбит/с.

По *типу среды передачи* компьютерные сети разделяются:

- на проводные (на коаксиальном кабеле, витой паре, оптоволоконные);
- беспроводные, с передачей информации по радиоканалам или в инфракрасном диапазоне.

По *способу организации взаимодействия компьютеров* сети делят на *одноранговые* и *с выделенным сервером* (иерархические сети).

Одноранговая сеть. Все компьютеры равноправны. Любой пользователь сети может получить доступ к данным, хранящимся на любом компьютере.

Достоинство – простота установки и эксплуатации.

Недостаток – затруднено решение вопросов защиты информации.

Такой способ организации используется для сетей с небольшим количеством компьютеров и там, где вопрос защиты данных не является принципиальным.

Иерархическая сеть. При установке заранее выделяются один или несколько серверов – компьютеров, управляющих обменом данных и распределением ресурсов сети. Сервер – это постоянное хранилище разделяемых ресурсов. Любой компьютер, имеющий доступ к услугам сервера, называют *клиентом сети*, или *рабочей станцией*. Сам сервер также может быть клиентом сервера более высокого уровня иерархии. Серверы обычно представляют собой высокопроизводительные компьютеры, возможно, с несколькими параллельно работающими процессорами, винчестерами большой емкости и высокоскоростной сетевой картой.

Достоинство – позволяет создать наиболее устойчивую структуру сети, более рационально распределить ресурсы и обеспечить более высокий уровень защиты данных.

Недостатки:

- необходимость дополнительной операционной системы для сервера;
- более высокая сложность установки и модернизации сети;
- необходимость выделения отдельного компьютера в качестве сервера.

По *технологии использования сервера* различают сети с архитектурой файл-сервер и архитектурой клиент-сервер.

Файл-сервер. На сервере хранится большинство программ и данных. По требованию пользователя ему пересылается необходимая программа и данные. Обработка информации выполняется на рабочей станции.

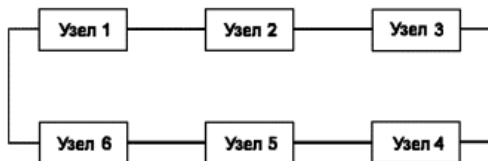
Клиент-сервер. Хранение данных и их обработка производится на сервере, который выполняет также контроль за доступом к ресурсам и данным. Рабочая станция получает только результаты запроса.

Локальная вычислительная сеть (ЛВС). Вычислительные машины, входящие в состав ЛВС, могут быть расположены самым случайным образом на территории, где создается вычислительная сеть.

Топология ЛВС – это усредненная геометрическая схема соединений узлов сети.

Топологии вычислительных сетей могут быть самыми различными, но для ЛВС сетей типичными являются три: *кольцевая, шинная, звездообразная.*

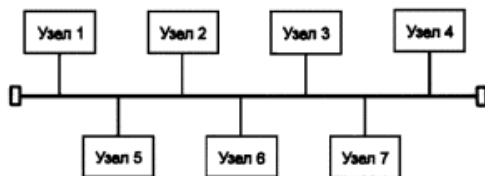
Кольцевая топология предусматривает соединение узлов сети замкнутой кривой – кабелем передающей среды. Выход одного узла сети соединяется со входом другого. Информация по кольцу передается от узла к узлу. Каждый промежуточный узел между передатчиком и приемником ретранслирует посланное сообщение. Принимающий узел распознает и получает только адресованные ему сообщения.



Кольцевая топология является идеальной для сетей, занимающих сравнительно небольшое пространство. В ней отсутствует центральный узел, что повышает надежность сети. В качестве *передающей среды используются любые типы кабелей.* Но последовательная дисциплина обслуживания узлов такой сети снижает ее быстродействие, а выход из строя одного из узлов нарушает целостность.

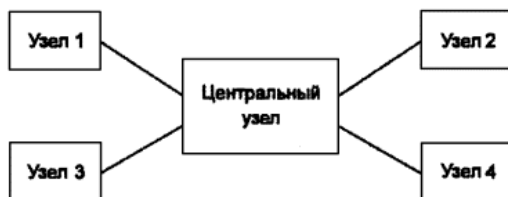
Шинная топология – одна из наиболее простых. Она связана с использованием в качестве *передающей среды коаксиального кабеля.* Данные от передающего узла сети распространяются по шине в обе стороны. Промежуточные узлы не транслируют поступающих сообщений. Информация поступает на все узлы, но принимает сообщение

только тот, которому оно адресовано. Дисциплина обслуживания параллельная.



Высокое быстродействие ЛВС. Сеть легко наращивать и конфигурировать, а также адаптировать к различным системам. Сеть устойчива к возможным неисправностям отдельных узлов, но имеет малую протяженность и не позволяет использовать различные типы кабеля в пределах одной сети. На концах сети устанавливают специальные устройства – *терминаторы*.

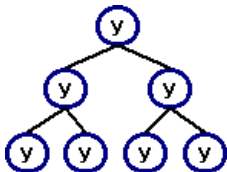
Звездообразная топология базируется на концепции центрального узла, называемого *концентратором*, к которому подключаются периферийные узлы. Каждый периферийный узел имеет свою отдельную линию связи с центральным узлом. Вся информация передается через центральный узел, который ретранслирует, переключает и маршрутизирует информационные потоки в сети.



Звездообразная топология значительно упрощает взаимодействие узлов ЛВС друг с другом, позволяет использовать более простые сетевые адаптеры. В то же время работоспособность ЛВС со звездообразной топологией целиком зависит от центрального узла.

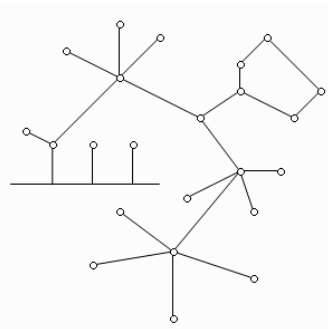
В реальных вычислительных сетях могут использоваться более сложные топологии, представляющие в некоторых случаях сочетания рассмотренных. Выбор той или иной топологии определяется областью применения сети, географическим расположением ее узлов и размерностью сети в целом. Например:

Ячеистая топология. Для нее характерна схема соединения узлов, при которой физические линии связи установлены со всеми рядом стоящими компьютерами:



В сети с ячеистой топологией непосредственно связываются только те компьютеры, между которыми происходит интенсивный обмен данными, а для обмена данными между компьютерами, несоединенными прямыми связями, используются транзитные передачи через промежуточные узлы. Ячеистая топология допускает соединение большого количества компьютеров и характерна, как правило, для глобальных сетей. Достоинства данной топологии в ее устойчивости к отказам и перегрузкам, так как имеется несколько способов обойти отдельные узлы.

