

Игорь Борисов

Веб-мастеринг. Настройка веб-серверов для веб-разработчиков

<http://igor-borisov.ru>

Темы курса

- Основы сетей: как всё происходит
- Основы HTTP
- Работа с веб-сервером Apache
- Работа с веб-сервером Nginx

Модуль 1

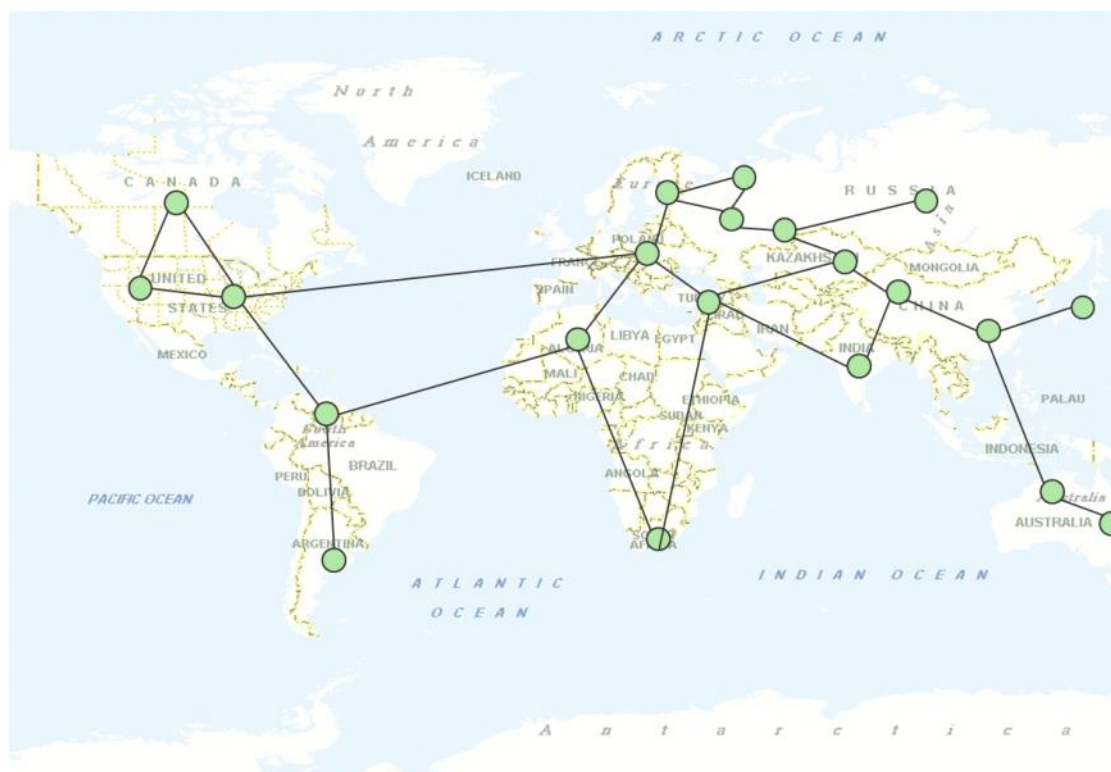
Веб-мастеринг.

Основы сетей: как всё происходит

Темы модуля

- Сети TCP/IP
- Модель OSI
- Адресация в сетях
- Маршрутизация
- Порты
- Разрешение имен
- DNS
- Использование полезных утилит

Сети в XXI веке



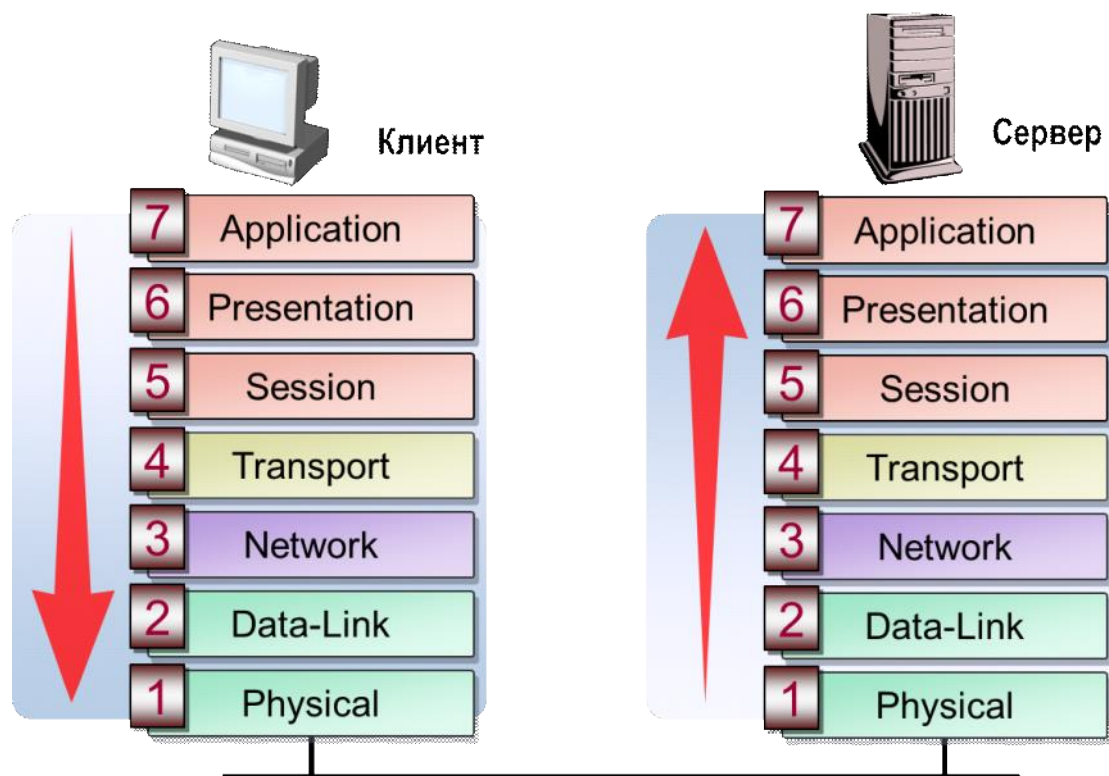
Модель OSI



Open Systems Interconnection Reference Model-OSI

эталонная модель взаимодействия открытых систем

Сетевые коммуникации



Адресация в сетях

- IP(v4) адрес: 213.180.204.8

- | | | | |
|----------|----------|----------|----------|
| 11010101 | 10110100 | 11001100 | 00001000 |
|----------|----------|----------|----------|

- Маска сети: 255.255.255.0

- | | | | |
|----------|----------|----------|----------|
| 11111111 | 11111111 | 11111111 | 00000000 |
|----------|----------|----------|----------|

- Классы сетей

- Класс А

- 255.0.0.0

- 1.x.x.x - 126.x.x.x

- Класс В

- 255.255.0.0

- 128.0.x.x - 191.255.x.x

- Класс С

- 255.255.255.0

- 192.0.0.x - 223.255.255.x

- Специальные и частные адреса:

- Широковещательное сообщение (direct broadcast)

- x.x.x.255

- Ограниченное широковещательное сообщение (limited broadcast)

