**2 неделя**

**Лекция 3-4** Среды применения Web-технологий. Клиент-серверная архитектура в Интернет. Клиент-серверное взаимодействие компьютеров и приложений. Понятия ISP, POP,NAP, ''последняя миля‘’ в Интернет. Варианты доступа в Интернет для различных категорий пользователей и из сетей различного уровня. Браузеры и серверы. Прокси-сервер. Формат URL – адреса.

Строго говоря, следует отличать технологию клиент-сервер в широком смысле, которая может быть использована в любых компьютерных системах от собственно архитектуры клиент-сервер применительно к информационным приложениям вообще и автоматизированным системам управления предприятием особенно.

Согласно онлайновому словарю компьютерных терминов, клиент-сервер - это вид распределенной системы, в которой есть сервер, выполняющий запросы клиента, причем сервер и клиент общаются между собой с использованием того или иного протокола.

Под клиентом понимается программа, использующая ресурсы, а под сервером (по английски - слуга) программа, обслуживающая запросы клиентов на получение ресурсов определенного вида. Столь широкое определение включает в себя практически любую программную технологию, в которой участвуют больше одной программы, функции между которыми распределены асимметрично. Соответственно, говорят о технологии КС применительно к операционным системам, локальным и глобальным сетям и т. д.

Такое широкое определение рождает некоторую путаницу. Так, файл-серверная система тоже использует технологию клиент-сервер, однако с точки зрения архитектуры прикладных программ важным является то, какого рода ресурсы сервер предоставляет клиентам.

Понятие архитектуры клиент-сервер в системах управления предприятием связано с делением любой прикладной программы на три основных компонента или слоя. Этими тремя компонентами являются:

* компонент представления (визуализации) данных;
* компонент прикладной логики;
* компонент управления базой данных.

Действительно, любая программа, компьютеризирующая выполнение той или иной прикладной задачи, должна обмениваться информацией с пользователем, осуществлять собственно обработку этой информации в рамках автоматизации того или иного бизнес-процесса, и, наконец, хранить данные используемые в программе, на том или ином постоянном носителе.

Для локальных приложений, полностью работающих на ПЭВМ (например, Word или Excel), все эти компоненты собраны вместе и не могут быть распределены между различными компьютеры. Такая программа является монолитной и использует для выполнения ресурсы только того компьютера, на котором выполняется.

В файл-серверных приложениях часть компоненты хранения переносится на файловый сервер, однако, все манипуляции со структурами данных выполняются на клиентской машине, и код пользовательской программы тоже работает только на ней.

Критерием, позволяющим отнести прикладную программы к архитектуре клиент-сервер является то, что хотя бы один из трех ее компонентов полностью выполняется на другом компьютере, и взаимодействие между компонентами на разных компьютерах осуществляется через ту или иную сетевую среду посредством передачи запросов на получение того или иного ресурса.

Поскольку архитектура клиент-сервер является частным случаем технологии клиент-сервер, в ней обязательно есть клиент и сервер. Соответственно, выделяют клиентскую и серверную стороны приложения. Клиентская сторона приложения функционирует на рабочем месте пользователя, в роли которого в подавляющем числе случаев выступает персональный компьютер. Серверная сторона функционирует на специализированном комплексе, включающем в себя мощные аппаратные средства, требуемый набор стандартного программного обеспечения, систему управления базами данных и собственно структуры данных.

Взаимодействие клиентской и серверной частей приложения осуществляется через сеть - локальную или глобальную. При этом с точки зрения клиента и сервера взаимодействие осуществляется прозрачно, соответственно сетевой компонент здесь включает в себя совокупность необходимого сетевого оборудования, набор программных технологий, обеспечивающих передачу данных между узлами сети, а также собственно протокол или протоколы для обмена запросами и результатами их выполнения.

Компонент визуализации прикладной задачи осуществляет ввод информации пользователем с помощью тех или иных средств, а также вывод информации на экран и печать. Компонент визуализации для архитектуры клиент-сервер всегда исполняется на рабочем месте пользователя (поскольку должен же он наблюдать какие-либо результаты работы программы).