Смешанная топология. В таких сетях можно выделить отдельные подсети, имеющие типовую топологию – звезду, кольцо или общую шину, которые для крупных сетей связываются произвольно.



Построение локальных сетей. Передающая среда является общим ресурсом для всех узлов сети. Чтобы получить возможность доступа к этому ресурсу из узла сети, необходимы специальные механизмы – методы доступа. *Метод доступа к передающей среде* – метод, обеспе-

чивающий выполнение совокупности правил, по которым узлы сети получают доступ к ресурсу.

Маркерный доступ. Компьютер-абонент получает от центрального компьютера сети маркер-сигнал на право ведения передачи в течение определенного времени, после чего маркер передается другому абоненту.

При *конкурентном методе доступа* абонент начинает передачу данных, если обнаруживает свободную линию.

Сеть Ethernet. Схема передачи данных конкурентная, элементы сети могут быть соединены по шинной или звездной топологии с использованием витых пар, коаксиальных и волоконно-оптических кабелей. Основное преимущество – быстродействие от 10 до 100 Мб/с.

Сеть Token Ring. Схема с маркерным доступом. Физически выполнена как звезда, но ведет себя как кольцевая. Данные передаются последовательно от станции к станции, но постоянно проходят через центральный узел. Используются витые пары и волоконно-оптические кабели. Скорость передачи 4 или 16 Мб/с.

Сеть ARCnet. Схема с маркерным доступом, может работать как с шинной, так и звездной топологией. Совместима с витой парой, коаксиальным и волоконно-оптическим кабелем. Скорость передачи 2.5 Мб/с.

Беспроводные сети. *Беспроводные сети* являются одной из главных тенденций развития компьютерных сетей. Они дают возможность развертывания компьютерной сети там, где прокладка традиционной кабельной системы невозможна или нежелательна.

Беспроводные сетевые технологии и построенные на их основе сети передачи данных (БСПД) – это технологии и сети, позволяющие организовать передачу информации без использования кабельной проводки. В качестве среды передачи в таких сетях обычно выступают радиоволны, в редких случаях – инфракрасное излучение.

Беспроводные сети передачи данных классифицируют:

по географической протяженности:

– персональные (WPAN, Wireless Personal Area Network); локальные, региональные (WLAN, Wireless Local Area Network, WMAN Wireless Metropolitan Area Networks); глобальные (WWAN, Wireless Wide Area Network);

по применимой технологии связи:

– радиорелейные (радиосвязь по линии, образованной цепочкой приемопередающих радиостанций);

– спутниковые;
– атмосферные оптические (делают возможной передачу данных Интернет, голос, видео, телефония, телевидение между объектами в атмосфере, предоставляя оптическое соединение без использования оптоволоконна);

– мобильные сотовые;
– транковые (подвижная радиосвязь) и т. д.;

по каналу передачи данных:

– радио; лазерные; инфракрасные и т. д.;

по ширине полосы передачи (чем шире полоса, тем выше скорость передачи данных):

– узкополосные; широкополосные; сверхширокополосные;

по локализации абонентов:

– фиксированные; подвижные.

Одним из важнейших факторов, влияющих на работоспособность БСПД, является обеспечение качества передачи и приема радиосигнала. При этом необходимо обеспечить не только требуемую скорость, но и защиту передаваемой информации от помех и посторонних вторжений. Это является одним из главных недостатков по сравнению с проводными сетями.

Одна из основных проблем построения БПСД – решение **задачи доступа многих пользователей к ограниченному ресурсу среды передачи**, которой является радиоканал, т. е. другими словами, как отличить в радиоканале сигнал одного пользователя от сигнала другого.

Сети WPAN обеспечивают обмен информацией между такими устройствами, как карманные и обычные персональные компьютеры, мобильные телефоны, ноутбуки, принтеры, цифровые фотоаппараты, мышки, клавиатуры, джойстики, наушники, гарнитуры на надежной, недорогой, повсеместно доступной радиочастоте для ближней связи. Сети работоспособны в радиусе от 10 до 100 (дальность очень сильно зависит от преград и помех) даже в разных помещениях.

Наиболее известные сети WPAN – это сети, использующие технологии Bluetooth и ZigBee.

5.2. Глобальная сеть Интернет

В 1961 г. Defence Advanced Research Agency (DARPA – оборонное агентство передовых исследовательских проектов) по заданию министерства обороны США приступило к проекту по созданию экспери-

ментальной сети передачи пакетов. Эта сеть, названная ARPANET, первоначально предназначалась для изучения того, как поддерживать связь в случае ядерного нападения и для помощи исследователям в обмене информацией между разбросанными по всем штатам исследовательскими организациями оборонной промышленности.

В основу проекта были положены три основные идеи:

- каждый узел сети соединен с другими таким образом, что существует несколько различных путей от узла к узлу;
- все узлы и связи рассматриваются как ненадежные;
- существуют автоматически обновляемые таблицы перенаправления пакетов. Если нужный узел в данный момент недоступен, пакет, предназначенный для него, согласно этой таблице отправляется на ближайший к нему, затем на следующий и т. д.

Созданная по таким принципам система не имела централизованного узла управления и, следовательно, безболезненно могла изменять свою конфигурацию.

Эксперимент с ARPANET был настолько успешен, что многие организации захотели войти в нее с целью использования для ежедневной передачи данных. И в 1975 г. ARPANET превратилась из экспериментальной сети в рабочую сеть.

В конце 80-х гг. Россия подключилась к сети ARPANET. В 1990 г. сеть ARPANET перестала существовать, и на ее месте возник Интернет. Интернет сделала возможным свободный обмен информацией, невзирая на расстояния, государственные границы.

