HTTP

Программирование на стороне сервера:

- Протокол HTTP.
- CGI.
- Передача параметров серверу.
- Запоминание состояния.
- Меры безопасности.
- CGI и базы данных

Протокол HTTP.

Исходно WWW состояла из HTML, URL и HTTP.

- HTML - язык форматирования, используемый для представления содержания в Web.
- URL - это адрес, используемый для получения содержимого в формате HTML (или каком-либо ином) с веб-сервера.
- HTTP - это язык, который понятен веб-серверу и позволяет клиентам запрашивать у сервера документы.

Протокол HTTP.

- программа-клиент устанавливает TCP-соединение с сервером (стандартный номер порта - 80) и выдает ему HTTP-запрос. Сервер обрабатывает этот запрос и выдает HTTP-ответ клиенту.
- Необходимо знать структуру HTTP-запроса и ответа.

Протокол HTTP.

Структура HTTP-запроса.

- HTTP-запрос состоит из заголовка запроса и тела запроса, разделенных пустой строкой.
- Тело запроса может отсутствовать.
- Заголовок запроса состоит из главной (первой) строки запроса и последующих строк, уточняющих запрос в главной строке.
- Последующие строки также могут отсутствовать.

HTTP-запрос

- Запрос в главной строке состоит из трех частей, разделенных пробелами:

1. *Метод* (иначе говоря, команда HTTP):
   - GET - запрос документа. Наиболее часто употребляемый метод.
   - HEAD - запрос заголовка документа.
   - POST - для передачи данных CGI-скриптам. Сами данные следуют в последующих строках запроса в виде параметров.
   - PUT - разместить документ на сервере. Используется редко. Запрос с этим методом имеет тело, в котором передается сам документ.

HTTP-запрос

2. *Ресурс* - это путь к определенному файлу на сервере, который клиент хочет получить (или разместить - для метода PUT). Если ресурс - просто какой-либо файл для считывания, сервер должен по этому запросу выдать его в теле ответа. Если же это путь к какому-либо CGI-скрипту, то сервер запускает скрипт и возвращает результат его выполнения.
3. *Версия протокола* - версия протокола HTTP, с которой работает клиентская программа.

HTTP-запрос

- Таким образом, простейший HTTP-запрос может выглядеть следующим образом:

```
GET / HTTP/1.0
```

- запрашивается корневой файл из корневой директории web-сервера.

HTTP-запрос

- Строки после главной строки запроса имеют следующий формат: *Параметр: значение*.
- Таким образом задаются параметры запроса. Это является необязательным, все строки после главной строки запроса могут отсутствовать; в этом случае сервер принимает их значение по умолчанию или по результатам предыдущего запроса (при работе в режиме Keep-Alive).

HTTP-запрос

- наиболее употребительные параметры HTTP-запроса:
  - Connection (соединение)- может принимать значения Keep-Alive и close.
    - Keep-Alive ("оставить в живых") - после выдачи данного документа соединение с сервером не разрывается, и можно выдавать еще запросы. Позволяет за одно соединение с сервером "скачать" html-страницу и рисунки к ней. Будучи установленным, режим сохраняется до первой ошибки или до явного указания в очередном запросе Connection: close.
    - close ("закрыть") - соединение закрывается после ответа на данный запрос.

HTTP-запрос

- User-Agent - "кодовое обозначение" браузера, например: *Mozilla/4.0 (compatible; MSIE 5.0; Windows 95; DigExt)*
- Accept - список поддерживаемых браузером типов содержимого в порядке их предпочтения, например, для IE5*: Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/msword, */**.
- Referer - URL, с которого перешли на этот ресурс.
- Host - имя хоста, с которого запрашивается ресурс.
- Accept-Language - поддерживаемый язык.

HTTP-ответ

- имеет заголовок и тело, разделенные пустой строкой. Заголовок состоит из основной строки и строк параметров. Основная строка запроса состоит из 3-х полей, разделенных пробелами:
  - Версия протокола - аналогичен соответствующему параметру запроса.
  - Код ошибки - кодовое обозначение "успешности" выполнения запроса. Код 200 означает "все нормально" (OK).
  - Словесное описание ошибки - "расшифровка" предыдущего кода. Например, для 200 это OK, для 500 - Internal Server Error.

HTTP-ответ

- Наиболее употребительные параметры:
  - Connection - аналогичен параметру запроса.
  - Content-Type ("тип содержимого") - обозначение типа содержимого ответа. Некоторые типы содержимого:
    - text/html - текст в формате HTML (веб-страница);
    - text/plain - простой текст (аналогичен "блокнотовскому");
    - image/jpeg - картинка в формате JPEG;
    - application/octet-stream - поток "октетов" (байт) для записи на диск.
  - Content-Length ("длина содержимого") - длина содержимого ответа в байтах.
  - Last-Modified ("Модифицирован в последний раз") - дата последнего изменения документа.

Спецификация CGI

- CGI (Common Gateway Interface - общий шлюзовой интерфейс) - это набор правил, согласно которым программы на сервере могут через веб-сервер посылать данные клиентам.
- Спецификация CGI ввела изменения в HTML и HTTP, добавив формы и параметры запроса - данные для этой CGI-программы.