- 255.255.255.255

- Для построения локальных сетей

- 10.x.x.x

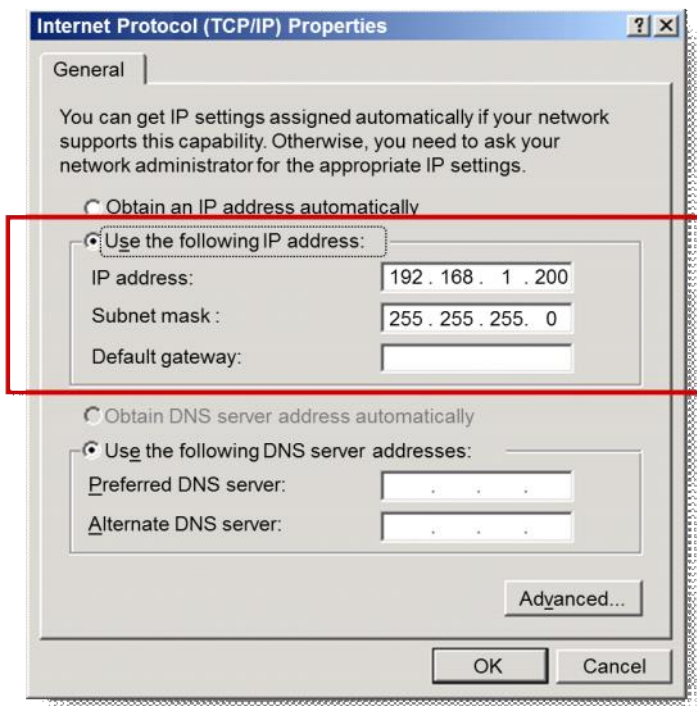
- 172.16.x.x – 172.31.x.x

- 192.168.x.x

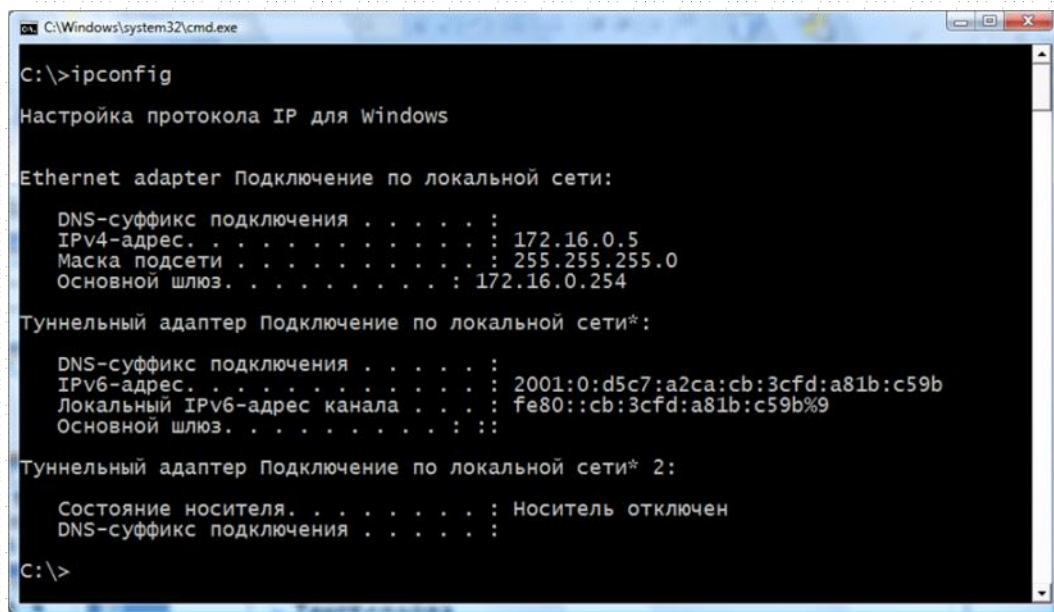
- Для внутреннего использования

- 127.0.0.x

Назначение IP-адресов



Утилита ipconfig



```
C:\Windows\system32\cmd.exe
C:\>ipconfig

Настройка протокола IP для Windows

Ethernet adapter Подключение по локальной сети:

    DNS-суффикс подключения . . . . . : 
    IPv4-адрес. . . . . : 172.16.0.5
    Маска подсети . . . . . : 255.255.255.0
    Основной шлюз. . . . . : 172.16.0.254

Туннельный адаптер Подключение по локальной сети*:

    DNS-суффикс подключения . . . . . : 
    IPv6-адрес. . . . . : 2001:0:d5c7:a2ca:cb:3cfd:a81b:c59b
    Локальный IPv6-адрес канала . . . : fe80::cb:3cfd:a81b:c59b%9
    Основной шлюз. . . . . : ::

Туннельный адаптер Подключение по локальной сети* 2:

    Состояние носителя. . . . . : Носитель отключен
    DNS-суффикс подключения . . . . . :
```

- /all
 - вывод подробных сведений
- /renew
 - обновление адресов всех адаптеров

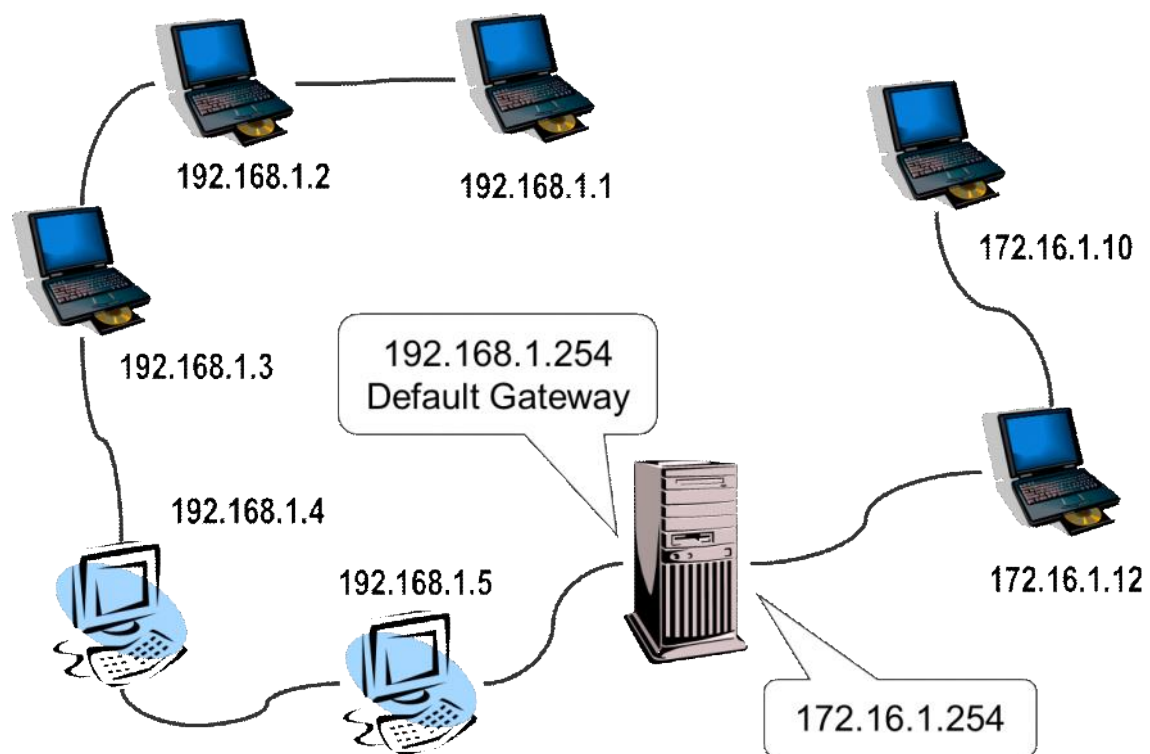
Общение в локальной сети

- Сетевой концентратор (hub)
- Сетевой коммутатор (switch)