Компонент прикладной логики решает собственно ту или иную задачу, связанную с обработкой данных в той или иной предметной области. Этот компонент может быть распределен между клиентской и серверной частью различным образом в зависимости от применяемой модели.

Компонент хранения базы данных осуществляет физические операции, связанные с хранением данных, чтением информации из БД и записью в нее. В архитектуре клиент-сервер этот компонент всегда выполняется на сервере.

С точки зрения количества составных частей клиент-серверные системы делятся на двухуровневые и трехуровневые. Двухуровневые системы состоят только из клиента и сервера. В трехуровневых же между пользовательским клиентом и сервером, осуществляющим хранение и обработку базы данных появляется третий, промежуточный слой, являющийся для пользователя сервером, а для системы управления базами данных - клиентом. Такая архитектура позволяет более гибко распределять функции системы и нагрузку между компонентами программно-аппаратного комплекса, а также может снизить требования к ресурсам рабочих мест пользователей. Необходимой платой за это является то, что подобные системы намного сложнее в разработке, внедрении и эксплуатации и требуют значительных затрат и высококвалифицированного персонала.

В архитектуре клиент-сервер выделяются несколько различных моделей приложения, в зависимости от распределения компонентов приложения между клиентской и серверной частями. Исторически самой первой была разработана модель сервера удаленного доступа к данным. В этой модели серверная часть осуществляет только хранение данных, а всю прикладную логику реализует клиентская часть. При этом клиент будет передавать серверу запросы на получение данных, а сервер возвращать клиенту те или иные выборки. Самым распространенным средством общения между клиентом и сервером в этом случае является SQL (структурированный язык запросов) - стандартный непроцедурный язык, ориентированный на обработку данных.

В модели сервера удаленного доступа к данным на стороне сервера не исполняется никакой прикладной части системы, что может повлечь за собой недогрузку сервера и перегрузку клиента. Поэтому впоследствии была предложена, а затем реализована архитектура сервера базы данных. В ней часть прикладной логики реализуется на сервере, при помощи специального языка программирования, а часть - на клиенте. Это стало возможным благодаря росту производительности серверов современных СУБД. По сравнению с вариантом сервера удаленного доступа к данным, в данном случае несколько уменьшается нагрузка на клиентскую часть, интенсивность сетевого обмена данными, а также в ряде случаев упрощается структура приложения. В настоящее время этот вариант построения систем является самым распространенным.

Еще одним вариантом архитектуры клиент-сервер является сервер приложений. В данном случае клиент выполняет только операции визуализации и ввода данных, а всю прикладную логику реализует сервер. Обмен между клиентом и сервером в таких системах осуществляется на уровне команд вывода данных на экран и результатов пользовательского ввода. Наиболее ярким примером данной архитектуры является хорошо известный веб-браузер. Чаще всего, в модели сервера приложений компоненты прикладной логики и управления данными реализуются раздельно.

Архитектуру сервера приложений часто называют так называемым "тонким" клиентом, в отличие от традиционного "толстого" клиента, реализуемого в архитектуре сервера баз данных. "Тонкий" клиент является вариантом, который может быть использован, когда ресурсов, доступных на рабочих местах пользователей, недостаточно для исполнения логики приложения. Кроме того, эта технология позволяет сократить расходы на эксплуатацию клиентских компонент системы за счет их сильного упрощения.

**Интернет** происходит от выражения **interconnected networks**, то есть в узком смысле— это глобальное сообщество малых и больших сетей. В более широком смысле — это глобальное информационное пространство, хранящее огромное количество информации на миллионах компьютеров, которые обмениваются данными. В 1969 году, когда был создан **Интернет**, эта сеть объединяла всего лишь четыре компьютера, а сегодня их число измеряется десятками миллионов. Каждый компьютер, подключенный к **Интернет**, — это часть Сети.

Для того чтобы начать с наиболее привычной всем схемы, рассмотрим, как подключается к **Интернет** домашний компьютер, и проследим, по каким каналам путешествует информация, передаваемая и принимаемая нами из Сети. Если вы выходите в **Интернет** с домашнего компьютера, то, скорее всего, используете модемное подключение. Стоит сразу пояснить, что слово «провайдер» происходит от английского слова provide (предоставлять). Провайдером называется компания, которая предоставляет вам [услуги](http://ar-studio.info/service.html) доступа к **Интернет**'.

