

Лекция 1. Введение

В данном курсе лекций рассматриваются способы создания динамических страниц. Понятия *веб-страница*, *сайт*, *веб-приложение*, *веб-сервер* определены [здесь](#). Рассмотрим подробнее требования, предъявляемые к веб-серверу

Конфигурация веб-сервера зависит от нагрузки, которую он испытывает. Введём обозначения:

- λ - интенсивность входного потока запросов, т.е. среднее количество запросов, поступающих за единицу времени;
- μ - интенсивность выполнения (обслуживания) запросов, т.е. среднее количество запросов, обслуженных веб-сервером за единицу времени;
- ρ - коэффициент загрузки

Коэффициент загрузки ρ определяется как отношение интенсивность входного потока запросов λ к интенсивности их обслуживания μ :

$$\rho = \frac{\lambda}{\mu} .$$

Средняя длина очереди запросов, ожидающих обслуживания на веб-сервере, определяется по формуле

$$L = \frac{\rho^2}{1 - \rho} ;$$

Длина очереди при $\rho \rightarrow 1$ растет неограниченно:

$$\lim_{\rho \rightarrow 1} L = \infty$$

Вместе с длиной очереди неограниченно растёт и время получения ответа на запрос пользователя, т.е. при большой загрузке сервер перестаёт отвечать на запросы. Именно это и происходит при хакерских атаках. На практике для приемлемого времени ожидания необходимо выполнение условия:

$$\rho \leq 0.6 .$$

Эти сведения необходимо учитывать при разработке сайта и выборе веб-сервера для установки сайта.

Сделанные выводы о зависимости времени обслуживания и длины очереди от коэффициента загрузки следуют из теории массового обслуживания. Подробнее на элементарном уровне проблема выбора производительности системы рассматривается в книге "С.П. Митрофанов и др. Организационно-технологическое проектирование ГПС – Ленинград «Машиностроение» ленинградское отделение, 1986. 292 с." в главе 5.

Если веб-сервер работает в локальной сети и обслуживает несколько десятков человек (например, веб-сервер кафедры), то его можно установить на ЭВМ, выполняющую роль сетевого сервера. По мере возрастания нагрузки требуется всё более мощное оборудование

- многопроцессорный сервер с веб-сервером, настроенным на многопоточное обслуживание;
- вычислительный комплекс, состоящий из объединённых в единую систему компьютеров

В последнем случае в системе работает множество экземпляров программных веб-серверов. Заявки на выдачу статических страниц обслуживаются в десятки раз быстрее заявок на генерацию динамических страниц. Веб-сервер, предназначенный для обслуживания любых заявок, очень сложный и относительно медленный, поэтому разработаны "лёгкие" веб-серверы, умеющие обслуживать только заявки на передачу статических страниц. Лёгкие веб-серверы работают гораздо быстрее "тяжёлых". Примеры "тяжёлых" веб-серверов - Apache, IIS (Internet Information Server). К лёгким относятся:

- nginx (произносится как энгинкс или энжин-икс) - очень быстрый и очень популярный веб-сервер, разработанный Игорем Сысоевым;
- lighttpd (также «lighty», «лайти») — веб-сервер, разрабатываемый с расчётом на скорость и защищённость, а также соответствие стандартам.

На рис.1.1 показан веб-кластер, состоящий из одного лёгкого и N тяжёлых веб-серверов. Лёгкий веб-сервер принимает запросы пользователей и, если запрошена статическая страница, отправляет её пользователю. Если запрошена динамическая страница, то запрос передается одному из тяжёлых веб-серверов. Выбор тяжёлого веб-сервера делается по круговой (циклической) дисциплине. Сначала выбирается первый веб-сервер, для выполнения следующего запроса выбирается второй веб-сервер и так далее. После назначения N-ого веб-сервера, снова выбирается первый.