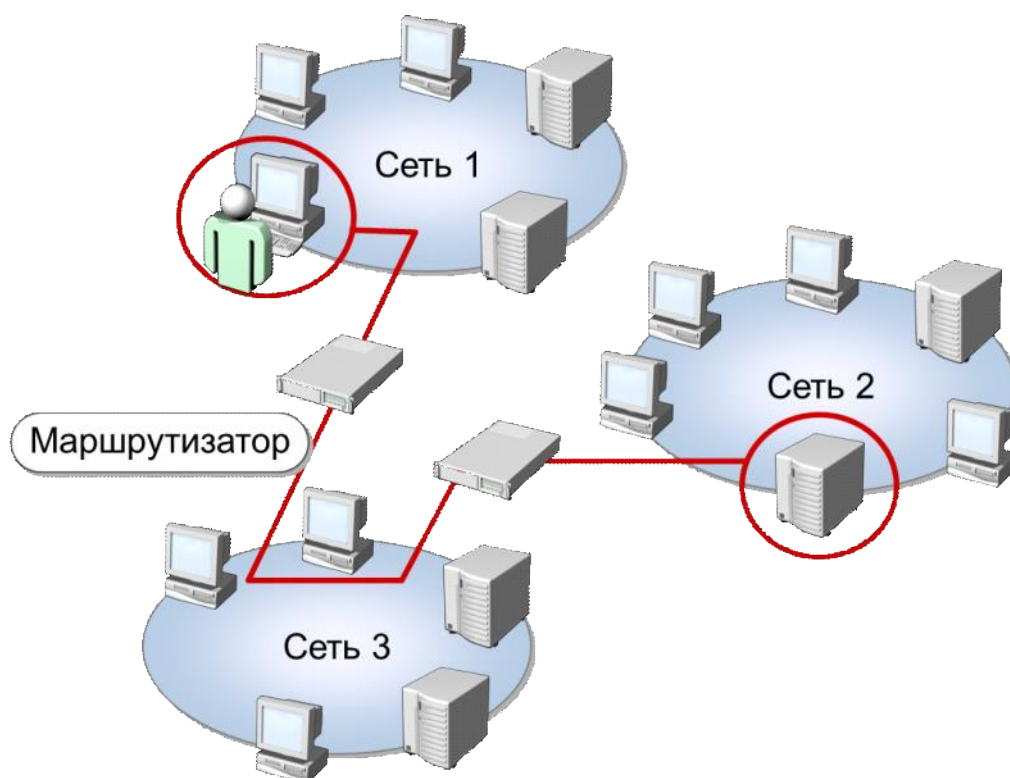


- MAC-адрес (Media Access Control)
 - 00-26-9E-59-3A-BD

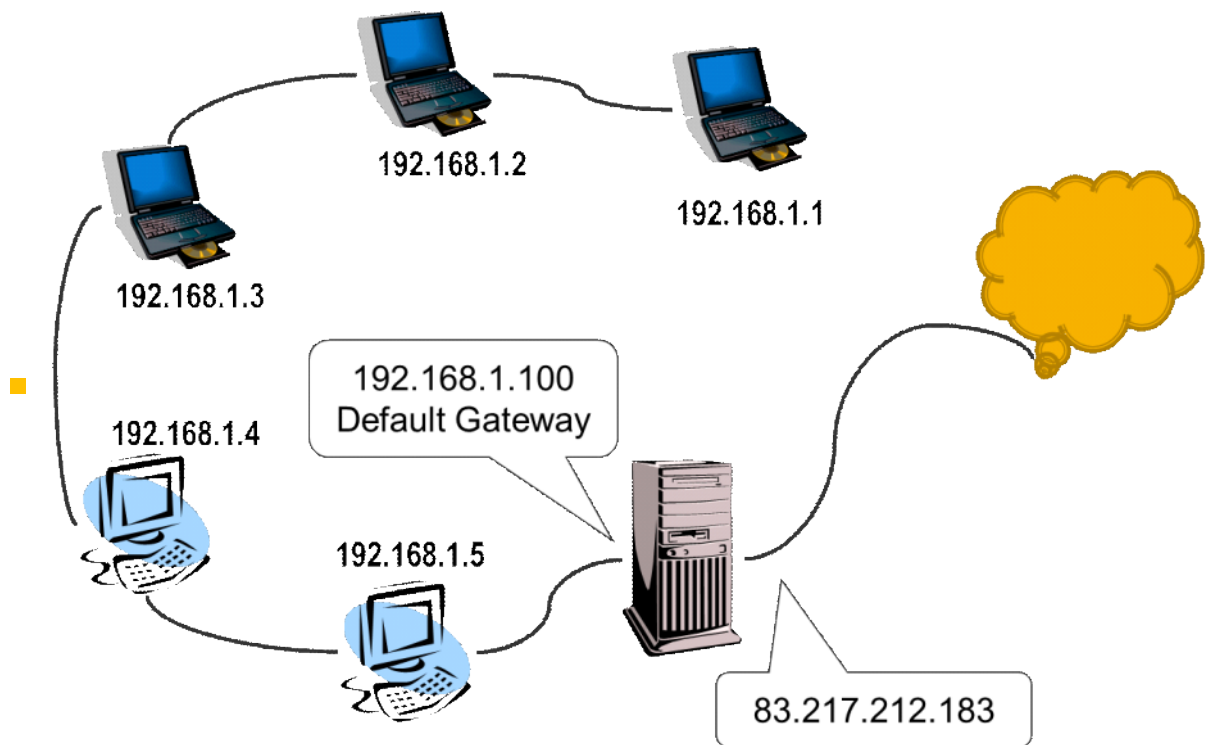
Маршрутизация



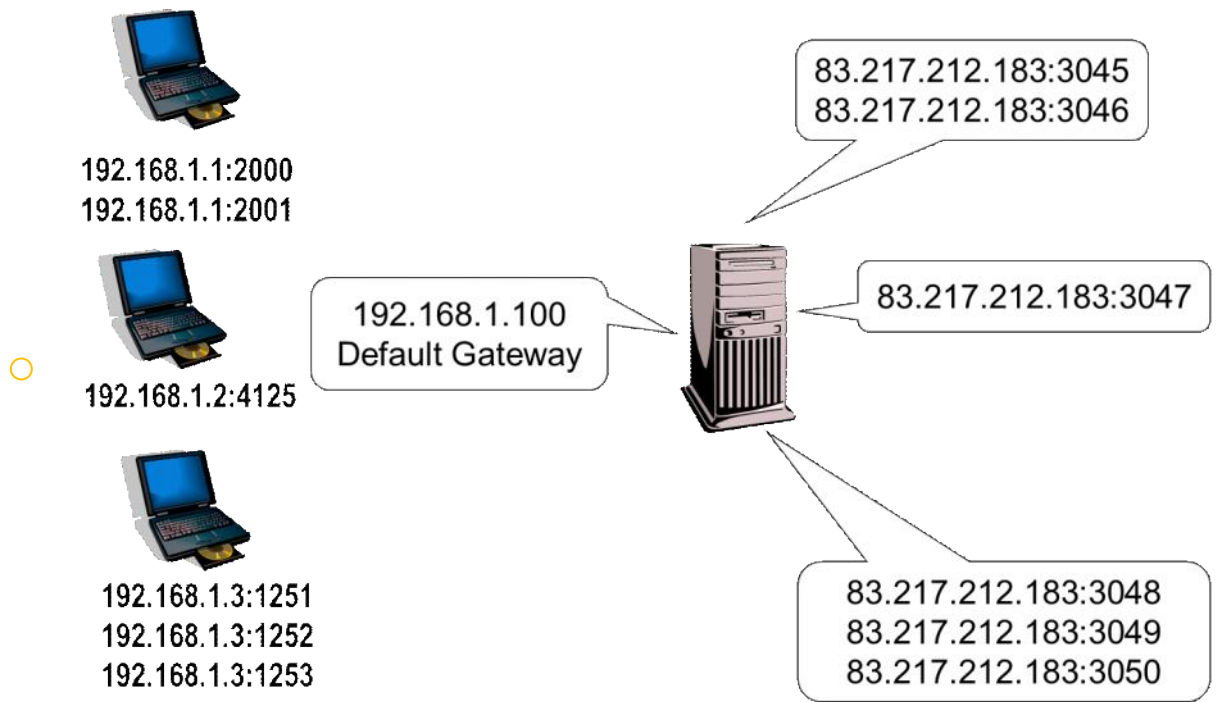
Маршрутизаторы в сети



NAT – Network Address Translation

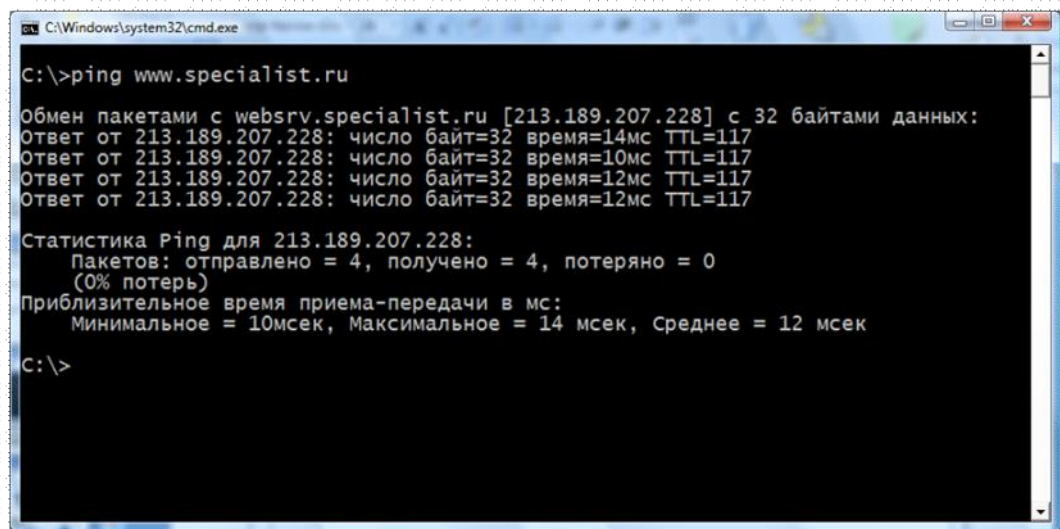


- Что такое "порт"?
 - 192.168.0.1:**2125**
- Сокет
 - 192.168.0.1:**2125** <==> 213.180.204.3:**80**
- NAT: Overload



Трассировка

■ Утилита **ping**



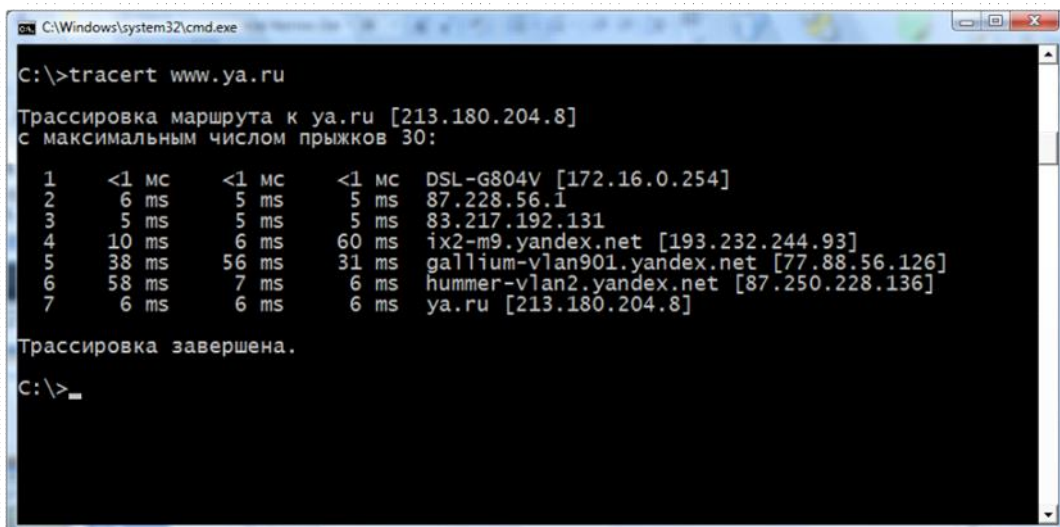
```
C:\Windows\system32\cmd.exe
C:\>ping www.specialist.ru

Обмен пакетами с webserv.specialist.ru [213.189.207.228] с 32 байтами данных:
Ответ от 213.189.207.228: число байт=32 время=14мс TTL=117
Ответ от 213.189.207.228: число байт=32 время=10мс TTL=117
Ответ от 213.189.207.228: число байт=32 время=12мс TTL=117
Ответ от 213.189.207.228: число байт=32 время=12мс TTL=117

Статистика Ping для 213.189.207.228:
  Пакетов: отправлено = 4, получено = 4, потеряно = 0
  (0% потеря)
  Приблизительное время приема-передачи в мс:
    Минимальное = 10мсек, Максимальное = 14 мсек, Среднее = 12 мсек

C:\>
```

■ Утилита **tracert**



```
C:\Windows\system32\cmd.exe
C:\>tracert www.ya.ru

Трассировка маршрута к ya.ru [213.180.204.8]
с максимальным числом прыжков 30:

 1  <1 мс    <1 мс    <1 мс    DSL-G804V [172.16.0.254]
 2   6 мс     5 мс     5 мс     87.228.56.1
 3   5 мс     5 мс     5 мс     83.217.192.131
 4  10 мс     6 мс    60 мс    ix2-m9.yandex.net [193.232.244.93]
 5  38 мс    56 мс    31 мс    gallium-vlan901.yandex.net [77.88.56.126]
 6  58 мс     7 мс     6 мс    hummer-vlan2.yandex.net [87.250.228.136]
 7   6 мс     6 мс     6 мс     ya.ru [213.180.204.8]

Трассировка завершена.

C:\>_
```

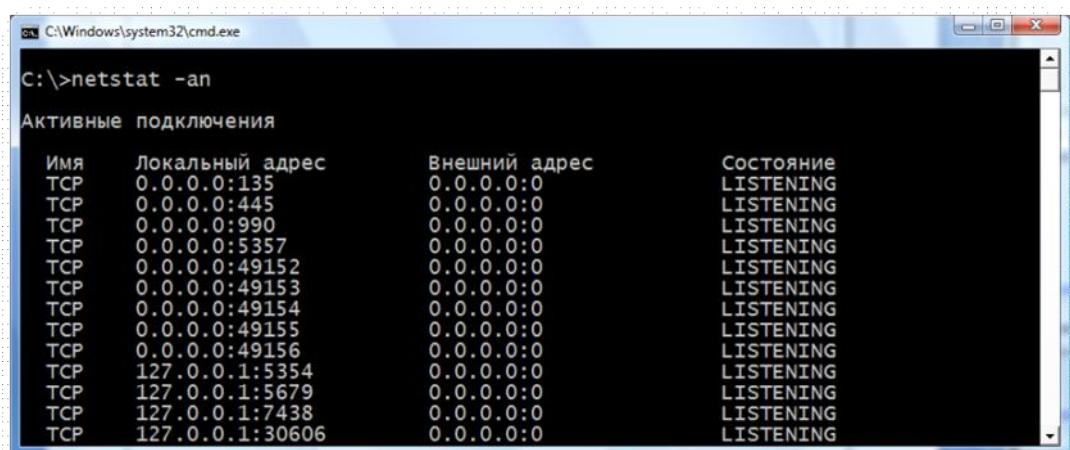

Сетевая статистика

■ Порты некоторых служб по умолчанию

- FTP: 21
- Telnet: 23
- SMTP: 25
- DNS: 53
- HTTP: 80
- POP3: 110
- HTTPS: 443
- MySQL: 3306

■ Утилита **netstat**

○



```
C:\Windows\system32\cmd.exe
C:\>netstat -an

Активные подключения

Имя      Локальный адрес      Внешний адрес      Состояние
TCP      0.0.0.0:135           0.0.0.0:0          LISTENING
TCP      0.0.0.0:445           0.0.0.0:0          LISTENING
TCP      0.0.0.0:990           0.0.0.0:0          LISTENING
TCP      0.0.0.0:5357          0.0.0.0:0          LISTENING
TCP      0.0.0.0:49152         0.0.0.0:0          LISTENING
TCP      0.0.0.0:49153         0.0.0.0:0          LISTENING
TCP      0.0.0.0:49154         0.0.0.0:0          LISTENING
TCP      0.0.0.0:49155         0.0.0.0:0          LISTENING
TCP      0.0.0.0:49156         0.0.0.0:0          LISTENING
TCP      127.0.0.1:5354        0.0.0.0:0          LISTENING
TCP      127.0.0.1:5679        0.0.0.0:0          LISTENING
TCP      127.0.0.1:7438        0.0.0.0:0          LISTENING
TCP      127.0.0.1:30606       0.0.0.0:0          LISTENING
```