Слово *Интернет* происходит от выражения *interconnected networks* (связанные сети). В узком смысле – это *глобальное сообщество малых и больших сетей*, принадлежащих различным компаниям и предприятиям; работающих по самым разнообразным протоколам; связанных между собой различными линиями связи; передающих физически данные по телефонным проводам, оптоволокну, через спутники и радиомодемы. В более широком смысле – это *глобальное информационное пространство*, хранящее огромное количество информации на миллионах компьютерах, которые обмениваются данными. Каждый компьютер, подключенный к Интернету, – это часть Сети.

Интернет состоит из разнородных подсетей, соединенных друг с другом шлюзами. В качестве подсетей могут выступать разные локальные сети (Token Ring, Ethernet и т. п.), различные национальные, региональные и глобальные сети. К этим сетям могут подключаться

машины разных типов. Каждая из подсетей работает в соответствии со своими принципами и типом связи.

За Интернет никто централизованно не платит, каждый платит за свою часть. Представители сетей собираются вместе и решают, как им соединиться друг с другом и содержать эти взаимосвязи. Пользователь платит за подключение к некоторой региональной сети, которая в свою очередь платит за свой доступ сетевому владельцу государственного масштаба. Интернет не имеет никакого собственника, здесь нет и специального органа управления, который бы контролировал всю работу сети Интернет. Локальные сети различных стран финансируются и управляются местными органами согласно политике данной страны.

Существование Интернета стало возможным только после создания базовых протоколов ТСП/IP, которые и предназначены для объединения сетей. Эти протоколы работают на двух уровнях: ТСП – на транспортном, IP – на сетевом (адресном).

В основе принципа сети Интернет положен метод пакетной коммутации.

Коммутация пакетов – метод, специально разработанный для компьютерных сетей, где различные компьютеры сети могут иметь различное быстродействие. При коммутации пакетов все передаваемые сообщения разбиваются передающим компьютером на части (от 46 до 1500 байт), называемые пакетами. Каждый пакет снабжается заголовком, в котором указывается адресная информация, необходимая для доставки пакета к принимающему компьютеру, а также номер пакета, используемый для «сборки» сообщений на принимающем компьютере. Пакеты транспортируются в сети как независимые информационные блоки. Специальные устройства сети (коммутаторы) принимают пакеты от передающих компьютеров и на основании адресной информации передают их друг другу до конечного принимающего компьютера. За счет буферизации (задержки) пакета во внутренней памяти коммутатора (если требуемый участок сети занят передачей другой информации) выравнивается скорость передачи данных в сети в целом и повышается пропускная способность сети.

Поскольку компьютеры и сетевое оборудование могут быть от разных производителей, то возникает проблема их совместимости. Без принятия всеми производителями общепринятых правил построения оборудования создание компьютерной сети было бы невозможно. Поэтому разработка и создание компьютерных сетей может происходить только в рамках утвержденных стандартов.

В основу стандартизации компьютерных сетей положен принцип декомпозиции, т. е. разделения сложных задач на отдельные более простые подзадачи. При рассмотрении работы компьютера в сети можно выделить две основные подзадачи:

- взаимодействие программного обеспечения пользователя с физическим каналом связи (посредством сетевой карты) в пределах одного компьютера;
- взаимодействие компьютера через канал связи с другим компьютером.

Современное программное обеспечение компьютера имеет многоуровневую модульную структуру, т. е. программный код проходит несколько уровней обработки, прежде чем превратиться в электрический сигнал (модуль нижнего уровня), передаваемый в канал связи.

При взаимодействии компьютеров через канал связи оба компьютера должны выполнять ряд соглашений. Например, они должны согласовать величину и форму электрических сигналов, длину сообщений, методы контроля достоверности и т. д. На рис. 3.1 представлена модель взаимодействия двух компьютеров в сети. Процедура взаимодействия каждого уровня этих компьютеров может быть описана в виде набора правил взаимодействия каждой пары модулей соответствующих уровней.

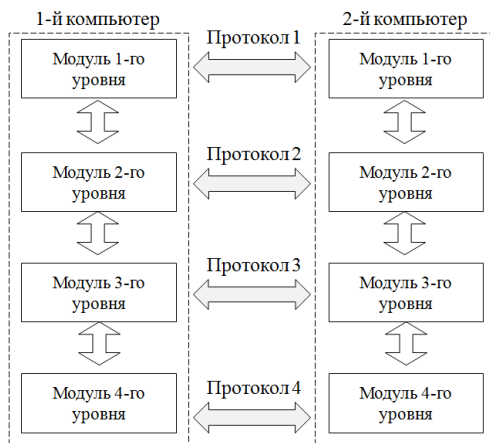


Рис. 3.1. Взаимодействие двух компьютеров в сети

Формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются модули, лежащие на одном уровне, но в различных компьютерах, называются *протоколами*.

Суть работы многоуровневого протокола можно представить как «письмо в конверте». Каждый уровень протокола надписывает на «конверте» свою информацию. Сетям нужно только понимать «надпись» на «конверте», чтобы предать его в место назначения, а до содержания письма им дела нет.

Модули, реализующие протоколы соседнего уровня и находящиеся в одном компьютере, также взаимодействуют друг с другом в соответствии с четко определенными правилами и с помощью стандартизованных форматов сообщений. Эти правила называются *интерфейсом* и определяют набор сервисов, предоставляемых данным уровнем соседнему уровню. Модули должны обрабатывать, во-первых, свой собственный протокол, а во-вторых, интерфейсы с соседними уровнями.

Иерархически организованный набор протоколов для взаимодействия компьютеров в сети называется *стеком коммуникационных протоколов*. Коммуникационные протоколы могут быть реализованы как программно, так и аппаратно. Протоколы нижних уровней, как правило, реализуются комбинацией программно-аппаратных средств, а протоколы верхних уровней – чисто программными средствами. Протоколы каждого уровня обладают независимостью друг от друга, т. е. протокол любого уровня может быть изменен, не оказывая при этом никакого влияния на протокол другого уровня. Независимость протоколов посредством интерфейсов является важнейшей предпосылкой для создания стандартных протоколов для компьютерных сетей.

Международной организацией по стандартизации (ISO – International Standards Organization) была предложена модель взаимодействия двух компьютеров в сети, названная OSI (Open system interconnection – взаимодействие открытых систем). В ней реализована семиуровневая архитектура, причем каждому из уровней соответствует определенная функция (рис. 3.2).