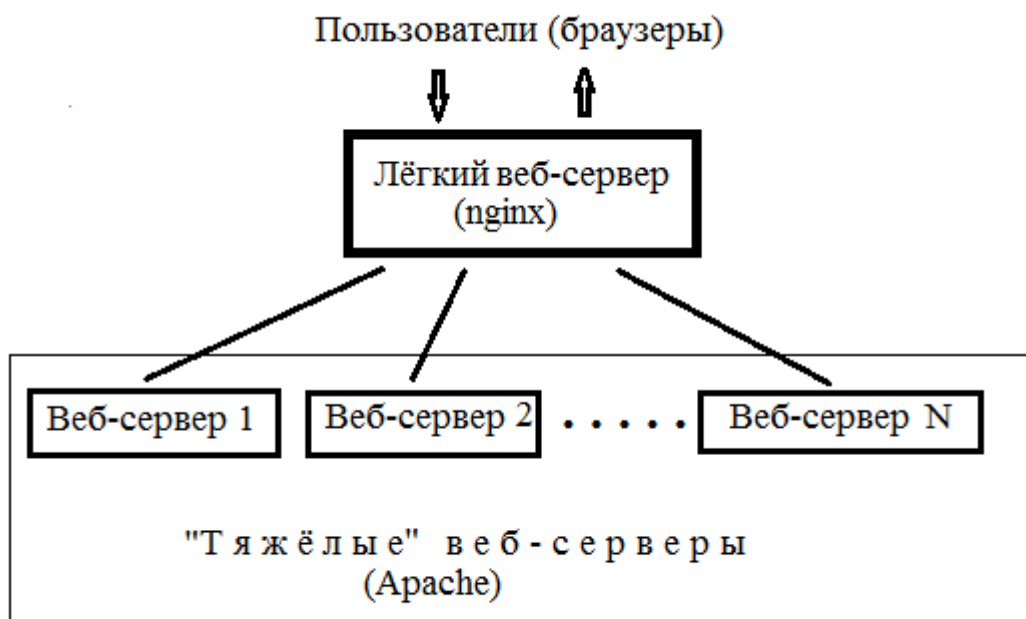


Рис. 1.1. Использование лёгкого веб-сервера

Для формирования динамических страниц требуется целый комплекс, состоящий из веб-сервера и управляемых им транслятов с языков типа PHP, Perl, Python, баз данных и СУБД.

Веб-сервер Apache

Apache -наиболее распространённый среди веб-серверов, кроссплатформенный, состоит из ядра и более, чем 500 модулей. Создан в 1995 году.

Настройка конфигурации Apache производится с помощью текстовых конфигурационных файлов, которые состоят из директив настройки. Рассматриваются три уровня конфигурации:

- конфигурация уровня сервера;
- конфигурация виртуального хоста;
- конфигурация уровня каталога.

Директива имеет синтаксис

имя_директивы значение

В файле директив символ # означает комментарий. Рассмотрим несколько примеров директив.

```
#Каталог с файлами сервера
ServerRoot "C:/server/Apache/Apache2" # Для Windows
ServerRoot "etc/apache2" # Для Ubuntu
#Включение модуля PHP5
```

```

LoadModule php5_module
"/usr/local/php5/php5apache2_2.dll"
# Привязывание к порту
Listen 127.0.0.1:80
#имена страниц, запускаемых по умолчанию в качестве
стартовых,
#если пользователь указывает только имя сайта
<IfModule dir_module>
    DirectoryIndex index.php index.htm index.html
index.shtm index.shtml
</IfModule>

```

Для того чтобы задать для какого-либо сайта или страницы директивы, отличные от заданных для всего веб-сервера, т.е. задать конфигурацию уровня каталога, служит файл со странным именем, начинающимся с точки:

.htaccess

Этот файл нужно поместить в каталог сайта, для которого предназначены директивы. Например, для сайта pricom.loc в каталоге Z:\home\pricom.loc\www помещён следующий файл с именем .htaccess :

```

AddDefaultCharset utf-8
AddLanguage ru-RU .html .htm .css .js .php
DefaultLanguage ru-RU
php_value include_path
"./:/home/pricom.loc/www/inc;/usr/local/lib/php"
AddType text/csv csv

#CSV - значения, разделённые запятыми
php_flag session.use_trans_sid Off
php_flag session.use_only_cookies On
php_flag session.use_cookies On

```

В директиве *php_value include_path* указаны пути, по которым скрипт, написанный на языке PHP, должен искать файлы по умолчанию.

Виртуальные хосты и ДЕНВЕР

Механизм виртуальных хостов позволяет обслуживать на одном IP-адресе, т.е. на одной ЭВМ, множество сайтов с разными доменными именами. Протокол TCP/IP позволяет использовать одну ЭВМ одновременно и как веб-сервер и как клиента с браузером, образуя петлю между веб-сервером и браузером. Эти два обстоятельства используются для отладки и сопровождения множества сайтов на ЭВМ, не подключённой к Интернет. Для образования петли используется локальный IP-адрес 127.0.0.1.