■ Утилита **TCPView**

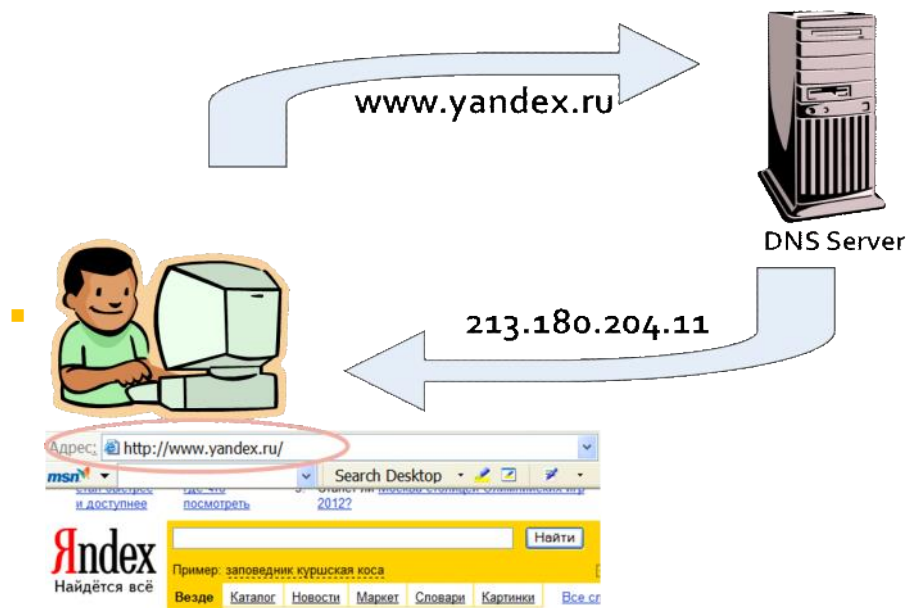
TCPView - Sysinternals: www.sysinternals.com

File Options Process View Help

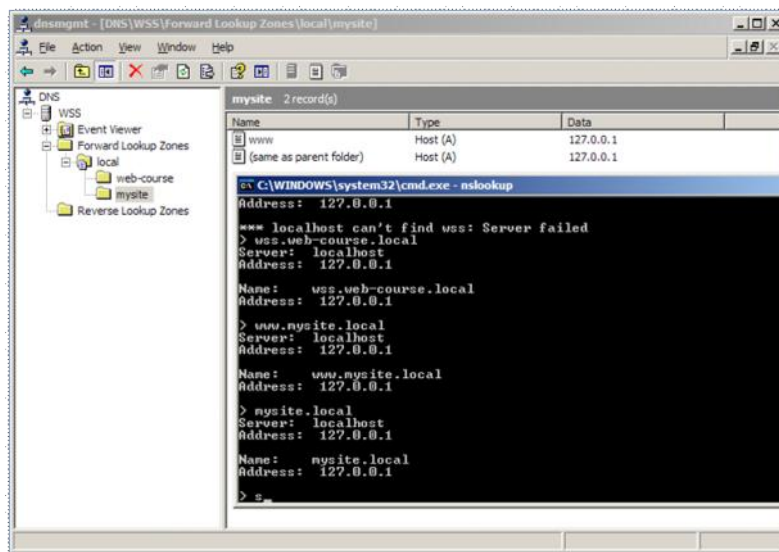
A ↩ ↗

Process	Protocol	Local Address	Remote Address	State
[System Process]:0	TCP	83.217.212.183:4...	212.48.41.37:995	TIME_WAIT
alg.exe:2904	TCP	in:1028	in:0	LISTENING
CCAPP.EXE:332	TCP	in:4143	in:0	LISTENING
explorer.exe:4028	UDP	in:4247	in:0	
ieexplore.exe:3280	UDP	in:4176	in:0	
inetinfo.exe:1176	TCP	in:http	in:0	LISTENING
inetinfo.exe:1176	TCP	in:https	in:0	LISTENING
inetinfo.exe:1176	TCP	in:1025	in:0	LISTENING
inetinfo.exe:1176	UDP	in:3456	in:0	
lsass.exe:1592	UDP	in:isakmp	in:0	
lsass.exe:1592	UDP	in:4500	in:0	
mqsvc.exe:1280	TCP	in:1027	in:0	LISTENING
mqsvc.exe:1280	TCP	in:1801	in:0	LISTENING
mqsvc.exe:1280	TCP	in:2103	in:0	LISTENING
mqsvc.exe:1280	TCP	in:2105	in:0	LISTENING
mqsvc.exe:1280	TCP	in:2107	in:0	LISTENING
mqsvc.exe:1280	UDP	in:1026	in:0	
mqsvc.exe:1280	UDP	in:3527	in:0	
msnmsgr.exe:4524	TCP	83.217.212.183:4...	207.46.2.46:1863	ESTABLISHED
msnmsgr.exe:4524	UDP	83.217.212.183:di...	in:0	
msnmsgr.exe:4524	UDP	in:4142	in:0	
MSTORDB.EXE:5132	UDP	in:4221	in:0	
OUTLOOK.EXE:4540	UDP	in:4150	in:0	

DNS - Domain Name System



■ DNS сервер

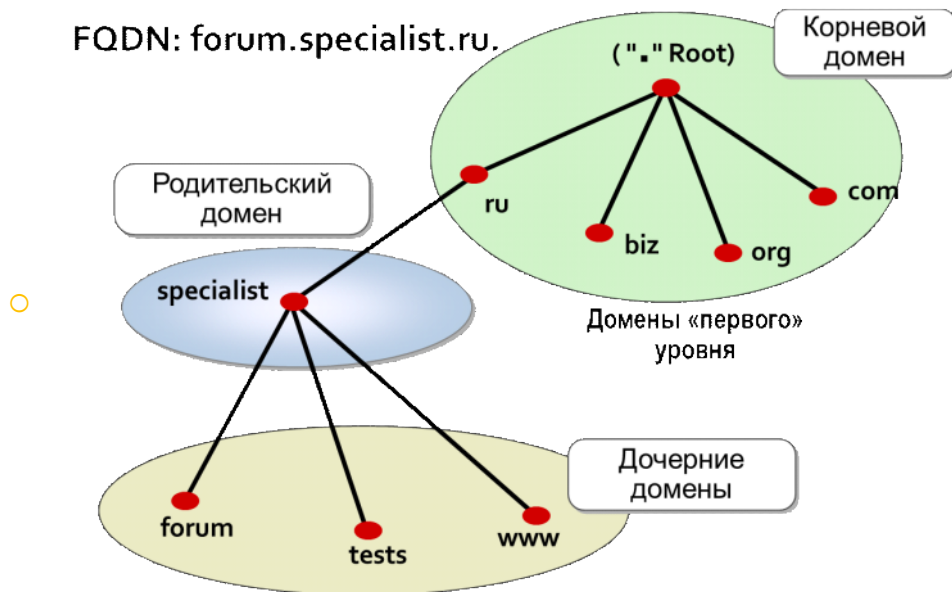


■ Типы записей DNS

- A – (address record)
 - запись адреса связывает имя хоста с адресом IP
- AAAA – (IPv6 address record)
 - связывает имя хоста с адресом протокола IPv6
- CNAME – (canonical name record) или псевдоним
 - используется для перенаправления на другое имя
- MX – (mail exchange)
 - почтовый обменник указывает на почтовый сервер

- PTR – (pointer) или запись указателя
 - связывает IP хоста с его каноническим именем
- NS – (name server)
 - указывает на DNS-сервер для данного домена
- SOA – (Start of Authority) начальная запись зоны
 - на каком сервере хранится эталонная информация о данном домене

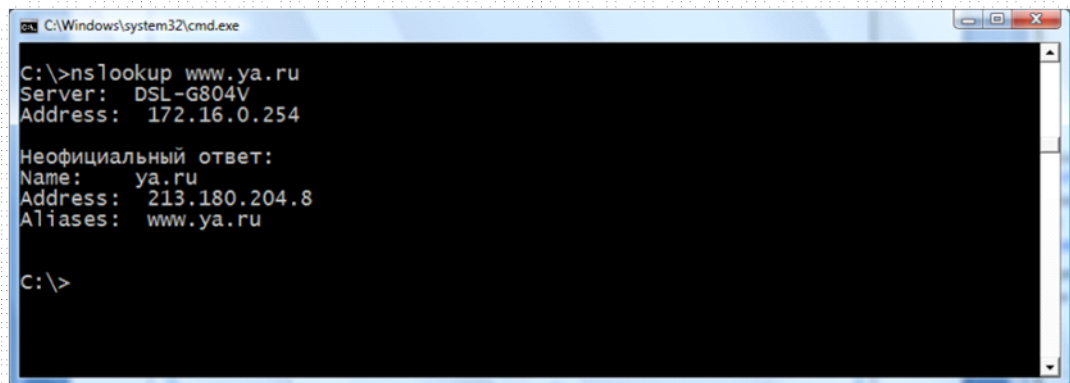
■ Принципы DNS



Способы определения адреса

- Файл **hosts**
 - C:\windows\system32\drivers\etc
- Обращение к локальному кэшу DNS
 - ipconfig /displaydns
- Обращение к назначенному DNS серверу
- Утилита **nslookup**

○



```
C:\Windows\system32\cmd.exe
C:\>nslookup www.ya.ru
Server: DSL-G804V
Address: 172.16.0.254

Неофициальный ответ:
Name: ya.ru
Address: 213.180.204.8
Aliases: www.ya.ru

C:\>
```

Что мы изучили?

