ВЕБ-ПРОГРАММИРОВАНИЕ

1. WWW

1.1. **Основные определения.**

Интернет — глобальная компьютерная сеть, построенная на базе стека протоколов TCP/IP и объединяющая локальные сети и отдельные компьютеры в общее информационное пространство. Сеть Internet выросла из сети **ARPANET** (1969, Калифорнийский университет), которая использовала протокол NCP, а в 1983 перешла на TCP/IP.

TCP/IP — сетевая модель передачи данных в сети Интернет. Состоит из 4-х уровней: канальный, межсетевой, транспортный и прикладной.

(1) **IP** (Internet Protocol) — межсетевой протокол, который отвечает за передачу отдельных датаграмм. Сам по себе IP не гарантирует надежной доставки пакета до адресата.

(2) **TCP** (Transmission Control Protocol) отвечает за разбивку сообщения на датаграммы. Гарантирует целостность передаваемых данных и уведомление отправителя о результатах передачи.

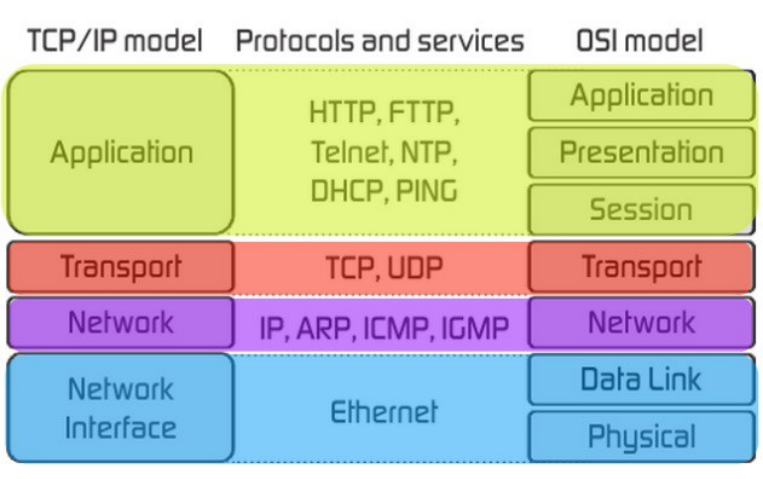


Рис. 1. Уровни стека протоколов TCP/IP (и связь с сетевой моделью OSI)

World Wide Web (www, Всемирная Паутина) — распределенная информационная система, предоставляющая доступ к гипертекстовым документам по протоколу **HTTP**. Создана в 1989 г.

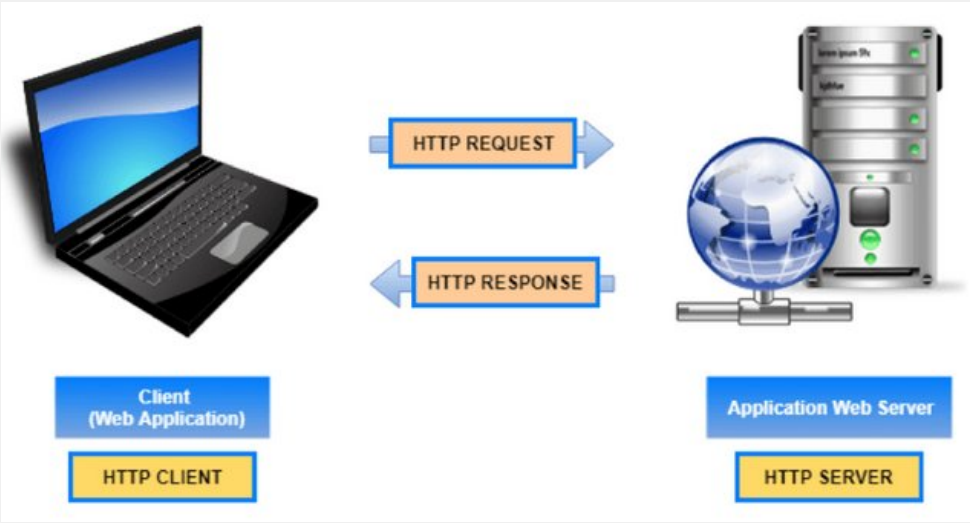


Рис. 2. Архитектура WWW

Взаимодействие браузера и веб-сервера:

- Браузер делает запрос к веб-серверу.
- Веб-сервер в ответ отправляет требуемый ресурс.