Взаимодействие компьютеров в сети начинается с того, что приложение (программа пользователя) одного компьютера обращается к прикладному уровню другого компьютера, например, к файловой системе. Приложение первого компьютера формирует с помощью операционной системы сообщение стандартного формата, состоящее из заголовка и поля данных.

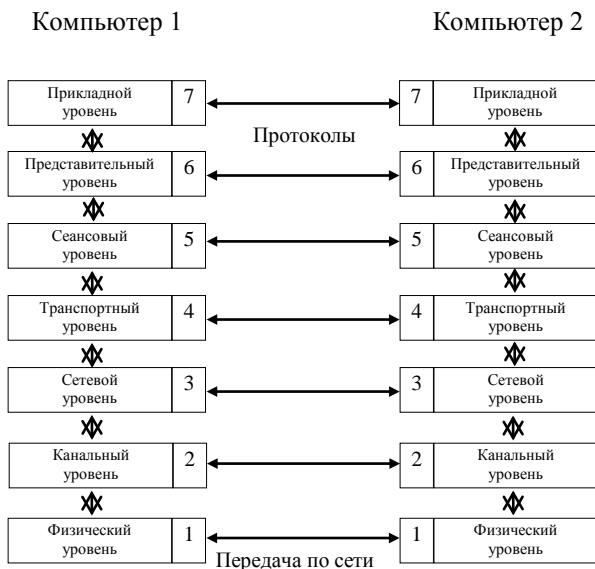


Рис. 3.2. Модель OSI

Заголовок содержит служебную информацию, которую необходимо передать через сеть прикладному уровню другого компьютера, чтобы сообщить ему, какую работу необходимо выполнить. В заголовке имеется информация для следующего нижнего уровня, чтобы он «знал», что делать с этим сообщением. В поле данных находится передаваемая информация. Сформировав сообщение, прикладной уровень направляет его представительному уровню, который, прочитав заголовок, выполняет требуемые действия над сообщением и добавляет к сообщению собственную служебную информацию (указания для протоколов представительного уровня второго компьютера). Полученное в результате сообщение передается сеансовому уровню, который в свою очередь добавляет свой заголовок и т. д. При достижении сообщением физического уровня у него имеется множество заголовков, добавленных на предыдущем уровне. В таком виде оно и передается по сети (рис. 3.3).

Второй компьютер принимает его на физическом уровне и последовательно перемещает вверх с уровня на уровень (рис. 3.3).

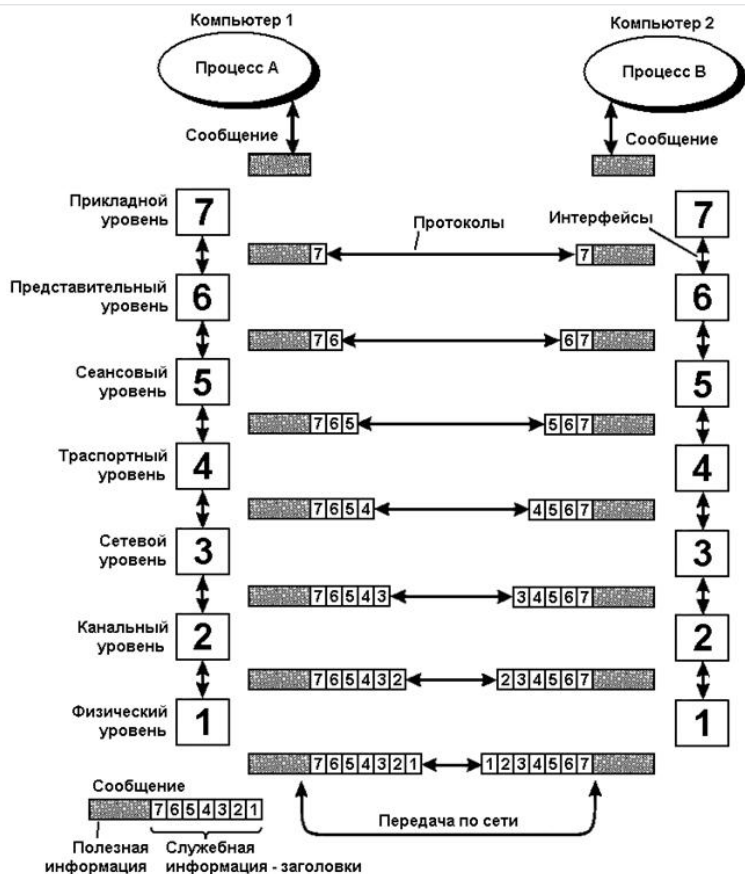


Рис. 3.3. Взаимодействие компьютеров в сети

Каждый уровень анализирует и обрабатывает заголовок своего уровня, выполняя соответствующие этому уровню функции, а затем удаляет этот заголовок и передает сообщение вышележащему уровню. В результате взаимодействия протоколов всех уровней и их единому стандарту на прикладном уровне второго компьютера получают данные, переданные первым компьютером.

В стандарте OSI для обозначения единиц данных, с которыми имеют дело протоколы различных уровней, используются специальные названия: *кадр*, *пакет*, *дейтаграмма*, *сегмент*.

Физический уровень имеет дело с передачей битов информации по физическим каналам связи (коаксиальный кабель, витая пара, оптоволоконный кабель). На этом уровне стандартизируются характеристики электрических сигналов, уровни напряжения и тока, тип кодировки информации, скорость передачи сигналов, а также типы разъемов и назначение каждого контакта. Пример стандарта физического уровня: спецификация 10Base-T технологии Ethernet, которая использует витую пару с волновым сопротивлением 100 Ом, разъем RJ45, максимальную длину 100 м.

Канальный уровень обеспечивает надежную передачу данных через физический канал. Канальный уровень оперирует блоками данных, называемых *кадрами*. Основной задачей канального уровня является прием кадра из сети и отправка его в сеть. Он осуществляет физическую адресацию передаваемых сообщений, контролирует соблюдение правил использования физического канала, выявляет неисправности, управляет потоками информации. Кадр может быть доставлен только в случае, если протокол соответствует той топологии, для которой он был разработан (общая шина, кольцо и звезда). Протоколы канального уровня для локальных сетей – Ethernet, Token Ring, FDDI, для глобальных – PPP, SLIP, LAP-B, LAP-D.

Для реализации протоколов канального уровня используется специальное оборудование – *коммутаторы*. Раньше использовались *концентраторы и мосты*, которые в настоящее время сняты с производства.