- TCP/IP – стек протоколов организации сети
- Адресация в сетях, IP адрес, маска
- Маршрутизация – передача пакетов в различные сети
- Порты – каналы ввода/вывода информации
- Разрешение имен – преобразование имени в адрес
- DNS – служба доменных имен

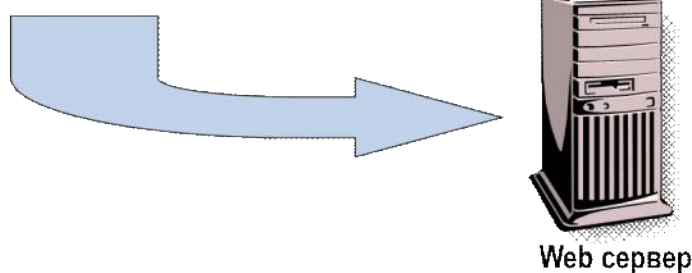
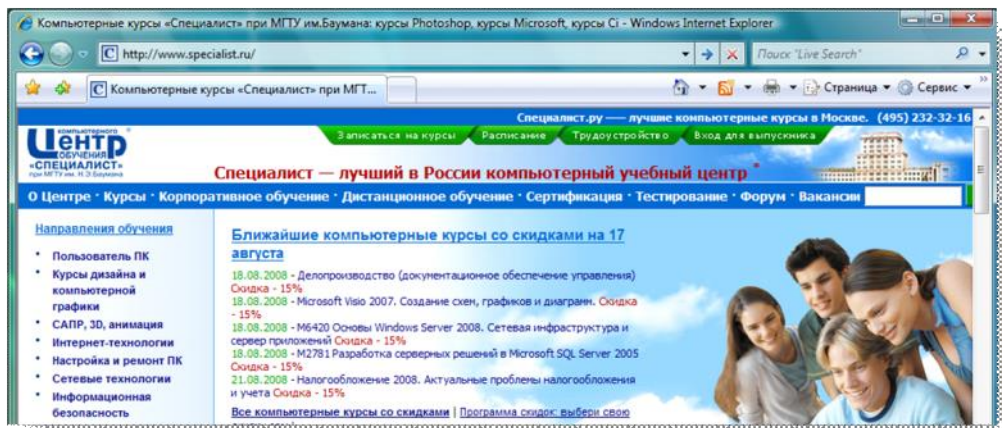
Модуль 2

Веб-мастеринг. Основы HTTP

Темы модуля

- Терминология (RFC 7231)
- Методы запросов
- Статус сервера
- Заголовки запроса и ответа
- Передача данных
- Расширяемость протокола
- Обзор существующих Веб-серверов

HyperText Transfer Protocol



■ Запрос клиента

```
GET /folder/index.html HTTP/1.1
Host: mysite.ru
User-Agent: Mozilla/5.0 ...
Accept: */*
Accept-Language: ru,en-us
Accept-Encoding: gzip,deflate
Accept-Charset: windows-1251,utf-8
Keep-Alive: 300
X-Some-Header: 1234
Connection: keep-alive
```

■ Ответ сервера

```
HTTP/1.1 200 OK
```

Date: Sun, 14 Aug 2016 07:47:24 GMT
Server: Microsoft-IIS/8.0
Last-Modified: Mon, 04 Jul 2016 11:07:04 GMT
Accept-Ranges: bytes
Content-Length: 34234
Content-Type: text/html
<html>
...
</html>

Статусы ответа сервера

- 1xx – Информационное сообщение
 - 100 Continue
- 2xx – Успешное выполнение команды
 - 200 OK
 - 201 Created
 - 202 Accepted
 - 204 No Content
- 3xx – Переадресация (как правило, требуется еще один запрос)
 - 301 Moved Permanently
 - + Location
 - 302 Found
 - + Location
 - 304 Not Modified
 - -> If-Modified-Since и/или If-None-Match
- 4xx – Ошибка на стороне клиента
 - 400 Bad Request
 - 401 Unauthorized
 - + WWW-Authenticate
 - -> Authorization
 - 403 Forbidden
 - 404 Not Found
 - 405 Method Not Allowed
 - + Allow
 - 410 Gone
 - 411 Length Required
 - 429 Too Many Requests
 - 451 Unavailable For Legal Reasons
- 5xx – Ошибка на стороне сервера
 - 500 Internal Server Error
 - 501 Not Implemented
 - 502 Bad Gateway

- 503 Service Unavailable
 - + Retry-After
- 504 Gateway Timeout

Лабораторная работа 2.1

Просмотр обмена данными между браузером и веб-сервером

Содержание лабораторной работы 2.1

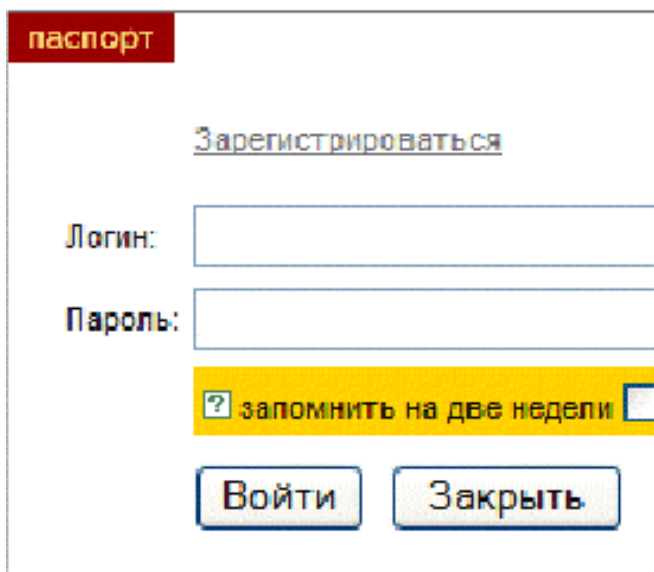
Просмотр обмена данными между браузером и веб-сервером

Упражнение: Обмен данными между клиентом и сервером

- Откройте текстовый редактор.
- Наберите следующий текст:

```
<html>
  <head>
    <title>Test page</title>
  </head>
  <body>
    <h1>Test page</h1>
  </body>
</html>
```
- Сохраните файл под именем **test.html** в папке **\www**
- В браузере Google Chrome откройте "Инструменты разработчика" нажав на клавишу F12 и выберите пункт "Network".
- В адресной строке браузера наберите:
<http://127.0.0.1/test.html>
Вы должны увидеть свою страницу
- В "Инструментах разработчика" в списке (находится справа) нажмите на строку "test.html" и выберите слева вкладку "Headers". Посмотрите, каким методом был произведён запрос страницы и какой статус ответа прислал сервер.
- В браузере нажмите клавишу **F5** и повторите действия из предыдущего пункта. Обратите внимание, какой статус прислал сервер?
- Если сервер ответил статусом **304**, то среди заголовков ответа (Request Headers) найдите заголовок-валидатор, который привел к такому ответу.
- В текстовом редакторе в файле **test.html** измените содержимое заголовка первого уровня на **Test page!** и сохраните изменение
- В браузере нажмите клавишу **F5**.
- Посмотрите статус и заголовки ответа сервера. Обратите внимание, сервер ответил статусом **200**, хотя заголовок-валидатор присутствует. Почему?
- Остановите сервер. Для этого в окне командной строки сервера нажмите сочетание клавиш **Ctrl+C**

Передача данных из формы



- Веб-форма

```
<form action="action.php" method="get">  
  Логин:  
  <input name="login" type="text">  
  Пароль:  
  <input name="pwd" type="password">  
  <input type="submit">  
</form>
```

- Запрос методом GET

```
GET /action.php?login=Vasya&pwd=Parol HTTP/1.1  
Host: mysite.ru  
User-Agent: Mozilla/5.0 ...  
Accept: */*  
Accept-Language: ru,en-us
```

- Запрос методом POST

```
POST /action.php HTTP/1.1
Host: mysite.ru
User-Agent: Mozilla/5.0 ...
Accept: */*
Accept-Language: ru,en-us
Referer: http://mysite.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 26

login=vasya&password=parol
```


Лабораторная работа 2.2

Передача и просмотр данных, передаваемых методами GET и POST

Содержание лабораторной работы 2.2

Передача и просмотр данных, передаваемые методами GET и POST

Упражнение 1: Создание файла с кодом исполняемым на сервере

- В данном упражнении мы будем писать код на серверной технологии PHP
- В текстовом редакторе создайте новый файл
- Наберите следующий текст:

```
<?php
$params = [];
$method = $_SERVER["REQUEST_METHOD"];
if($method == "POST")
    $params = $_POST;
else
    $params = $_GET;
print "<h1>Method: $method</h1>";
foreach($params as $key=>$value)
    print "$key : $value<br>";
?>
```
- Сохраните файл под именем **test.php** в папку **\www**
- Запустите веб-сервер. Для этого:
 - в проводнике откройте папку **\server**
 - кликните два раза по файлу **http-server.bat**. Откроется окно командной строки - не закрывайте его
- В браузере наберите в адресной строке:
<http://127.0.0.1/test.php>
Вы должны увидеть страницу с фразой **Method: GET**
- Если вы увидели сообщение об ошибке, проверьте текст php-скрипта и только в самом крайнем случае обратитесь к преподавателю за помощью. Переходите к следующему упражнению лабораторной работы только в случае успешного завершения этого упражнения

