15. Технология глобальной сети World Wide Web

Газизов Тимур Тальгатович,
к.т.н., доцент кафедры информатики ТГПУ

ИНФОРМАЦИОННЫЕ СЕТИ ИНФОКОММУНИКАЦИОННЫЕ СИСТЕМЫ И СЕТИ

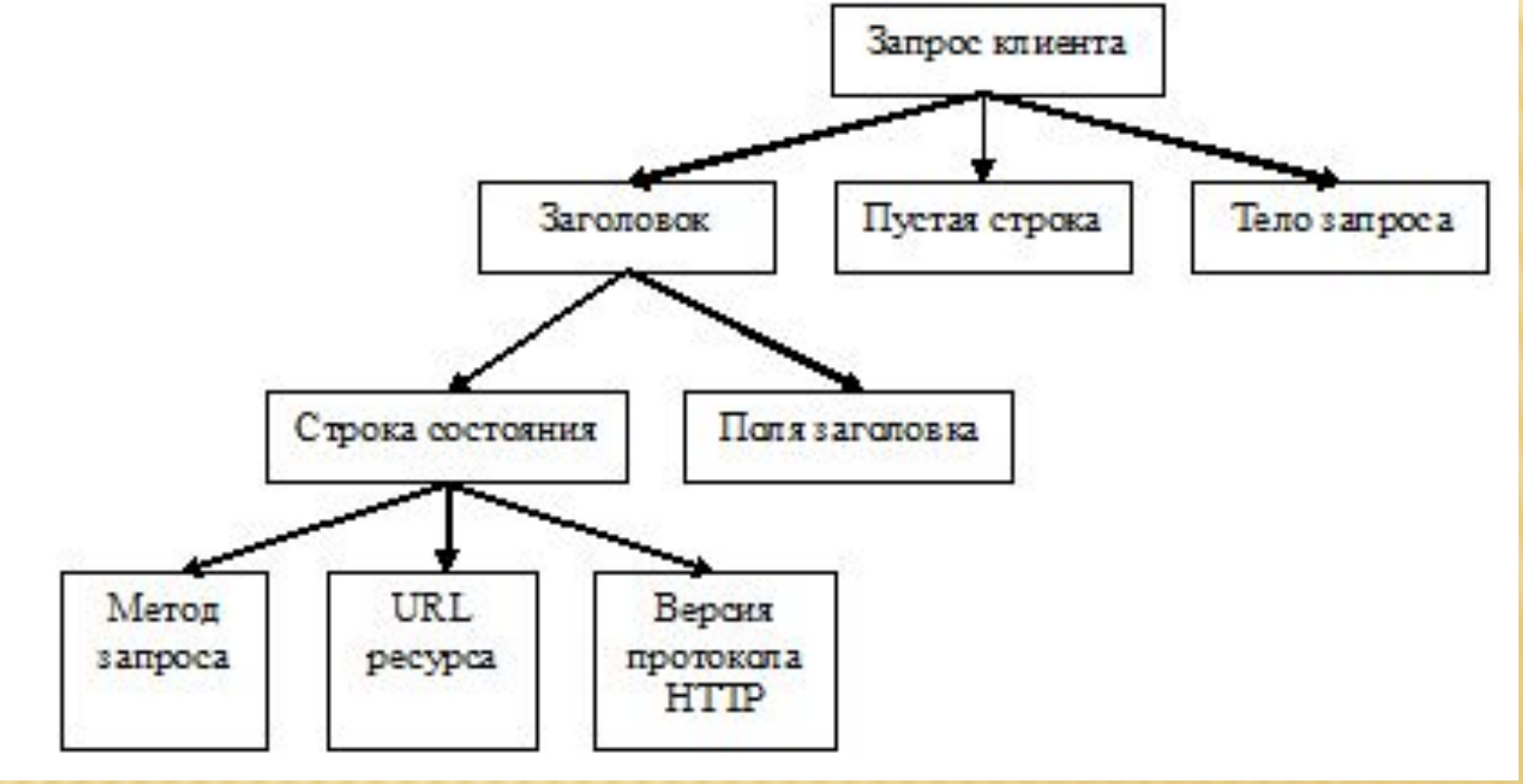
ВВЕДЕНИЕ

- Базовым протоколом сети гипертекстовых ресурсов Веб является протокол HTTP. В его основу положено взаимодействие "клиент-сервер", то есть предполагается, что:
  - Потребитель-клиент инициировав соединение с поставщиком-сервером посылает ему запрос;
  - Поставщик-сервер, получив запрос, производит необходимые действия и возвращает обратно клиенту ответ с результатом.
- При этом возможны два способа организации работы компьютера-клиента:
  - Тонкий клиент - это компьютер-клиент, который переносит все задачи по обработке информации на сервер. Примером тонкого клиента может служить компьютер с браузером, использующийся для работы с веб-приложениями.
  - Толстый клиент, напротив, производит обработку информации независимо от сервера, использует последний в основном лишь для хранения данных.
- Прежде чем перейти к конкретным клиент-серверным веб-технологиям, рассмотрим основные принципы и структуру базового протокола HTTP.

ПРОТОКОЛ HTTP

- HTTP (HyperText Transfer Protocol - RFC 1945, RFC 2616) - протокол прикладного уровня для передачи гипертекста.
- Центральным объектом в HTTP является ресурс, на который указывает URI в запросе клиента. Обычно такими ресурсами являются хранящиеся на сервере файлы. Особенностью протокола HTTP является возможность указать в запросе и ответе способ представления одного и того же ресурса по различным параметрам: формату, кодировке, языку и т. д. Именно благодаря возможности указания способа кодирования сообщения клиент и сервер могут обмениваться двоичными данными, хотя изначально данный протокол предназначен для передачи символьной информации. На первый взгляд это может показаться излишней тратой ресурсов. Действительно, данные в символьном виде занимают больше памяти, сообщения создают дополнительную нагрузку на каналы связи, однако подобный формат