Сайт должен быть описан в одном из файлов конфигурации веб-сервера. Ниже приведён фрагмент файла конфигурации с описанием сайтов с доменными именами *localhost* и *KafI5.loc*.

```
# Host /home/localhost/www (2):
```

```

Listen 127.0.0.1:443
NameVirtualHost 127.0.0.1:443
<VirtualHost 127.0.0.1:443>
    SSLEngine on
    DocumentRoot "Z:/home/localhost/www"
    ServerName "localhost"
    ServerAlias "localhost" "www.localhost"
    ScriptAlias /cgi/ "/home/localhost/cgi/"
    ScriptAlias /cgi-bin/ "/home/localhost/cgi-bin/"
</VirtualHost>
# Host /home/KafI5.loc/www :
#Listen 127.0.0.1:80
#NameVirtualHost 127.0.0.1:80
<VirtualHost 127.0.0.1:80>
    DocumentRoot "Z:/home/KafI5.loc/www"
    ServerName "KafI5.loc"
    ServerAlias "KafI5.loc" "www.KafI5.loc"
    ScriptAlias /cgi/ "/home/KafI5.loc/cgi/"
    ScriptAlias /cgi-bin/ "/home/KafI5.loc/cgi-bin/"
</VirtualHost>

```

Директивы `<VirtualHost ...>` и `</VirtualHost>` используются, чтобы выделить группу директив, которые применяются только к данному виртуальному хосту. Приведённый фрагмент взят из файла `vhosts.conf`, сформированного автоматически системой ДЕНВЕР.

Система ДЕНВЕР (Джентльментский набор веб-разработчика) разработана Дмитрием Котеровым, для того чтобы избавить пользователя от необходимости описывать вручную разрабатываемый сайт в файле конфигурации. ДЕНВЕР работает под управлением операционных систем семейства WINDOWS. В состав ДЕНВЕР входят несколько переделанный веб-сервер Apache, СУБД MySQL и встроенный интерпретатор языка PHP. Дополнительно можно подсоединить транслятор Perl.

Для создания сайта в ДЕНВЕР необходимо в каталоге ДЕНВЕРа `C:\WebServers\home` создать каталог с именем, совпадающим с именем сайта, например, `kam.loc`.

ДЕНВЕР может быть установлен на любом локальном диске. При запуске ДЕНВЕР создаётся виртуальный диск `Z:`, соответствующий каталогу *имя_локального_диска:\WebServers* (например, `C:\WebServers`).

В созданный каталог нужно поместить каталог `www`. В каталоге `www` будут храниться все файлы с HTML-документами и со скриптами на PHP. После создания каталогов сайта нужно запустить (или перезапустить) ДЕНВЕР. При запуске ДЕНВЕР создаёт файл `vhosts.conf`, читает каталог `home` и записывает в `vhosts.conf` данные о существующих сайтах. Например, обнаружив в каталоге `home` подкаталог `anketaWin.loc`, ДЕНВЕР добавит в файл `vhosts.conf` следующий фрагмент:

```

# Host /home/anketaWin.loc/www (3) :

```

```
#Listen 127.0.0.1:80
#NameVirtualHost 127.0.0.1:80
<VirtualHost 127.0.0.1:80>
    DocumentRoot "Z:/home/anketaWin.loc/www"
    ServerName "anketaWin.loc"
    ServerAlias "anketaWin.loc" "www.anketaWin.loc"
    ScriptAlias /cgi/ "/home/anketaWin.loc/cgi/"
    ScriptAlias /cgi-bin/ "/home/anketaWin.loc/cgi-
bin/"
</VirtualHost>
```

После занесения данных о сайте в `vhosts.conf` можно запрашивать этот сайт через браузер, задавая доменное имя `anketaWin.loc`.

Стек протоколов TCP/IP

Последовательность передачи сообщения в сети Internet определяется набором правил, который называется *стек протоколов TCP/IP*. На рис 1.2 представлена схема передачи сообщения, содержащего HTML-страницу.