Упражнение 2: Передача данных методом GET

- В тестовом редакторе создайте новый файл
- Наберите следующий текст:

```
<html>
<head>
    <title>Method GET</title>
</head>
<body>
    <h1>Method GET</h1>
    <form action="test.php" method="get">
        <div>
            <label for="n">Ваше имя:</label>
            <input id="n" type="text" name="name">
        </div>
        <div>
            <label for="a">Ваш возраст:</label>
            <input id="a" type="text" name="age">
        </div>
    </form>
</body>
</html>
```

```
        <input type="submit" value="Передать">
    </form>
</body>
</html>
```

- Сохраните файл под именем **get.html** в папке **\www**
- В браузере наберите в адресной строке:
<http://127.0.0.1/get.html>
- Введите произвольные данные в поля формы и нажмите кнопку **Передать**
- Переключитесь на "Инструменты разработчика" (см. Лабораторная работа 2.1) и посмотрите, каким методом был осуществлён запрос?
- Найдите ваши данные в запросе

Упражнение 3: Передача данных методом POST

- Пересохраните файл **get.html** как **post.html**
- В содержимом файла внесите следующие изменения:
 - В содержимом html-элемента **title** поменяйте слово **GET** на **POST**
 - В содержимом html-элемента **h1** поменяйте слово **GET** на **POST**
 - Установите значение атрибута **method** html-элемента **form** в **post**
- В браузере наберите в адресной строке:
<http://127.0.0.1/post.html>
- Введите произвольные данные в поля формы и нажмите кнопку **Передать**
- В "Инструментах разработчика" посмотрите, каким методом был осуществлён запрос?
- Найдите ваши данные в запросе
- Остановите сервер. Для этого в окне командной строки сервера нажмите сочетание клавиш **Ctrl+C**

Куки (cookie)

- Веб-форма

```
<form action="action.php" method="post">  
  Как вас зовут?  
  <input name="name" type="text">  
  <input type="submit">  
</form>
```

- Передача данных на сервер

```
POST /action.php HTTP/1.1  
Host: mysite.ru  
...  
name=Вася
```

- Установка cookie

```
HTTP/1.1 200 OK  
Server: Microsoft-IIS/8.0  
...  
Set-Cookie: user=Вася  
<html>  
...  
</html>
```

- Передача cookie на сервер

GET /page.php HTTP/1.1 ↵

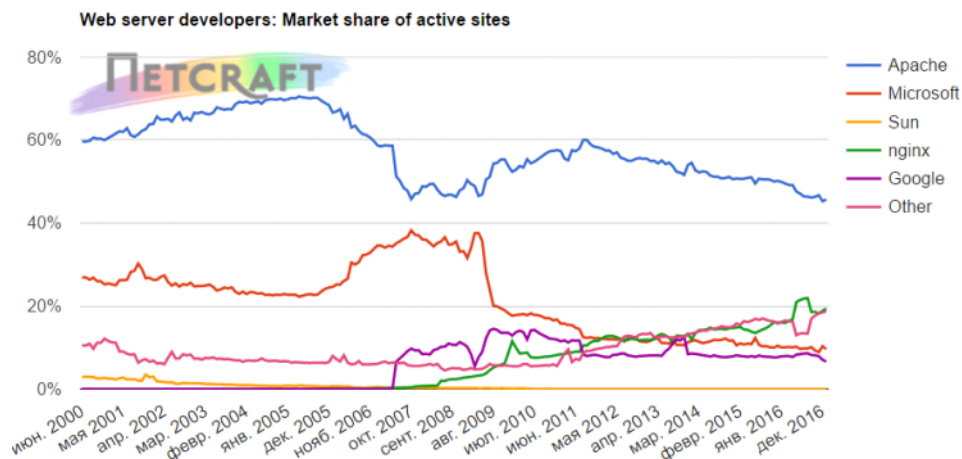
Host: mysite.ru ↵

... ↵

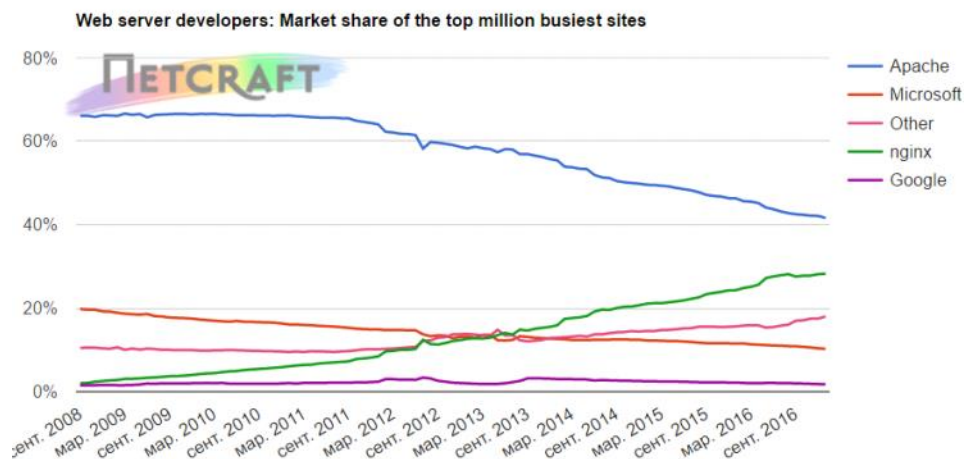
Cookie: user=Вася ↵

↵

Популярность веб-серверов



Developer	December 2016	Percent	January 2017	Percent	Change
Apache	77,011,462	45.27%	78,707,037	45.67%	0.40
nginx	32,113,723	18.88%	33,331,358	19.34%	0.46
Microsoft	17,554,286	10.32%	16,601,302	9.63%	-0.69
Google	12,002,411	7.05%	11,372,796	6.60%	-0.46



Developer	December 2016	Percent	January 2017	Percent	Change
Apache	421,308	42.13%	416,257	41.63%	-0.51
nginx	281,525	28.15%	282,986	28.30%	0.15
Microsoft	103,862	10.39%	102,660	10.27%	-0.12
Google	18,230	1.82%	17,702	1.77%	-0.05

Что мы изучили?

- Методы запроса - команды серверу на выполнение операций
- Статус ответа сервера - код выполненной операции
- Заголовки запроса и ответа - обмен мета-информацией между клиентом и сервером
- Передача данных на сервер с помощью веб-формы осуществляется методами GET и POST
- HTTP имеет возможность расширения
- Apache и Nginx - самые популярные веб-серверы

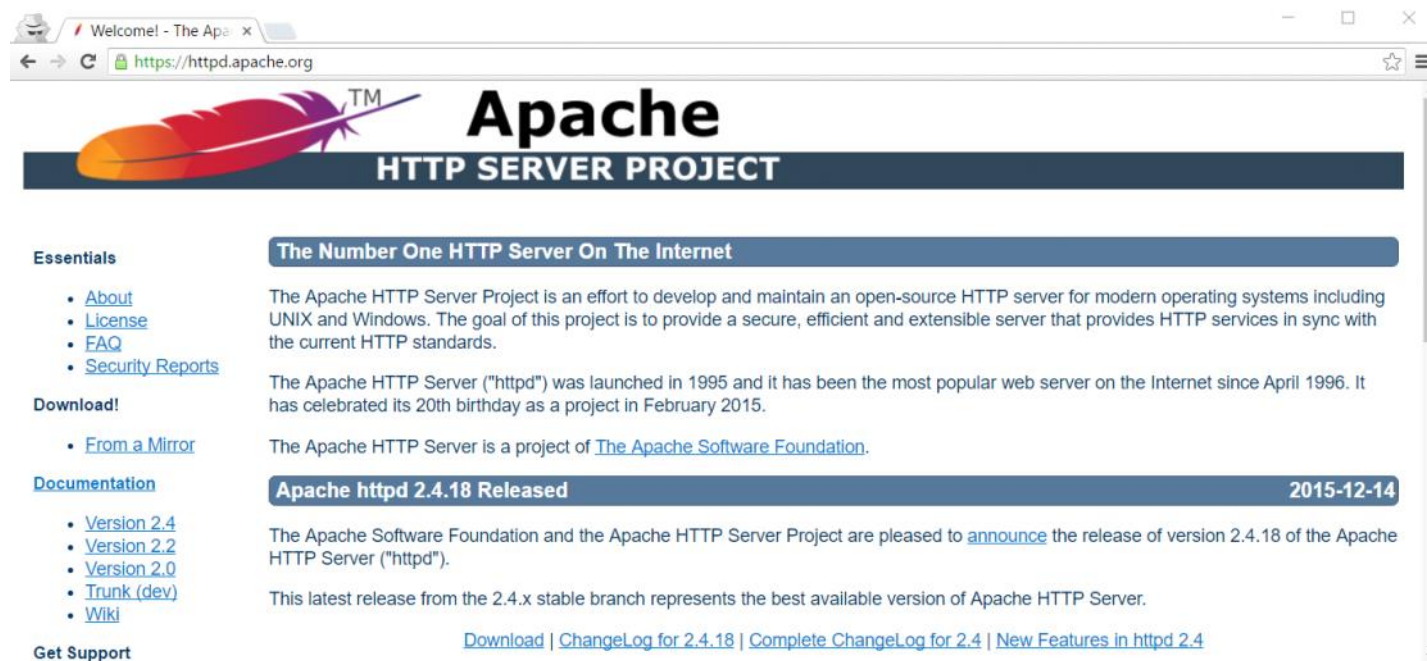
Модуль 3

Веб-мастеринг. Веб-сервер Apache

Темы модуля

- Обзор и установка сервера Apache
- Основные директивы настройки сервера Apache
- Файлы конфигурации сервера
- Виртуальные хосты
- Виртуальные директории
- Пользовательские сообщения об ошибках
- Использование пользовательских заголовков ответа
- Привязка расширений файлов к MIME-типам
- Управление кэшированием
- Привязка и использование серверных обработчиков
- Аутентификация и авторизация пользователя
- Защита HTTP-трафика с использованием SSL
- Использование модуля mod_rewrite

Веб-сервер Apache



The screenshot shows the Apache HTTP Server Project website in a web browser. The browser's address bar displays "https://httpd.apache.org". The website features the Apache logo (a feather) and the text "Apache HTTP SERVER PROJECT". The main content area is divided into several sections:

- Essentials**
 - [About](#)
 - [License](#)
 - [FAQ](#)
 - [Security Reports](#)
- Download!**
 - [From a Mirror](#)
- Documentation**
 - [Version 2.4](#)
 - [Version 2.2](#)
 - [Version 2.0](#)
 - [Trunk \(dev\)](#)
 - [Wiki](#)
- Get Support**

The main content area also includes a section titled "The Number One HTTP Server On The Internet" with a sub-header "The Number One HTTP Server On The Internet". This section contains text about the project's goals and history, and a link to "The Apache Software Foundation".