Гипертекст — электронные страницы, связанные между собой гиперссылками.

Веб-браузер (пользовательский клиент) — программа, способная обрабатывать гипертекстовую разметку и отображать содержимое веб-страниц.

Веб-сервер — программа, запущенная нас сетевом компьютере, и ожидающая запросы по протоколу HTTP. Веб-сервером также называется сам этот компьютер.

Четыры основные составляющие сервиса WWW:

- **URL/URI** — унифицированный способ адресации и идентификации сетевых ресурсов;
- **HTML** — язык гипертекстовой разметки веб-документов;
- **HTTP** — протокол передачи гипертекста;
- **CGI** (+ WSGI, FastCGI) — общий шлюзовый интерфейс, представляющий доступ к серверным приложениям.

1.2. **URL.**

Общий вид URL-адреса:

`<схема>://<логин>:<пароль>@<хост>:<порт>/<полный-путь-к-ресурсу>`

Примеры:

```
ftp://user:pass@ftp.example.org:81/articles/script.php
http://www.ibm.com/developerworks/news/
```

1.3. **HTTP.**

HTTP — протокол передачи гипертекста между веб-сервером и веб-браузером по схеме «запрос-ответ».

Запросы записываются в виде ASCII-команд.

Пример *запроса*:

```
GET /home.html HTTP/1.1
Host: www.yoursite.com
```

Пример *ответа*:

```
HTTP/1.1 200 OK
Date: Sun, 28 Jul 2013 15:37:37 GMT
Server: Apache
Last-Modified: Sun, 07 Jul 2013 06:13:43 GMT
Transfer-Encoding: chunked
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
Webpage Content
```


curl — утилита для передачи и получения данных по различным протоколам:

`curl -IL http://google.com`

(-I выдает только заголовок HEAD http-ответа, -L — follow redirects)

HTTP-сообщение (запрос либо ответ) состоит из 3-х частей:

(1) Стартовая строка (Starting line) — определяет тип сообщения;
(2) Заголовки (Headers) — различные сведения о теле сообщения (не обязательный элемент);
(3) Тело сообщения (Message Body)

Стартовая строка запроса: `Метод URI HTTP/Версия_протокола`
Например, `GET /vsu.by/index.html HTTP/1.1`

Стартовая строка ответа: `HTTP/Версия КодСостояния [Пояснение]`
Например, `HTTP/1.1 200 Ok`

Методы протокола HTTP:

- `GET` — используется для запроса содержимого указанного ресурса. Параметры выполнения запроса передаются в URI-адресе после символа «?» в виде пар «`параметр=значение`», разделенных символом «&»:
  `http://vk.com/profile.php?id=12345678`
  `http://mysite.com/register.php?name=Иван&id=01&passwd=123456`
- `POST` — в отличие от GET-запросов POST позволяет передавать большие объемы данных в бинарном виде, например, для загрузки файлов на сервер.