Сетевой уровень служит для образования единой системы, объединяющей несколько сетей, причем эти сети могут быть различной топологии, использовать совершенно различные принципы сообщений между конечными узлами и обладать произвольной структурой. Сети соединяются между собой специальными устройствами – *маршрутизаторами*. *Маршрутизатор* – устройство, которое собирает данные о топологии межсетевых соединений и на ее основании пересылает пакеты информации из одной сети в другую. Последовательность маршрутизаторов, через которые проходит пакет, называется *маршрутом*, а выбор маршрута называется *маршрутизацией*. На сетевом уровне действуют три протокола: *сетевой протокол* – для определения правил передачи пакетов от конечных узлов к маршрутизаторам и между маршрутизаторами; *протокол маршрутизации* – для сбора информации о топологии сети; *протокол разрешения адресов* – для отображения адреса узла, используемого на сетевом уровне в локальный адрес

сети. Примеры протоколов сетевого уровня: протокол межсетевого взаимодействия IP стека TCP/IP и протокол межсетевого обмена пакетами IPX стека Novell.

Транспортный уровень предназначен для оптимизации передачи данных от отправителя к получателю с той степенью надежности, которая требуется. Основная задача транспортного уровня – обнаружение и исправление ошибок в сообщениях. Начиная с транспортного уровня, все дальнейшие протоколы реализуются программным обеспечением компьютера, включаемого обычно в состав сетевой операционной системы. Примеры транспортных протоколов: протоколы TCP стека TCP/IP и протокол SPX Novell.

Сеансовый уровень управляет диалогом между двумя компьютерами и устанавливает правила начала и завершения взаимодействия, определяет, какая из сторон является активной в данный момент, а какая принимает данные. Приложение должно различать разные потоки данных в пределах одного соединения (например, приложение может одновременно запрашивать два файла с одного сервера, благодаря сеансовому уровню оно будет различать эти два потока).

Представительный уровень выполняет преобразование данных между устройствами с различными форматами данных, не меняя при этом содержания. Благодаря этому уровню информация, передаваемая прикладным уровнем одного компьютера, всегда понятна прикладному уровню другого компьютера. На этом уровне, как правило, происходит шифрование и дешифрование данных, благодаря которому обеспечивается секретность передаваемого сообщения.

Прикладной уровень является пользовательским интерфейсом для работы с сетью. Он непосредственно взаимодействует с пользовательскими прикладными программами, предоставляя им доступ в сеть. С помощью протоколов этого уровня пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры, гипертекстовые web-страницы, электронная почта.

Необходимо отметить, что три нижних уровня модели OSI – физический, канальный и сетевой – зависят от сети, т. е. их протоколы тесно связаны с технической реализацией сети и с используемым коммутационным оборудованием. Три верхних уровня – сеансовый, представления и прикладной – реализуются программным способом и мало зависят от особенностей построения сети. Транспортный уровень является промежуточным. Он скрывает детали функционирования нижних уровней от верхних уровней. Благодаря этому уровню можно раз-

рабатывать приложения, не зависящие от технических средств транспортировки сообщений.

Системы, взаимодействующие в соответствии с моделью OSI, называются *открытыми системами*. Для них опубликованы общедоступные спецификации и стандарты, принятые в результате достижения согласия многих разработчиков и пользователей после всестороннего обсуждения. Если две сети построены с соблюдением правил открытости, то у них есть возможность использования аппаратных и программных средств разных производителей, придерживающихся одного и того же стандарта, они легко сопрягаются друг с другом, просты в освоении и обслуживании. Примером открытой системы является глобальная компьютерная сеть Интернет. В результате сеть Интернет сумела объединить в себе самое разнообразное оборудование и программное обеспечение сетей, разбросанных по всему миру.

Стек протоколов TCP/IP (Transmission Control Protocol/Internet Protocol) является базовым протоколом сети Интернет.

Базовые протоколы Интернета. Применительно к Интернету протокол – это правило передачи информации в сети. Различают два типа протоколов: базовые (TCP/IP) и прикладные.

Базовый протокол TCP (протокол управления передачей) является протоколом *транспортного* уровня Интернета. Именно он нарезает файлы на пакеты определенной длины, маркирует их, следит за тем, чтобы все пакеты дошли до места назначения неповрежденными, в случае повреждения требует повторной пересылки пакета, а также собирает пакет, доставленный по месту назначения. Благодаря таким свойствам TCP Интернет обладает высокой устойчивостью и способностью самозалечивать повреждения.

Базовый протокол IP (протокол взаимодействия сетей) является протоколом *сетевого (адресного)* уровня Интернета. Если протокол TCP определяет, как происходит передача, то протокол IP определяет, куда она происходит.

На сетевом уровне протокол IP каждому TCP-пакету приписывает адрес отправителя и адрес получателя. По этому IP-заголовку маршрутизатор, через который проходит пакет, и определяет, куда нужно передать пакет дальше (к какому из «ближайших» соседей, понятие ближайший – не географическое), а если с пакетом что-то случается, куда обратиться за повторной передачей.

Каждый пакет даже одного сообщения путешествует самостоятельно, причем пакет, отправленный позже, может прийти раньше.

Но независимо от траектории в результате конечного числа пересылок TCP-пакеты достигают адресата, где затем собираются и передаются на более высокие уровни.

Основные преимущества протокола TCP/IP:

– *независимость от сетевой технологии отдельной сети* – определяет только элемент передачи – *дейтаграмму* – и описывает способ ее движения по сети;

– *всеобщая связанность сетей* – позволяет любой паре компьютеров взаимодействовать друг с другом, каждому компьютеру назначается логический адрес, а каждая дейтаграмма содержит адреса отправителей и получателей. Промежуточные маршрутизаторы используют адрес получателя для принятия решения о маршрутизации;

– *подтверждение* – обеспечивает подтверждение правильности прохождения информации при обмене между отправителем и получателем.

В стеке TCP/IP определены 4 уровня взаимодействия, каждый из которых берет на себя определенную функцию по организации надежной работы глобальной сети: прикладной уровень, основной (транспортный) уровень, уровень межсетевое взаимодействие, уровень сетевых интерфейсов.

Хост – объект сети, который может передавать и принимать IP-пакеты, например, компьютер или маршрутизатор.