В принципе, соединение с провайдером может идти по различным каналам: по телефонной линии, по выделенной линии, на основе беспроводной или спутниковой связи, по сети кабельного телевидения или даже по силовым линиям. Чаще всего это так называемое временное (сеансовое) соединение по телефонной линии. Вы набираете один из телефонных номеров, который предоставил вам провайдер, и дозваниваетесь на один из его модемов. После того как вы соединились с вашим **ISP** (Internet Service Prоvider) - провайдером, вы становитесь частью сети данного **ISP**.

Обычно **ISP**-провайдеры — это крупные компании, которые в ряде регионов имеют так называемые точки присутствия (**POP, Point of Presence**), где происходит подключение локальных пользователей. Обычно крупный провайдер имеет точки присутствия (POP) в нескольких крупных городах. В каждом городе находятся аналогичные модемные пулы, на которые звонят локальные клиенты этого ISP в данном городе. Провайдер может арендовать волоконно-оптические линии у телефонной компании для соединения всех своих точек присутствия (POP), а может протянуть свои собственные волоконно-оптические линии. Крупнейшие коммуникационные компании имеют собственные высокопропускные каналы. Очевидно, что все клиенты провайдера **ISP-A** могут взаимодействовать между собой по собственной сети, а все клиенты компании **ISP-B** — по своей, но при отсутствии связи между сетями **ISP-А и ISP-B** клиенты компании «А» и клиенты компании «В» не могут связаться друг с другом. Для реализации данной услуги компании «А» и «В» договариваются подключиться к так называемым точкам доступа (**NAP — Network Access Points**) в разных городах, и трафик между двумя компаниями течет по сетям через **NAP**.

Аналогично организуется подключение к другим магистральным сетям, в результате чего образуется объединение множества сетей высокого уровня. В **Интернете** действуют сотни крупных Интернет-провайдеров, их магистральные сети связаны через **NAP** в различных городах, и миллиарды байтов данных текут по разным сетям через **NAP-узлы**. Если вы пользуетесь Интернетом в офисе, то, скорее всего, вы подключены к локальной сети (**LAN — Local Area Network**). В этом случае рассмотренная нами схема несколько видоизменяется.

Сеть организации обычно отделена от внешнего мира определенной службой защиты информации, которая на нашей схеме условно показана в виде кирпичной стены. Варианты подключения к провайдеру могут быть различными, хотя чаще всего это выделенная линия. При этом у крупных провайдеров могут быть региональные провайдеры, у организаций, подключенных к **Интернету**, — тысячи компьютеров, объединенных в корпоративные сети.

**Как происходит передача информации в Интернет?** Для того чтобы объяснить этот процесс, необходимо ввести несколько понятий и прежде всего рассказать о работе маршрутизаторов. Доставка информации по нужному адресу невозможна без маршрутизаторов, определяющих, по какому маршруту передавать информацию. Маршрутизатор — это устройство, которое работает с несколькими каналами, направляя в выбранный канал очередной блок данных. Таким образом, маршрутизатор выполняет две различные, но взаимосвязанные функции:

он направляет информацию по свободным каналам, предотвращая «закупорку» узких мест в Сети;

он проверяет, что информация следует в нужном направлении. При объединении двух сетей маршрутизатор включается в обе сети, пропуская информацию из одной в другую, и в некоторых случаях осуществляет перевод данных из одного протокола в другой, при этом защищая сети от лишнего трафика.

**Протоколы Интернет.** Перейдем теперь к рассмотрению способов передачи информации в Интернете. Для этого необходимо ввести такое понятие, как протокол. Протокол — это правило передачи информации в Сети. Следует различать два типа протоколов: базовые и прикладные.

Базовые протоколы - отвечают за физическую пересылку сообщений между компьютерами в сети Интернет. Это протоколы **IP и TCP**. Прикладными называют протоколы более высокого уровня, они отвечают за функционирование специализированных служб. Например, протокол **http** служит для передачи гипертекстовых сообщений, протокол ftp — для передачи файлов, **SMTP** — для передачи электронной почты и т.д.