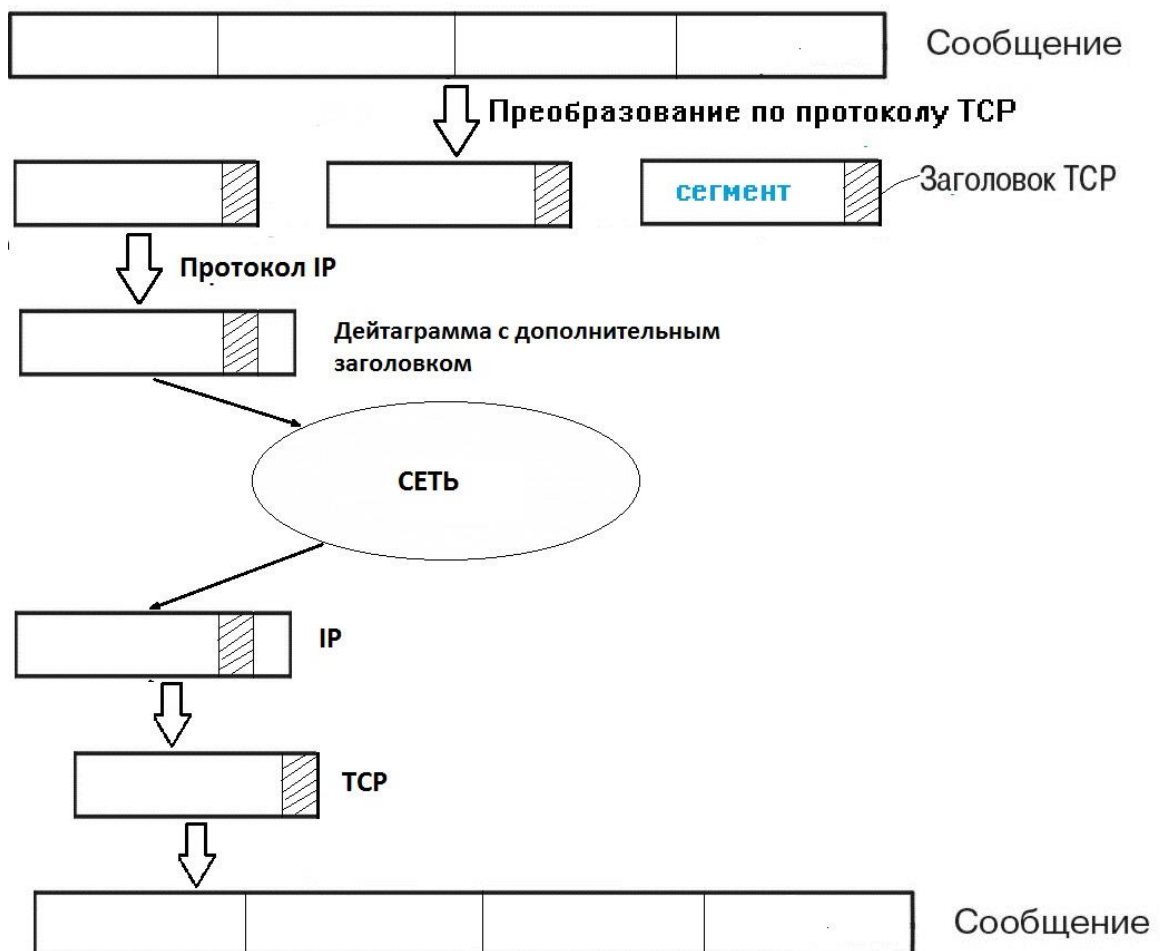


Рис. 1.2. Схема передачи сообщения по протоколу TCP/IP

Стек протоколов TCP/IP включает в себя четыре уровня:

- Прикладной уровень (Application Layer),
- Транспортный уровень (Transport Layer),
- Межсетевой уровень (Сетевой уровень[7]) (Internet Layer),
- Канальный уровень (Network Access Layer).

На каждом уровне располагается один или несколько протоколов (табл. 1.1)

Табл. 1.1. Распределение протоколов по уровням модели TCP/IP

Прикладной (Application layer)	напр., HTTP, FTP, SMTP, POP3, IMAP
Транспортный (Transport Layer)	TCP
Сетевой(Internet Layer)	IP
Канальный (Network Access Layer)	Ethernet, IEEE 802.11 WLAN, SLIP, Token Ring, ATM и MPLS, физическая среда и принципы кодирования информации, T1, E1

Для переход от протокола TCP к протоколам прикладного уровня служит, хранящийся в заголовке TCP (рис. 1.3) номер порта.



Рис. 1.3. Заголовок TCP.

Соответствие между номерами портов и протоколами прикладного уровня приведено в табл. 1.2.

Табл. 1.2. Соответствие между портами и протоколами прикладного уровня

Порт	Протокол прикладного уровня
80	HTTP (HyperText Transfer Protocol) для передачи сообщений во всемирной паутине
20	FTP (File Transfer Protocol) для данных (файлов)
21	FTP для команд
25	SMTP (Simple MAIL Transfer Protocol) для отправки почты
110	POP3 для приёма почты (сообщения хранятся у клиента)
143	IMAP для приёма почты (сообщения хранятся на сервере)

На рис. 1.4. представлена структура IP-заголовка дейтаграммы.