Спецификация CGI

- Распространенные приложения CGI включают в себя:
  - Динамические сайты - генерируются одной CGI-программой.
  - Поисковые механизмы, находящие документы с заданными пользователем словами.
  - Гостевые книги и доски объявлений, в которые пользователи могут добавлять свои сообщения.
  - Бланки заказов, анкеты.
  - Извлечение информации из размещенной на сервере базы данных.

Спецификация CGI

- четыре способа, которыми CGI передает данные между CGI-программой и веб-сервером:
  - Переменные окружения.
  - Командная строка.
  - Стандартное устройство ввода.
  - Стандартное устройство вывода.

Переменные окружения

- Переменные окружения - в спецификации официально определены семнадцать переменных, но неофициально используется значительно больше - с помощью HTTP_mechanism.
- Обращение к переменной окружения FOO:
  - $FOO - в сценарии командного процессора;
  - $ENV{'FOO'} - в Perl;
  - getenv("FOO") - в C;
  - данные, возвращаемые клиентом в заголовке запроса, присваиваются переменным вида HTTP_FOO, где FOO - имя заголовка.

Переменные окружения

CONTENT_LENGTH	Длина данных, переданных методами POST или PUT, в байтах
CONTENT_TYPE	Тип MIME данных, присоединенных с помощью методов POST или PUT.
GATEWAY_INTERFACE	Номер версии спецификации CGI, поддерживаемой сервером.
PATH_INFO	Дополнительная информация пути, переданная клиентом.
PATH_TRANSLATED	PATH_INFO расширением имен типа «~account»).
QUERY_STRING	Все данные, следующие за символом «?» в URL. Также данные формы, передаваемые методом GET.
REQUEST_METHOD	Метод, используемый клиентом для запроса. Для CGI-программ это обычно POST или GET.

Переменные окружения

REMOTE_ADDR	IP-адрес клиента, делающего запрос.
REMOTE_HOST	Имя узла машины клиента, если оно доступно.
REMOTE_IDENT	Если сервер и клиент поддерживают идентификацию типа identd, то это имя учетной записи пользователя, который делает запрос.
SCRIPT_NAME	Путь к выполняемому сценарию, указанный клиентом. Может использоваться при ссылке URL на самого себя.
SERVER_NAME	Имя узла - или IP-адрес, если имя недоступно, машины, на которой выполняется веб-сервер.
SERVER_SOFTWARE	Данные о версии веб-сервера, выполняющего CGI-программу.

Командная строка.

- CGI допускает передачу CGI-программе аргументов в качестве параметров командной строки, которая редко используется.
- Если переменная окружения QUERY_STRING не содержит символа « = », то CGI-программа будет выполняться с параметрами командной строки, взятыми из OUERY_STRING. Например, http://www.myseruer.com/cgi-bin/finger?root запустит finger root на www.myserver.com.

Стандартное устройство ввода.

- Если клиент передает данные методом PUT или POST, длина и тип MIME данных помещаются в CONTENT_LENGTH и CONTENT_TYPE. Передаваемые данные посылаются на стандартное устройство ввода CGI-программы.
- Признак конца данных может не посылаться программе - она должна взять значение переменной CONTENT_LENGTH и прочесть столько байтов, сколько в ней указано.
- Это основной метод передачи данных из форм.

Стандартное устройство вывода.

- Данные, посылаемые CGI-программой на стандартное устройство вывода, читаются веб-сервером и отправляются клиенту.
- Если имя сценария начинается с nph-, то данные посылаются прямо клиенту без вмешательства со стороны веб-сервера. В этом случае CGI-программа должна сформировать правильный заголовок HTTP, который будет понятен клиенту. В противном случае предоставьте веб-серверу сформировать HTTP-заголовок.

Стандартное устройство вывода.

- Даже если вы не используете nph-сценарий, серверу нужно дать одну директиву, которая сообщит ему сведения о вашей выдаче. Обычно это HTTP-заголовок Content-Type , но может быть и заголовок Location . За заголовком должна следовать пустая строка, то есть перевод строки или комбинация CR/LF.