Предположим, имеется некое послание, отправляемое по электронной почте. Передача почты осуществляется по прикладному протоколу **SMTP**, который опирается на протоколы **TCP/IP**. Согласно протоколу **TCP**, отправляемые данные разбиваются на небольшие пакеты фиксированной структуры и длины, маркирующиеся таким образом, чтобы при получении данные можно было бы собрать в правильной последовательности. Обычно длина одного пакета не превышает 1500 байт. Поэтому одно электронное письмо может состоять из нескольких сотен таких пакетов. Малая длина пакета не приводит к блокировке линий связи и не позволяет отдельным пользователям надолго захватывать канал связи.

К каждому полученному **TCP**-пакету протокол IP добавляет информацию, по которой можно определить адреса отправителя и получателя. Для каждого поступающего пакета маршрутизатор, через который проходит какой-либо пакет, по данным **IP**-адреса определяет, кому из ближайших соседей необходимо переслать данный пакет, чтобы он быстрее оказался у получателя. При этом географически самый короткий путь не всегда оказывается оптимальным (быстрый канал на другой континент может быть лучше медленного в соседний город). Очевидно, что скорость и пути прохождения разных пакетов могут быть различными.

Возможно, что пакеты будут путешествовать через разные континенты, с различной скоростью. При этом пакеты, отправленные позже, могут дойти раньше. Независимо от траектории в результате конечного числа пересылок **TCP-пакеты** достигают адресата. Наконец, TCP-модуль адресата собирает и распаковывает IP-конверты, затем распаковывает TCP-конверты и помещает данные в нужной последовательности. Если чего-либо не достает, он требует переслать этот пакет снова. Пакеты не только теряются, но и могут искажаться при передаче из-за наличия помех на линиях связи. TCP решает и эту проблему. В конце концов, информация собирается в нужном порядке и полностью восстанавливается. Таким образом, протокол IP осуществляет перемещение данных в сети, а протокол TCP обеспечивает надежную доставку данных, используя систему кодов, исправляющих ошибки. Причем два сетевых сервера могут одновременно передавать в обе стороны по одной линии множество TCP-пакетов от различных клиентов.

Некоторые начинающие пользователи думают, что связь по **Интернет** похожа на телефонную.  Хочется еще раз подчеркнуть основное различие передачи информации по телефонной сети и по **Интернет**: - когда вы звоните по телефону кому-нибудь в другой регион страны или даже на другой континент, телефонная система устанавливает канал между вашим телефоном и тем, на который вы звоните. Канал может состоять из десятков участков: медные провода, волоконно-оптические линии, беспроводные участки, спутниковая связь и т.д. Эти участки неизменны на протяжении всего сеанса связи. Это означает, что линия между вами и тем, кому вы звоните, постоянна в течение всего разговора, поэтому повреждения на любом участке данной линии, например, обрыв проводов в бурю, способны прервать ваш разговор. При этом, если соединение нормальное, значит выделенная вам часть сети для других уже не доступна. Речь идет о сети с коммутацией каналов.

- **Интернет** же является сетью с коммутацией пакетов, а это со всем другая [история](http://ar-studio.info/content/internet/78.html). Процесс пересылки электронной почты принципиально иной. Как уже было отмечено, Интернет-данные в любой форме (будь то электронное послание, Web-страница или скачиваемый файл) путешествуют в виде группы пакетов. Каждый пакет посылается на место назначения по оптимальному из доступных путей. Поэтому даже если какой-то участок Сети окажется нарушенным, то это не повлияет на доставку пакета, который будет направлен по альтернативному пути. Таким образом, во время доставки данных нет необходимости в фиксированной линии связи между двумя пользователями.

Принцип пакетной коммутации обеспечивает основное преимущество **Интернет** — надежность. Сеть может распределять нагрузку по различным участкам за тысячные доли секунды. Если какой-то участок оборудования сети поврежден, пакет может обойти это место и пройти по другому [пути](http://ar-studio.info/content/other/70.html), обеспечив доставку всего послания.