По модели протокола TCP/IP – программные модули, соответствующие транспортному и сеансовому уровню, устанавливаются только на конечных компьютерах (хостах).

Программный модуль протокола TCP/IP реализуется в операционной системе компьютера в виде драйвера.

Адресация в Интернет. Для выхода каждого компьютера в Интернет устанавливаются два уникальных адреса: цифровой IP-адрес и доменный адрес. Оба эти адреса могут применяться равноценно. Цифровой адрес удобен для обработки на компьютере, а доменный адрес – для восприятия пользователем.

Цифровой адрес имеет длину 32 бита. Для удобства он разделяется на четыре блока по 8 бит:

XXXXXXXX.XXXXXXXXXX.XXXXXXXXXX.XXXXXXXXXX

Адрес содержит полную информацию, необходимую для идентификации компьютера, читается слева направо, т. е. вначале указывается адрес сети, а затем адрес компьютера в сети.

Сеть типа А – под адрес сети выделяется 1 блок, остальные под адрес компьютера. **Сеть типа В** – под адрес сети выделяется 2 блока. **Сеть типа С** – под адрес сети выделяется три блока, следовательно, на адрес компьютера в сети С остается только 8 бит, значит, максимальное количество компьютеров, которое можно подключить, – 255, из них два адреса зарезервированы, следовательно, максимально можно подключить 253 компьютера.

Пример. В двоичном коде цифровой адрес в сети С записан следующим образом: 11000000.00101101.00001001.11001000. В десятичном коде он имеет вид: 192.45.9.200. Адрес сети – 192.45.9; адрес компьютера – 200.

Человек не в состоянии запомнить такое большое количество цифровых адресов, с ними работает компьютер. Для пользователя были введены текстовые адреса – **доменные имена**. Например, адрес <http://www.microsoft.com/> – содержит доменное имя microsoft.com, E-mail адрес alpron@aha.ru – содержит доменное имя aha.ru.

В отличие от цифрового адреса, доменный адрес читается в обратном порядке. Вначале идет имя компьютера, затем имя сети, в которой он находится.

Например: <http://www.pc.dpt1.company.msk.ru>.

Первым в имени стоит название реального компьютера с адресом pc. Это имя создано и поддерживается группой dpt1. Группа входит в более крупное подразделение company, далее следует домен msk, определяющий имена московской части сети, а ru – российской.

За соответствием цифровых и доменных адресов в Интернете следит служба DNS.

Прикладные протоколы и сервисы Интернета. В Интернете существует множество разнообразных служб (сервисов), предоставляющих те или иные услуги. Каждая служба имеет свою прикладную программу и использует свой прикладной протокол.

Сервис	Протокол	Программа	Пояснение
Электронная почта e-mail	SMTP, POP2, POP3, IMAP	MS Outlook Express, The Bat, Eudora Pro	Почтовая служба
WWW	Http	Браузеры: Microsoft Internet Explorer, Netscape Navigator, Opera	Просмотр web-страниц (гипертекста)

Сервис	Протокол	Программа	Пояснение
Ftp	ftp	Браузеры	Передача файлов
UseNet	nntp	MS Outlook Express	Телеконференции (группы новостей)
IRC (Internet Relay Chat)		mIRC.exe	Общение в режиме реального времени
Telnet		Telnet	Удаленное управление компьютером

3.3. Основы проектирования web-страниц

HTML (от 5нгл. *HyperText Markup Language* – язык гипертекстовой разметки) – стандартизированный язык гипертекстовой разметки документов для просмотра web-страниц в браузере. Web-браузеры получают документ HTML от сервера по протоколам HTTP/HTTPS или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора. Web-страница представляет собой документ HTML. Этот документ может содержать:

- текст и описание его форматирования;
- определение структуры и формата страницы: может задаваться, например, деление текста на строки и абзацы, заголовок страницы, поля, цветовая разметка страницы, фоновое изображение и фоновый звук;
- указание путей к мультимедийным объектам (сами мультимедийные объекты хранятся в отдельных файлах), а также способ их форматирования;
- таблицы;
- гиперссылки;
- активные компоненты или ссылки на документы, содержащие активные компоненты.

Интерпретирует код HTML программа-браузер, поэтому при создании web-страниц на языке HTML необходимо как можно чаще тестировать созданный документ, открывая его в браузере.

Документ HTML содержит объекты всего двух типов: текст и так называемые теги (управляющие конструкции). Именно с помощью тегов задается форматирование текста, структура и формат страницы, вставляются и формируются мультимедийные объекты, таблицы, гиперссылки и прочие элементы web-страниц.

У каждого тега есть свое уникальное имя. Для того чтобы теги отличались от обычного текста, имена их заключают в угловые скобки:

<имя_тега>. Имя тега может содержать латинские буквы и цифры. При записи имен тегов регистр не имеет значения.

Все теги HTML можно разделить на **парные** и **непарные**. Парные теги окружают тот фрагмент страницы, на который распространяется их действие.

Поясним это на примере. Пусть в предложении «Браузер – это программа для просмотра web-страниц.» слово «Браузер» необходимо выделить полужирным начертанием. Тег, задающий полужирное написание, носит имя ****. Таким образом, соответствующий код HTML должен выглядеть так:

```
<b>Браузер</b> – это программа для просмотра web-страниц.
```

Обратите внимание на то, что слово «Браузер» окружено парой тегов ****. Первый тег (открывающий) указывает начало фрагмента, на который должно распространяться действие тега, второй тег (закрывающий) – указывает конец этого фрагмента. В закрывающем теге сразу после угловой скобки (<) необходимо ставить знак **</>**. Браузер интерпретирует этот код HTML так:

```
Браузер – это программа для просмотра web-страниц.
```

Существуют и непарные теги, присутствие которых в определенном месте документа указывает на то, что здесь должен появиться некий объект. Для непарных тегов закрывающий тег отсутствует. Так, тег **** указывает на то, что в данной области документа должно появиться графическое изображение.

Большинство тегов имеет свой набор **атрибутов** (параметров). Атрибуты уточняют действие тегов. Как правило, использование атрибутов необязательно. Для парных тегов атрибуты указывают в открывающем теге. Атрибуту может присваиваться заданное или произвольное значение. Для одного и того же тега допускается использование нескольких атрибутов, разделенных пробелами.