Below this, there is a section titled "Apache httpd 2.4.18 Released" with a date "2015-12-14". This section contains text about the release and links to "Download", "ChangeLog for 2.4.18", "Complete ChangeLog for 2.4", and "New Features in httpd 2.4".

Основные директории

- bin
 - программы и утилиты
 - httpd.exe
- modules
 - расширения сервера
- conf
 - конфигурация сервера
 - httpd.conf (**настроить! ReadMe.txt**)
 - extra\

Управление сервером

- Запуск из командной строки
 - httpd.exe
 - Ctrl+C
- Сервер как службы ОС Windows
 - Установка/удаление
 - httpd -k install
 - httpd -k uninstall
 - Использование
 - net start apache2.4
 - net stop apache2.4

 - httpd -k start
 - httpd -k stop
 - httpd -k restart
 - bin\ApacheMonitor.exe

Лабораторная работа 3.1

Использование системной переменной PATH

Содержание лабораторной работы 3.1

Использование системной переменной PATH

Упражнение 1: Добавление пути к папке bin в переменную PATH

- Откройте окно **Система** через Панель управления ОС Windows:
- В левой части окна Система щелкните мышкой по ссылке (пункту) **Дополнительные параметры системы**
- В диалоговом окне **Свойства** системы выберите вкладку **Дополнительно**
- Нажмите на кнопку **[Переменные среды...]**
- В диалоговом окне **Переменные среды** прокрутите нижний список **Системные переменные** до появления строки со значением **PATH** в левом столбике **Переменная** и дважды кликните мышкой по этой строке
- Откройте проводник Windows и перейдите в папку **bin** веб-сервера **Apache**
- Скопируйте путь к этой папке из адресной строки проводника
- Переключитесь обратно на диалоговое окно **Изменение системной переменной**
- Установите курсор **в самый конец** текстовых данных в поле **Значение переменной**
- Введите символ **точки с запятой (;)**
- **После** точки с запятой вставьте скопированный путь из буфера обмена
- Закройте последовательно все диалоговые окна нажимая кнопки **[ОК]**

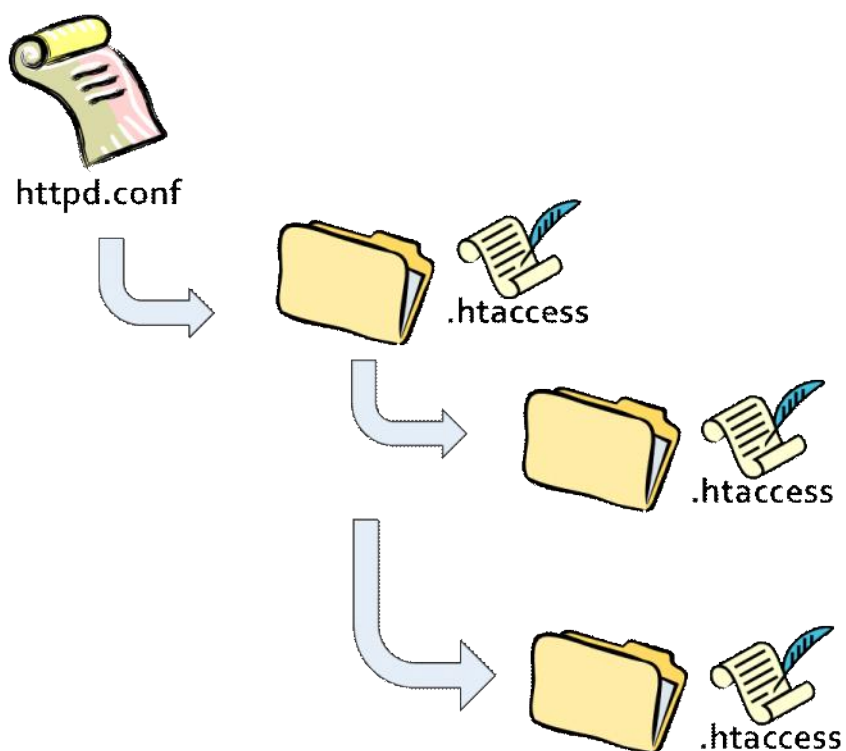
Правила конфигурации

- # Однострочный комментарий
- Однострочные директивы
 - ИмяДирективы Значение [Значение2 [...]]
- Блочные директивы
 - <ИмяДирективы Значение>
ИмяДирективы Значение
</ИмяДирективы>

Обзор основных директив

- ServerRoot
- Listen
- LoadModule
- ServerAdmin
- DocumentRoot
- Directory
 - AllowOverride (None | All)
 - Options (Indexes | -Indexes)
 - Require (all granted | all denied)
 - DirectoryIndex
- Files
- Include
- IndexIgnore
- ReadmeName
- HeaderName

Гибкая настройка директорий



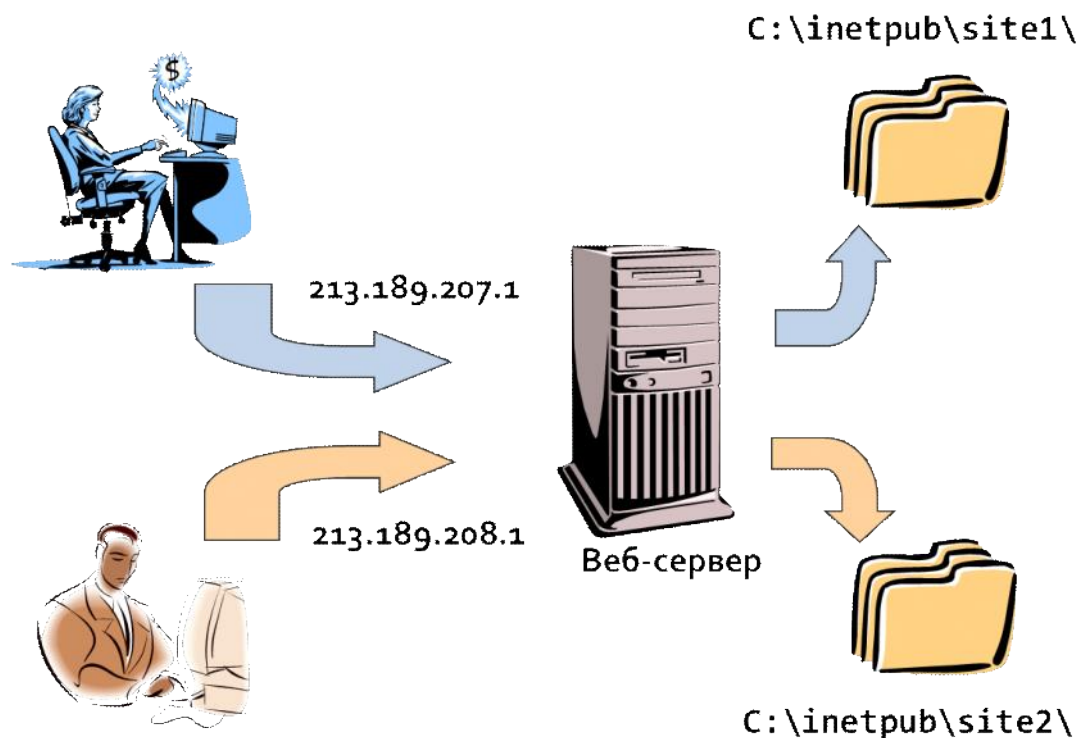
Перенаправление запросов

- Временное перенаправление
 - Redirect `"/old" "http://site.ru/new"`
 - Redirect 302 `"/old" "http://site.ru/new"`
 - Redirect temp `"/old" "http://site.ru/new"`
 - RedirectTemp `"/old" "http://site.ru/new"`
- Постоянное перенаправление
 - Redirect 301 `"/old" "http://site.ru/new"`
 - Redirect permanent `"/old" "http://site.ru/new"`
 - RedirectPermanent `"/old" "http://site.ru/new"`
- Использование статуса 410
 - Redirect gone `"/deleted.html"`

Настройка файлов журналов

- Журнал ошибок сервера
 - `ErrorLog "logs/error.log"`
- Журналы с данными запросов
 - `CustomLog "logs/access.log" common`
 - `CustomLog "logs/agents.log" agent`
 - `CustomLog "logs/refs.log" referer`
- Формат записи данных для журналов
 - `LogFormat "%h %t %r %>s %b" common`
 - `LogFormat "%{User-Agent}i" agent`
 - `LogFormat "%{Referer}i" referer`

Виртуальные хосты



- Директива
 - `<VirtualHost ip:port>`
 - # Описание виртуального хоста
 - `</VirtualHost>`
- По порту
 - `<VirtualHost *:8001>`
 - `<VirtualHost *:8002>`
- По IP
 - `<VirtualHost 192.168.1.1:80>`
 - `<VirtualHost 192.168.1.2:80>`
- По имени
 - `<VirtualHost *:80>`

- ServerName site.ru
- </VirtualHost>
- GET /folder HTTP/1.1
- Host: site.ru
- <VirtualHost *:80>
- ServerName super-site.ru
- </VirtualHost>
- GET /folder HTTP/1.1
- Host: super-site.ru

Настройка виртуального хоста

```
<VirtualHost *:80>
    # Имя сервера
    ServerName  site.ru
    ServerAlias  mysite.ru supersite.ru

    # Администратор
    ServerAdmin  admin@site.ru

    # Корень сервера
    DocumentRoot "c:\server\site\htdocs"

    # Настройка папки
    <Directory "c:\server\site\htdocs">
        # Директивы настройки папки
    </Directory>
</VirtualHost>
```