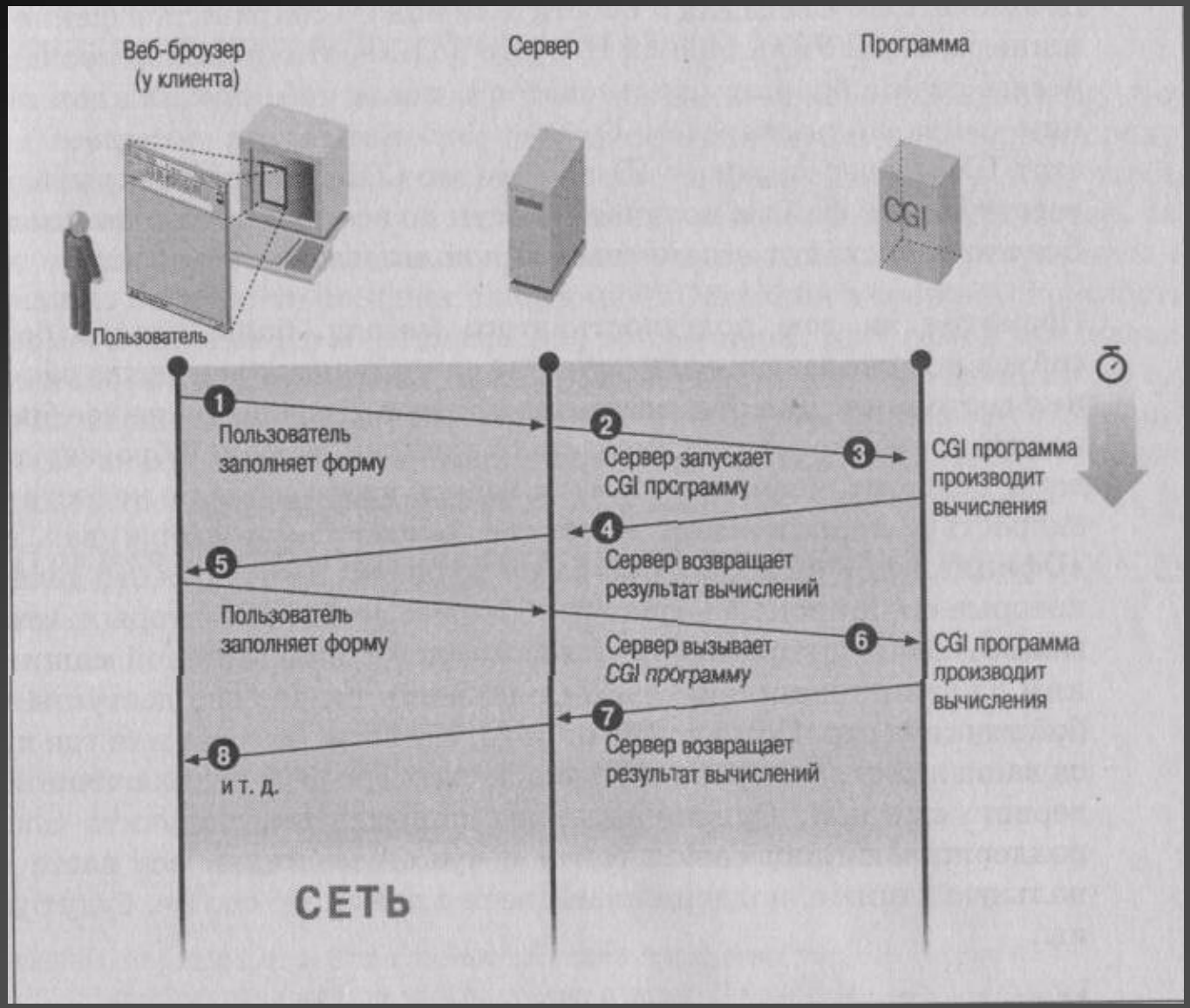
Стандартное устройство вывода.

- Заголовок Content-Type сообщает серверу, какого типа данные выдает ваша CGI-программа. Если это страница HTML, то строка должна быть Content-Type: text/html.
- Заголовок Location сообщает серверу другой URL или другой путь на том же сервере, куда нужно направить клиента. Заголовок должен иметь вид:
- `location:http://www.myserver.com/another/place`

Стандартное устройство вывода.

- После заголовков HTTP и пустой строки можно посылать собственно данные, выдаваемые вашей программой - страницу HTML, изображение, текст или что-либо еще.
- Если данные (HTML-код) отправляются ДО формирования заголовка, то заголовок будет сформирован автоматически, с параметрами (тип данных, язык, кодировка) по умолчанию.

Запоминание состояния



Запоминание состояния

- На стороне клиента – cookie (теневые посылки)
- На стороне сервера – сеансовые переменные. Для передачи их ID применяют:
  - Cookie
  - URL
  - теги <INPUT> типа HIDDEN в формах.

Меры безопасности

- Сам протокол CGI достаточно защищен. CGI-программа получает данные от сервера через стандартное устройство ввода или переменные окружения, и оба эти метода являются безопасными.
- Плохо написанная CGI-программа может позволить злоумышленнику получить доступ к системе сервера. Одно из решений состоит в синтаксическом анализе поступивших от формы данных с целью поиска злонамеренного содержания.

Меры безопасности

- Права пользователя. По умолчанию веб-сервер запускает программу CGI с правами того пользователя, который запустил сам сервер. Обычно это псевдопользователь, такой как «nobody», имеющий ограниченные права, поэтому у CGI-программы тоже мало прав.
- Если программе CGI нужно читать или записывать файлы, она может делать это только там, где у нее есть такое разрешение. CGI-программа должна иметь разрешение на чтение и запись в нужном ей каталоге, не говоря уже о самих файлах.

CGI и базы данных

- CGI-программы используются в качестве интерфейса:
  - к серверам баз данных, таким как MySQL, MS SQL
  - к настольным базам данных, таким как Microsoft Access.
- работают с плоскими текстовыми файлами, являющимися самыми простыми базами данных.