Так, тег **** имеет ряд атрибутов: **src** – указывает путь к графическому файлу – источнику изображения, **width** и **height** – ширину и высоту изображения (в пикселах или процентах), **alt** – альтернативный текст и др. Например:

```

```

При интерпретации кода HTML браузер отобразит изображение из указанного графического файла шириной 200 пикселей и высотой 150 пикселей. Атрибут **alt** не задан, значит, альтернативный текст отсутствует.

Регистр имен тегов и атрибутов может быть любым. Так, тег вставки графического изображения можно записать как ``, ``, `` и т. д.

Наличие и количество пробелов между тегом и текстом или соседними тегами не имеет значения. Значения атрибутов необязательно заключать в кавычки, если они содержат только буквы, цифры, точки и дефисы.

На один и тот же фрагмент страницы может распространяться действие сразу нескольких тегов. Так, пусть фрагмент текста необходимо выделить полужирным начертанием и курсивом. Тег курсива – `<i>`. Тогда код HTML записывают следующим образом:

```
<b><i>фрагмент текста</i></b>
```

Необходимо соблюдать порядок вложенности тегов. Соответствующие конструкции должны вкладываться друг в друга, не пересекаясь. Запись, подобная приведенной ниже, является неверной.

```
<b><i>фрагмент текста</b><i>
```

Поэтому многие браузеры могут неправильно интерпретировать такой HTML-код.

Следует учитывать и тот факт, что браузеры не проверяют написания кода HTML. Неправильно записанный и нераспознанный браузером фрагмент кода игнорируется. При этом никакого сообщения об ошибке браузер выдавать не будет. Таким образом, еще раз напомним о необходимости периодического просмотра документа HTML в браузере.

Версии языка HTML разрабатываются Консорциумом Всемирной паутины (World Wide Web Consortium, W3C) – организацией, разрабатывающей для Интернета единые принципы и стандарты (рекомендации), которые внедряются производителями программных продуктов и оборудования.

На момент написания курса лекций действующим стандартом является HTML5 (был представлен в 2017 г.).

Для всех документов в HTML5:

```
<!DOCTYPE html>
```

Структура документа HTML. Общий вид документа HTML следующий:

```
<html>  
<head>  
служебная область документа  
</head>
```

```
<body>
содержательная область документа
</body>
</html>
```

Итак, документ HTML начинается и заканчивается тегами `<html>`. Служебная область документа ограничивается парным тегом `<head>`. Она содержит заголовок страницы (это единственные данные из этой области, которые увидит пользователь при просмотре страницы в браузере), а также информацию для браузеров и поисковых систем, которые будут индексировать страницу.

Собственно содержимое страницы располагается внутри парного тега `<body>`. Именно о содержательной части HTML-документа и пойдет речь далее.

Цветовая разметка документа. Цвет в HTML.

Цвет в HTML можно задать двумя способами. Первый способ – указать название цвета. В HTML есть названия для 256 цветов, но не все браузеры распознают эти названия. Однако названия основных 16 цветов способны распознать все графические браузеры, и лучше использовать только эти цвета. Приведем их названия:

black – черный, **silver** – светло-серый, **gray** – серый, **white** – белый, **maroon** – бордовый, **red** – красный, **purple** – фиолетовый, **fuchsia** – лиловый, **green** – зеленый, **lime** – желто-зеленый, **olive** – оливковый, **yellow** – желтый, **navy** – темно-синий, **blue** – синий, **teal** – сине-зеленый, **aqua** – голубой.

Если ни один из перечисленных цветов не подходит, необходимо определить цвет, используя схему RGB.

Схема RGB (Red-Green-Blue – красный-зеленый-синий) предполагает, что цвет создается путем смешения в определенных пропорциях красного, зеленого и синего цветов. Интенсивность каждого компонента записывается как двузначное шестнадцатеричное число. Например, цвет A26720 означает, что взято достаточно много красного цвета, меньше зеленого, еще меньше синего. В результате получится терракотовый цвет.

Итак, фрагмент кода HTML

```
<body bgcolor=maroon text="#256AC1">
```

означает, что фон документа будет бордовым, а цвет текста – синим с примесью зеленого и небольшой примесью красного. Обратите внимание на то, что если цвет задается по схеме RGB, то перед значением цвета необходимо ставить символ **#** и обязательно заключать значение атрибута в кавычки.

Атрибуты цветовой схемы не рекомендуются к использованию в спецификации HTML 5, хотя и поддерживаются в настоящее время большинством браузеров. Для задания цветовой разметки рекомендуется использовать CSS (Cascading Style Sheets – каскадные таблицы стилей).

Заголовок web-страницы является единственным элементом в служебной области HTML-документа, который будет отображаться при просмотре страницы в браузере. Заголовок записывается внутри парного тега **<title>**.

В служебной области HTML-документа также могут находиться **Meta-теги**, которые содержат информацию для поисковых машин и для браузеров.

Каскадные таблицы стилей CSS. Преимущества использования.

Для форматирования элементов web-страницы можно применять **каскадные таблицы стилей** (Cascading Style Sheets, CSS), описывающие способы представления (отображение и расположение) элемента (или группы элементов) web-страницы. Преимущества использования каскадных таблиц стилей состоят в следующем:

- расширяются возможности форматирования (по сравнению с HTML);
- в основной части документа отсутствуют теги физического форматирования текста, т. е. представлена лишь структурная разметка документа;
- сокращается объем HTML-документа;
- возможно единое стилевое оформление для группы web-страниц.

Стандарты CSS, как и HTML, разрабатывает и рекомендует Консорциум Всемирной паутины (World Wide Web Consortium, W3C).

Синтаксис CSS. Технология CSS предполагает создание *правил*, которые переопределяют отображение тегов в документе или задают пользовательский стиль. Правило состоит из *селектора* и *определения* – свойств и их значений – и записывается следующим образом:

селектор {свойство1: значение1; свойство2: значение2;...}

Селектор указывает элемент, который необходимо форматировать.

Это может быть тег (форматирование будет применено ко всем элементам данного типа), класс (форматирование можно применять выборочно к нескольким элементам) или идентификатор (форматирование можно применять только к одному элементу).