имеет много преимуществ. Сообщения, передаваемые по сети, удобочитаемы, и, проанализировав полученные данные, системный администратор может легко найти ошибку и устранить ее. При необходимости роль одного из взаимодействующих приложений может выполнять человек, вручную вводя сообщения в требуемом формате.
- В отличие от многих других протоколов, HTTP является протоколом без памяти. Это означает, что протокол не хранит информацию о предыдущих запросах клиентов и ответах сервера. Компоненты, использующие HTTP, могут самостоятельно осуществлять сохранение информации о состоянии, связанной с последними запросами и ответами. Например, клиентское веб-приложение, посылающее запросы, может отслеживать задержки ответов, а веб-сервер может хранить IP-адреса и заголовки запросов последних клиентов.

ЗАПРОС КЛИЕНТА



ЗАГОЛОВКИ ОТВЕТА ОТ СЕРВЕРА

Таблица 1.3. Поля заголовка ответа веб-сервера.

Имя поля	Описание содержимого
Server	Имя и номер версии сервера
Age	Время в секундах, прошедшее с момента создания ресурса
Allow	Список методов, допустимых для данного ресурса
Content-Language	Языки, которые должен поддерживать клиент для того, чтобы корректно отобразить передаваемый ресурс
Content-Type	MIME-тип данных, содержащихся в теле ответа сервера
Content-Length	Число символов, содержащихся в теле ответа сервера
Last-Modified	Дата и время последнего изменения ресурса
Date	Дата и время, определяющие момент генерации ответа
Expires	Дата и время, определяющие момент, после которого информация, переданная клиенту, считается устаревшей
Location	В этом поле указывается реальное расположение ресурса. Оно используется для перенаправления запроса
Cache-Control	Директивы управления кэшированием. Например, no-cache означает, что данные не должны кэшироваться

ПРИМЕР ОТВЕТА

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.1
X-Powered-By: ASP.NET
Date: Mon, 20 OCT 2008 11:25:56 GMT
Content-Type: text/html
Accept-Ranges: bytes
Last-Modified: Sat, 18 Oct 2008 15:05:44 GMT
ETag: "b66a667f948c92:8a5"
Content-Length: 426