Октет	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Версия			IHL			Тип обслуживания								Длина пакета																	
4	Идентификатор															Флаги		Смещение фрагмента														
8	Время жизни							Протокол							Контрольная сумма заголовка																	
12	IP-адрес отправителя																															
16	IP-адрес получателя																															
20	Параметры от 0-я до 10-и 32-х битовых слов																															
	Данные																															

Рис. 1.3. Заголовок IP.

Протокол HTTP

HTTP (англ. HyperText Transfer Protocol — «протокол передачи гипертекста») — протокол прикладного уровня передачи данных, изначально — в виде гипертекстовых документов в формате HTML, в настоящее время используется для передачи произвольных данных.

Основным объектом манипуляции в HTTP является ресурс, на который указывает URI (Uniform Resource Identifier) в запросе клиента. Обычно такими ресурсами являются хранящиеся на сервере файлы, но ими могут быть логические объекты или что-то абстрактное. 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. (в частности, для этого используется HTTP-заголовок). Именно благодаря возможности указания способа кодирования сообщения клиент и сервер могут обмениваться двоичными данными, хотя данный протокол является текстовым.

Всё программное обеспечение для работы с протоколом HTTP разделяется на три большие категории:

- Серверы как основные поставщики услуг хранения и обработки информации (обработка запросов);
- Клиенты — конечные потребители услуг сервера (отправка запроса);
- Прокси (посредники) для выполнения транспортных служб.

HTTP — протокол прикладного уровня; аналогичными ему являются FTP и SMTP. Обмен сообщениями идёт по обыкновенной схеме «запрос-ответ». Для идентификации ресурсов HTTP использует глобальные URI. Каждое HTTP-сообщение состоит из трёх частей, которые передаются в указанном порядке:

- Стартовая строка (англ. Starting line) — определяет тип сообщения;
- Заголовки (англ. Headers) — характеризуют тело сообщения, параметры передачи и прочие сведения;
- Тело сообщения (англ. Message Body) — непосредственно данные сообщения. Обязательно должно отделяться от заголовков пустой строкой.

Заголовки HTTP (англ. HTTP Headers) — это строки в HTTP-сообщении, содержащие разделённую двоеточием пару имя-значение. Формат заголовков соответствует общему формату заголовков текстовых сетевых сообщений ARPA (см. RFC 822). Заголовки должны отделяться от тела сообщения хотя бы одной пустой строкой.

Все заголовки разделяются на четыре основных группы:

1. General Headers (рус. Основные заголовки) — должны включаться в любое сообщение клиента и сервера.
2. Request Headers (рус. Заголовки запроса) — используются только в запросах клиента.
3. Response Headers (рус. Заголовки ответа) — только для ответов от сервера.
4. Entity Headers (рус. Заголовки сущности) — сопровождают каждую сущность сообщения.

Именно в таком порядке рекомендуется посылать заголовки получателю.

Пример. Чтобы получить страницу с URL <https://ru.wikipedia.org/wiki/HTTP>, нужно послать серверу такое сообщение

```
GET /wiki/HTTP HTTP/1.0
Host: ru.wikipedia.org
```

Здесь

```
GET /wiki/HTTP HTTP/1.0  - стартовая строка запроса
GET                     - метод
```

/wiki/HTTP	- URI
HTTP/1.0	- версия HTML
Host: ru.wikipedia.org	- заголовок
Host	- имя заголовка
ru.wikipedia.org	- значение

В общем виде стартовая строка запроса имеет вид

GET URI — для версии протокола 0.9;
 Метод URI HTTP/Версия — для остальных версий.

Стартовая строка ответа сервера на предыдущий запрос может выглядеть так:

HTTP/1.0 200 OK

В PHP функция `get_headers` возвращает все заголовки из ответа сервера на HTTP-запрос

Пример использования `get_headers()`

```
<?php
$url = 'http://www.example.com';

print_r(get_headers($url));
?>
```

Результат выполнения данного примера будет примерно такой:

```
Array
(
    [0] => HTTP/1.1 200 OK
    [1] => Date: Sat, 29 May 2004 12:28:13 GMT
    [2] => Server: Apache/1.3.27 (Unix) (Red-
Hat/Linux)
    [3] => Last-Modified: Wed, 08 Jan 2003 23:11:55
GMT
    [4] => ETag: "3f80f-1b6-3e1cb03b"
    [5] => Accept-Ranges: bytes
    [6] => Content-Length: 438
    [7] => Connection: close
    [8] => Content-Type: text/html
)
```