Определение указывает способ оформления. Оно заключается в фигурных скобки, внутри которых перечислены свойства и их значения. Свойства определяют параметры форматирования. Например, свойство **color** указывает цвет элемента. Свойствам присваивается заданное или произвольное (в зависимости от свойства) значение. Между свойством и значением ставят двоеточие. Регистр символов и разбиение правила на строки не имеют значения.

Например:

```
p {color: red}
```

Такое правило означает, что во всех абзацах цвет шрифта должен быть красным.

Для одного и того же селектора можно указать несколько свойств. В таком случае их нужно перечислить в фигурных скобках через точку с запятой, например:

```
p {color: red; text-weight: bold}
```

Это означает, что во всех абзацах шрифт должен быть красного цвета и полужирный. Порядок записи свойств не имеет значения.

Если для разных селекторов необходимо указать одинаковые параметры форматирования, селекторы можно сгруппировать, перечислив через запятую:

```
h1, h2, h3 {color: green; text-align: center}
```

Это означает, что заголовки 1, 2 и 3-го уровней необходимо отображать зеленым цветом и выравнивать их по центру.

Для улучшения восприятия CSS-кода принято записывать селектор и каждое свойство в отдельных строках:

```
селектор {  
свойство1: значение1;  
свойство2: значение2;  
...  
}
```

Например, приведенные ниже правила CSS для заголовков 1-го уровня являются идентичными и любой из представленных вариантов записи считается допустимым.

```
h1 {color: green; font-size:20px}      h1 {  
color: green;  
font-size: 20px  
}
```

При создании правил CSS следует грамотно выбрать селектор. Если стилевые параметры планируется применить ко всем элементам

определенного типа, то в качестве селектора будет использоваться **имя тега**. Если стилевые параметры следует назначить только некоторой группе элементов (элементам, принадлежащим к определенному классу), то селектором будет **.имя_класса**. Если же стилевые параметры задаются для одного элемента с определенным именем (идентификатором), то селектором будет **#имя_идентификатора**.

Классы. Если необходимо изменить способ отображения всех элементов одного и того же типа, записывают правило, селектором которого является соответствующий тег. Если же форматирование нужно применить лишь к некоторым элементам, следует создать **класс**. Тогда селектором в правиле будет имя класса (оно должно содержать только латинские буквы, цифры, символы дефиса (-) и подчеркивания (_)). При записи правила для класса перед селектором ставится точка. Например:

```
.special {color: blue; font-size: 14 pt; text-align: center}
```

В данном примере описан класс **special**, задающий синий цвет шрифта, размер 14 пунктов и выравнивание текста по центру. Чтобы применить его к конкретному элементу, в соответствующий тег добавляют атрибут **class** и присваивают ему значение, соответствующее имени класса (без точки в начале):

```
<p class="special">  
<h1 class="special">
```

Если предполагается, что класс будет применяться только к элементам определенного типа, т. е. к конкретному тегу, то запись правила будет выглядеть:

```
p.special {color: blue; font-size: 14 pt; text-align: center}
```

В таком случае класс **special** можно применять только к абзацам.

Идентификаторы. Идентификатор определяет уникальное имя элемента для изменения его стиля. С помощью идентификатора можно обратиться к элементу через скрипт. Имя идентификатора должно содержать только латинские буквы, цифры, символы дефиса (-) и подчеркивания (_). При записи правила для идентификатора перед селектором ставится символ #:

```
#vvedenie {font-style: italic; text-align: right}
```

Этот идентификатор можно применять только к одному элементу на странице, используя атрибут **id**, например:

```
<p id="vvedenie">
```

Добавление стилей в HTML-документ: внедрение и связывание. Универсальный атрибут style.

Существуют два способа добавления стилей в HTML-документ:

1) таблица стилей задается непосредственно в HTML-документе (**внедрение**);

2) таблица стилей хранится в отдельном файле *.css и подключается к HTML-документу (**связывание**).

В первом случае используется тег `<style>`, который располагается в служебной области HTML-документа `<head>` и задает стилевую информацию для текущего документа. Кроме того, можно использовать атрибут **style** для определения стилевых параметров конкретного элемента документа.

Стилевая информация может храниться непосредственно в HTML-документе, в служебной области между тегами `<style>`:

```
<head>
<style>
h2 {color: green; font-size: 20 pt}
p.special {color: blue; font-size: 14 pt; text-align: center}
#vvedenie {font-style: italic; text-align: right}
</style>
</head>
```

Такой способ добавления стилей называется **внедрением**. Заметим, что в этом случае стилевое форматирование работает только для одной web-страницы. Чтобы применить такое же форматирование для других страниц, в служебных областях соответствующих HTML-документов также должен присутствовать тег `<style>` с описанием тех же правил.

Чтобы избежать многократного повторения одних и тех же правил в разных документах, таблицу стилей можно хранить в отдельном файле (с расширением **.css**). В этом случае таблица стилей присоединяется к HTML-документу с помощью тега `<link>` в служебной части документа:

```
<link href="адрес файла css" rel="stylesheet" type="text/css">
```

Такой способ добавления стилей называется **связыванием**. В этом случае одну и ту же таблицу стилей можно присоединить к нескольким.

Контрольные вопросы

1. Какие объекты могут содержаться внутри документа HTML?
2. Чем отличаются парные и непарные теги?
3. Каковы основные правила записи тегов и их атрибутов?
4. Какие теги определяют служебную и содержательную области документа HTML?
5. Назовите способы указания цвета в HTML.
6. Какие теги применяются для физического форматирования текста? Укажите недостатки физического форматирования текста.
7. Что представляет собой логическое форматирование текста?
8. Для чего служит тег ``? Почему атрибут `src` этого тега является обязательным?
9. Как задать таблицу в HTML?
10. Для чего используются Meta-теги и каковы их атрибуты?

ЛИТЕРАТУРА

1. Мамаев, М. Технологии защиты информации в Интернете: специальный справочник / М. Мамаев, С. Петренко. – Санкт-Петербург: Питер, 2002. – 848 с.
2. Таненбаум, Э. Компьютерные сети / Э. Таненбаум. – 5-е изд. – Санкт-Петербург: Питер, 2019. – 960 с. ил. – (Классика Computer Science).
3. Олифер, В. Г. Компьютерные сети: принципы, технологии, протоколы: учеб. пособие / В. Г. Олифер, Н. А. Олифер. – 4-е изд. – Санкт-Петербург: Питер, 2012. – 944 с. ил. – (Учебник для вузов).