<html>
 <body>
  <form action='http://localhost/Scripts/test.pl'>
   <p>Operand1: <input type='text' name='A'></p>
   <p>Operand2: <input type='text' name='B'></p>
   <p>Operation:<br>
   <select name='op'>
    <option value='+'>+</option>
    <option value='-'>-</option>
    <option value='*'>*</option>
    <option value='/'>/</option>
   <select>
   </p>
   <input type='submit' value='Calculate!'>
  </form>
 </body>
</html>
```

MIME

- Спецификация MIME (Multipurpose Internet Mail Extension - многоцелевое почтовое расширение Internet) первоначально была разработана для того, чтобы обеспечить передачу различных форматов данных в составе электронных писем. Однако применение MIME не исчерпывается электронной почтой. Средства MIME успешно используются в WWW и, по сути, стали неотъемлемой частью этой системы.
- Стандарт MIME разработан как расширяемая спецификация, в которой подразумевается, что число типов данных будет расти по мере развития форм представления данных. Каждый новый тип в обязательном порядке должен быть зарегистрирован в IANA (Internet Assigned Numbers Authority).
- До появления MIME компьютеры, взаимодействующие по протоколу HTTP, обменивались исключительно текстовой информацией. Для передачи изображений, как и для передачи любых других двоичных файлов, приходилось пользоваться протоколом FTP.

ПЕРЕЧЕНЬ ТИПОВ MIME

Таблица 1.4. MIME типы данных.

Тип/подтип	Расширение файла	Описание
application/pdf	.pdf	Документ, предназначенный для обработки Acrobat Reader
application/msexcel	.xls	Документ в формате Microsoft Excel
application/postscript	.ps, .eps	Документ в формате PostScript
application/x-tex	.tex	Документ в формате TeX
application/msword	.doc	Документ в формате Microsoft Word
application/rtf	.rtf	Документ в формате RTF, отображаемый с помощью Microsoft Word
image/gif	.gif	Изображение в формате GIF
image/ jpeg	.jpeg, .jpg,	Изображение в формате JPEG
image/tiff	.tiff, .tif	Изображение в формате TIFF
image/x-xbitmap	.xbm	Изображение в формате XBitmap
text/plain	.txt	ASCII-текст
text/html	. html , . htm	Документ в формате HTML
audio/midi	.midi, .mid	Аудиофайл в формате MIDI
audio/x-wav	.wav	Аудиофайл в формате WAV
message/rfc822		Почтовое сообщение
message/news		Сообщение в группы новостей
video /mpeg	.mpeg, .mpg, .mpe	Видеофрагмент в формате MPEG
video/avi	.avi	Видеофрагмент в формате AVI

ИДЕНТИФИКАЦИЯ РЕСУРСОВ В СЕТИ

- Единообразный идентификатор ресурса URI (Uniform Resource Identifier) представляет собой короткую последовательность символов, идентифицирующую абстрактный или физический ресурс. URI не указывает на то, как получить ресурс, а только идентифицирует его. Это даёт возможность описывать с помощью RDF (Resource Description Framework) ресурсы, которые не могут быть получены через Интернет (имена, названия и т.п.). Самые известные примеры URI - это URL и URN.
  - **URL** (Uniform Resource Locator) - это URI, который, помимо идентификации ресурса, предоставляет ещё и информацию о местонахождении этого ресурса.
  - **URN** (Uniform Resource Name) - это URI, который идентифицирует ресурс в определённом пространстве имён, но, в отличие от URL, URN не указывает на местонахождение этого ресурса.

СТРУКТУРА URL

- URL имеет следующую структуру:
  - <схема>://<логин>:<пароль>@<хост>:<порт>/<URL-путь> где:
- схема - схема обращения к ресурсу (обычно сетевой протокол);
- логин - имя пользователя, используемое для доступа к ресурсу;
- пароль - пароль, ассоциированный с указанным именем пользователя;
- хост - полностью прописанное доменное имя хоста в системе DNS или IP-адрес хоста;
- порт - порт хоста для подключения;
- URL-путь - уточняющая информация о месте нахождения ресурса.