- `HEAD` — аналогичен методу GET, за исключением отсутствия тела в ответе сервера (используется для извлечения метаданных): `curl -I`.
- `OPTIONS` — для проверки работоспособности сервера.
- `PUT` — применяется для загрузки содержимого запроса на указанный в запросе URI.
- `PATCH, DELETE, TRACE, LINK, UNLINK`

Коды состояния в ответе веб-сервера информируют клиента о результатах выполнения запроса. Представляют собой трезначные числа:

- `1xx` — Informational — информационные, например «102 Processing (Идет обработка)».
- `2xx` — Success — успешное принятие и обработка запроса клиента. Например, 200 OK (Успешно), 202 Accepted (Принято).
- `3xx` — Redirection (Перенаправление) — информируют клиента о необходимости произвести автоматический переход («редирект»).
- `4xx` — Client Error — ошибки со стороны клиента: «404 Not Found (Не найдено)»;
- `5xx` — Server Error — ошибки со стороны сервера: «504 Gateway Timeout (Шлюз не отвечает)».

HTTPS (*HyperText Transfer Protocol Secure*) — расширение протокола HTTP для поддержки шифрования в целях повышения безопасности. Данные передаются поверх криптографических протоколов **SSL** или **TLS**.

1.4. **HTML.**

HTML (HyperText Markup Language) — стандартный язык гипертекствой разметки документов во Всемирной паутине.

HTML-разметка обрабатывается браузером и преобразуется в человекочитаемый документ.

Структура HTML-документа:

(1) Описание (секция `DOCTYPE`) — указывается тип документа и версия HTML;

(2) Текст документа — находится внутри тега `<HTML>` и включает две части:

- Заголовок — `<HEAD>`;
- Тело документа — `<BODY>`;


```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.01//EN">
<HTML>
   <HEAD> <!-- заголовок (может отсутствовать) -->
     <TITLE> Название документа </TITLE>
     <META http-equiv="content-type" content="text/html; charset=UTF-8">
   </HEAD>
   <BODY>
      <!-- тело документа: содержит html-теги и текст -->
   </BODY>
</HTML>
```

HTML-тег — управляющая символьная последовательность, которая задает способ отображения гипертекстовой информации:

`<имя_тега [атрибуты]> данные </имя_тега>`

Часто используемые теги:

- `<html>...</html>` — контейнер гипертекста
- `<head>...</head>` — контейнер заголовка документа
- `<title>...</title>` — название документа (отображается в заголовке окна браузера)
- `<body>...</body>` — контейнер тела документа
- `<div>...</div>` — контейнер общего назначения (структурный блок)
- `<hN>...</hN>` — заголовок N-ного уровня (N = 1...6)
- `<p>...</p>` — основной текст
- `<a>...</a>` — гиперссылка
- `<ol>...</ol>` — нумерованный список
- `<ul>...</ul>` — маркированный список
- `<li>...</li>` — элемент списка
- `<table>...</table>` — контейнер таблицы
- `<tr>...</tr>` — строка таблицы
- `<td>...</td>` — ячейка таблицы
- `<img>...</img>` — изображение
- `<form>...</form>` — форма

- `<i>...</i>` — отображение текста курсивом
- `<b>...</b>` — отображение текста полужирным шрифтом
- `<em>...</em>` — выделение (курсивом)
- `<strong>...</strong>` — усиление (полужирным шрифтом)
- `<br>` — принудительный разрыв строки

Пример гиперссылки: `<a href="http://example.com/page1.html">Пример</a>`

Атрибуты тега уточняют его представление:

`<тег атрибут="значение">...</тег>`

Часто используемые атрибуты:

- `<html>...</html>` — контейнер гипертекста
- `style="описание_стилей"` — локальные стили
- `src="адрес"` — адрес (URI) источника данных (например картинки или скрипта)
- `align="left/center/right/justify"` — выравнивание, по умолчанию left (по левому краю)
- `width="число"` — ширина элемента (в пикселях, пиках, поинтах и др.)
- `height="число"` — высота элемента (в пикселях, пиках, поинтах и др.)
- `href="адрес"` — гиперссылка, адрес (URI) на который будет выполнен переход
- `name="имя"` — имя элемента
- `id="идентификатор"` — уникальный (в пределах веб-страницы) идентификатор элемента
- `size="число"` — размер элемента

- `class="имя_класса"` — имя класса во встроенной или связанной таблице стилей
- `title="строка"` — название элемента
- `alt="строка"` — альтернативный текст

1.5. CSS.

Каскадные таблицы стилей (Cascading Style Sheets, CSS) — это стандарт, определяющий представление данных в браузере.

Таблица стилей представляет собой набор правил, задающих значения свойств селекторов:
`селектор[, селектор[, ...]] {свойство: значение;}`

- *Селектор* — это элемент, к которому будут применяться назначаемые стили. Это может быть тег, класс или идентификатор;
- *Свойство* — определяет одну или несколько характеристик селектора. Свойства задают формат отображения селектора: отступы, шрифты, выравнивание, размеры и т.д.
- *Значения* — это константы, определяющие свойство селектора.

Примеры

```
h1, h2, h3 {
   color: blue; /* Синий цвет для заголовков  */
}
table, img {border: none;} /* Таблицы и изображения выводить без обрамления */
```

Три способа **применения** таблицы стилей к документу HTML:

(1) *Встраивание (Inline)* — стиль применяется к заданному тегу.

(2) *Внедрение (Embedded)* — управляет стилями страницы целиком.

(3) *Связывание (Linked или External)* — описание стилей выносится во внешний файл

Встроенные стили.

```
<p style="font: 12pt Courier">Это текст с кеглем 12 точек и гарнитурой Courier</P>
```

Внедренные стили.

```
<head>
   ...
   <style>
      /* правила CSS */
   </style>
   ...
</head>
```

Связанные таблицы стилей.

```
<head>
   <link rel=stylesheet href="sample.css" type="text/css">
</head>
```

1.6. **CGI.**

CGI (*Common Gateway Interface*) — механизм доступа к внешним программам на стороне веб-сервера для расширения его возможностей. Например, запуск скриптов Perl, PHP, Python или просто exe-файлов.

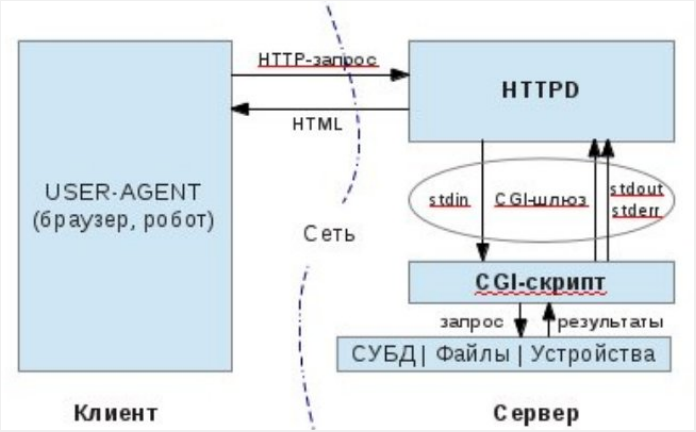


Рис. 3. Общий принцип работы CGI

httpd (*Hyper Text Transfer Protocol Daemon*) — название ПО, выполняющего роль веб-сервера в системе Linux (т.е. запущенное в режиме демона).

Алгоритм работы по протоколу CGI:

(1) Клиент запрашивает CGI-приложение по его URI.
(2) Веб-сервер принимает запрос и устанавливает необходимые переменные окружения.
(3) Веб-сервер перенаправляет запросы через стандартный поток ввода (stdin) на вход вызываемой программы.
(4) CGI-приложение выполняет формирует результаты в виде HTML-документа.
(5) Сформированный гипертекст возвращается веб-серверу через стандартный поток вывода (stdout)
(6) Веб-сервер передает результаты запроса клиенту.

CGI — первая и уже устаревшая технология создания интерактивных веб-страниц. Основной недостаток — CGI-программа должна всякий раз загружаться и выгружаться.

В настоящее время используются WSGI (python) или FastCGI (perl).

WSGI (*Web Server Gateway Interface*) — стандарт взаимодействия между Python-программой, выполняющейся на стороне сервера, и самим веб-сервером. Например, в Apache это реализуется через модуль `mod_wsgi`.

Наиболее популярные веб-сервера:

- **Apache** (1995) — написанный на Си веб-сервер с открытым исходным кодом, славящийся своей гибкостью.
- **nginx** (2004) — HTTP-сервер, совмещенный с кэширующим прокси-сервером.
- **IIS** (Internet Information Services)
- **GWS** (Google Web Server)

В больших проектах Apache и Nginx могут использоваться в связке, разделяя функции, а именно:

(1) Nginx — принимает запросы пользователей и выдает *статический* контент — изображения, файлы, js-скрипты;

(2) Apache — выполняет *динамический* контент на стороне бэкэнда, т.е. запускает скрипты PHP, Python, Ruby, Perl, ASP, Tcl и т.п.

Веб-сервер Apache состоит из ядра (apache core) и дополнительных модулей (устанавливаются отдельно):

- `mod_access` — отвечает за доступ к каталогам и файлам веб-сервера.
- `mod_alias` — отвечает за переадресацию и использование псевдонимов (алиасов).
- `mod_auth` — отвечают за аутентификацию пользователей.
- `mod_autoindex` — для автоматической генерации индексных файлов.
- `mod_status` позволяет администратору контролировать работу веб-сервера.
- `mod_proxy` — позволяет использовать Apache в качестве прямого или обратного прокси-сервера.
- `mod_rewrite` — отвечает за перенаправление запросов. Например, запрос `http://example.com/news/2009/05/03` может быть преобразован к виду: `http://example.com/news.php?date=20090503`
- `mod_perl` — загружаемый (а не вызываемый, как в CGI) интерпретатор Perl.
- `mod_php5` — для PHP5;
- `mod_wsgi` — для Python;
- `mod_python` — устаревший модуль (использует CGI-протокол)

2.1. **Конфигурирование Apache2.**

Два режимы доступа к сайту:

- на основе IP-адресов;
- *виртуальный хостинг* — привязка нескольких доменов к одной машине (с одним IP-адресом)

3. Динамический веб-сайт

Динамический (интерактивный) веб-сайт — сайт с динамическим контентом, который может формироваться как на стороне сервера, так и на стороне клиента.

- *на стороне сервера* — генерация конетнта с помощью скриптов на языках PHP, Perl, Ruby, Java, ASP.NET и др.
- *на стороне клиента* — используется динамический HTML (DHTML):
  `DHTML = HTML + CSS + DOM + JavaScript`.
  (DOM — Document Object Model)

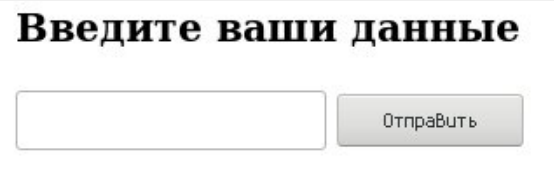
Для передачи данных от пользователя к веб-серверу по HTTP используются **HTML-формы**.

Формы создаются с помощью тега `<FORM>...</FORM>` и могут включать следующие элементы управления:

- `<INPUT>` — поле для ввода однострочного текста;
- `<TEXTAREA>` — поле для ввода многострочного текста;
- `<SELECT>` — выбор из нескольких вариантов;
- `<BUTTON>` — для создания кнопки с расширенными возможностями;

Пример создания формы:

```
<h3>Введите ваши данные</h3>
<form method="post" action="handler.php">
   <input type="text" name="username">
   <input type="submit" value="Отправить">
</form>
```



Параметры тега FORM:

- `action` — задается URL-адрес CGI-программы, которая будет обрабатывать введенную пользователем информацию.
- `method` — определяет способ передачи данных — принимает значения `POST` или `GET` (по умолчанию).