Лабораторная работа 3.2

Создание и настройка виртуального хоста

Содержание лабораторной работы 3.2

Создание и настройка виртуального хоста

Упражнение 1: Создание папки и конфигурационного файла виртуального хоста

- В своей рабочей папке создайте папку **mysite**
- В папке **mysite** создайте следующие папки: **conf**, **htdocs**, **logs** и **errors**
- В текстовом редакторе создайте новый файл и напишите в нём:
Виртуальный хост **mysite.local**
- Сохраните файл под именем **mysite.conf** в папке **mysite\conf**

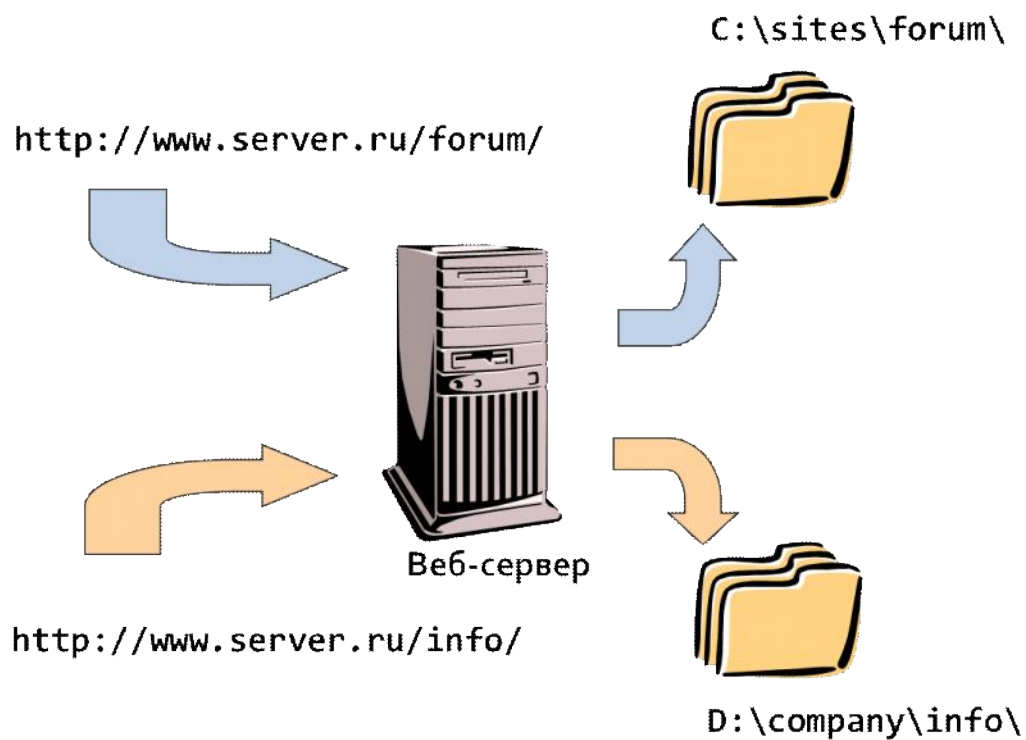
Упражнение 2: Создание виртуального хоста

- В файле **mysite\conf\mysite.conf** опишите виртуальный хост для **127.0.0.1:80**
- Задайте **ServerName** и **ServerAlias** для хостов **mysite.local**, www.mysite.local и **info.mysite.local**
- Определите файлы для **ErrorLog** и **CustomLog**. Файлы должны находиться в папке **mysite\logs**
- Определите **email** администратора сайта
- Определите корневую папку сайта **mysite\htdocs** и опишите её
- Откройте конфигурационный файл сервера Apache **httpd.conf** в текстовом редакторе и в самом конце этого файла подключите файл **mysite\conf\mysite.conf**
- Не забудьте сохранить все файлы

Упражнение 3: Проверка работы виртуального хоста

- Перезапустите сервер Apache
- Откройте браузер и наберите в адресной строке: <http://www.mysite.local>
 - Вы должны увидеть пустую корневую папку вашего сайта
- Самостоятельно создайте страницу **index.html** в корневой папке сайта
- Обращайте внимание на сообщения сервера. Если сервер не запускается, то где-то в конфигурации ошибка: найдите её и исправьте

Виртуальные директории



- Alias `"/info"` `"c:/users/folders/vasya/folder"`
- Alias `"/info/"` `"c:/users/folders/vasya/folder/"`
- Обязательно описать директорию!

Настройка сообщений об ошибках

- `ErrorDocument 500 "http://www.somesite.ru/error-handler"`
- `ErrorDocument 404 "/errors/script-404.php"`
- `ErrorDocument 401 "/errors/401.html"`
- `ErrorDocument 403 "У вас нет прав на просмотр этого файла"`

Лабораторная работа 3.3

Создание виртуальной директории и сообщений об ошибках

Содержание лабораторной работы 3.3

Создание виртуальной директории и сообщений об ошибках

Упражнение 1: Создание виртуальной директории

- В файле **mysite\conf\mysite.conf** в *виртуальном хосте* создайте виртуальную директорию **err/** для папки **mysite\errors**
- Опишите директорию **mysite\errors**
- Сохраните файл и перезапустите сервер
- В браузере наберите в адресной строке: <http://mysite.local/err/>
Вы должны увидеть пустую виртуальную директорию

Упражнение 2: Создание собственных сообщений об ошибках

- В тестовом редакторе создайте новый файл
- Наберите следующий текст:

```
<html>
  <head>
    <title>Ошибка 404</title>
  </head>
  <body>
    <h1>Ошибка 404</h1>
    <p>Файл не найден!</p>
    <p>Проверьте правильность написания адреса.</p>
  </body>
</html>
```
- Сохраните файл под именем **404.html** в папке **mysite\errors**
- В браузере наберите в адресной строке: <http://mysite.local/err/404.html>
- Убедитесь в том, что файл загружен в браузер
- В файле **mysite\conf\mysite.conf** в *виртуальном хосте* добавьте указание: для статуса **404** загружать страницу **404.html** из виртуальной папки **err**
- Сохраните файл и перезагрузите сервер
- В браузере наберите в адресной строке: <http://mysite.local/abc/>
- Если вы видите ваше сообщение об ошибке, значит, вы сделали все правильно

Типы MIME

- Multipurpose Internet Mail Extensions (MIME)
- Многоцелевые Расширения Почты Интернета
 - стандарт, описывающий передачу различных типов данных по электронной почте, либо спецификация для форматирования нетекстовых сообщений таким образом, чтобы их можно было пересылать по Internet.
 - text/html, text/xml, text/plain, image/gif, image/png
- Глобальный список типов MIME
 - conf\mime.types
- Тип по умолчанию
 - DefaultType text/plain
- Добавление нового расширения файлов
 - AddType text/html .abc .xyz

Использование заголовков ответа

- Модуль для всех заголовков ответа сервера
 - LoadModule headers_module modules/mod_headers.so
- Модуль для заголовка Expires
 - LoadModule expires_module modules/mod_expires.so
- Header **set** X-My-Header "Value of header"
- Header **add** X-My-Header "Value of header"
- Header **unset** X-My-Header

Управление кэшированием ресурсов

- Запрет кэширования
 - Header set Cache-Control "no-cache"
 - Header set Cache-Control "no-store"
 - Header set Cache-Control "private, no-cache, must-revalidate"
 - ExpiresActive On
 - ExpiresDefault "now"
- Разрешение кэширования
 - Header set Cache-Control "public, max-age=600"
 - ExpiresActive On
 - ExpiresDefault "access plus 2 days"
 - ExpiresDefault "modification plus 3 hours"
- Кэширование по типу ресурсов
 - ExpiresByType text/html "access plus 2 days"
 - ExpiresByType image/gif "access plus 2 weeks"
- Вариант задания значения заголовку Expires
 - ExpiresDefault A3600
 - ExpiresByType image/png M3600

Привязка обработчиков

- CGI
 - ScriptAlias "/cgi-bin/" "/www/cgi-bin/"
 - AddHandler cgi-script .cgi .pl
- Модуль
 - LoadModule php5_module "/PHP/php5apache2_2.dll"
 - AddType application/x-httpd-php .php
- SSI
 - Options +Includes
 - AddOutputFilter INCLUDES .shtml
 - AddType text/html .shtml

Лабораторная работа 3.4

Привязка серверного обработчика

Содержание лабораторной работы 3.4

Привязка серверного обработчика

Упражнение 1: Привязка обработчика как модуля сервера

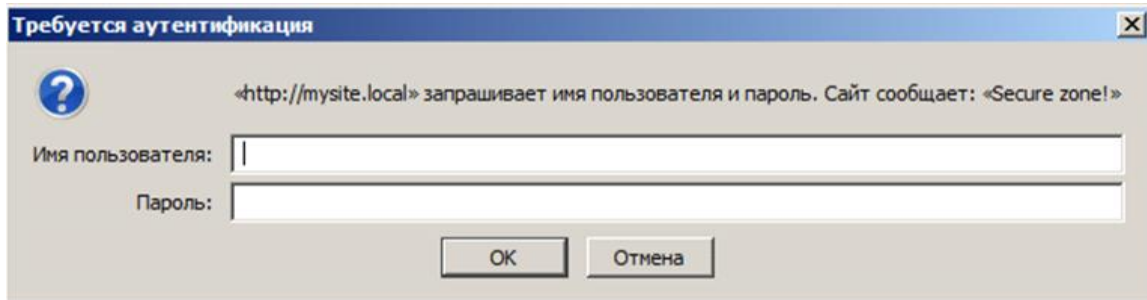
- Откройте конфигурационный файл сервера Apache **httpd.conf** в текстовом редакторе и в самом конце этого файла подключите файл **mysite\conf\mysite.conf**
- В самом конце этого файла допишите следующие строки:
PHP
LoadModule php7_module "путь-к-рабочей-папке/apps/php/php7apache2_4.dll"
AddType application/x-httpd-php .php
- Сохраните файл и перезапустите сервер
- В тестовом редакторе создайте новый файл
- Наберите следующий текст:
<?php phpinfo(); ?>
- Сохраните файл под именем **info.php** в папке **mysite\htdocs**
- В браузере наберите в адресной строке: <http://mysite.local/info.php>
- Если вы видите страницу с логотипом PHP, значит, вы сделали все правильно

Аутентификация и Авторизация

- RFC 2617
- HTTP/1.x 401 Unauthorized

...

WWW-Authenticate: Basic realm="www.mysite.local"



- GET / HTTP/1.1

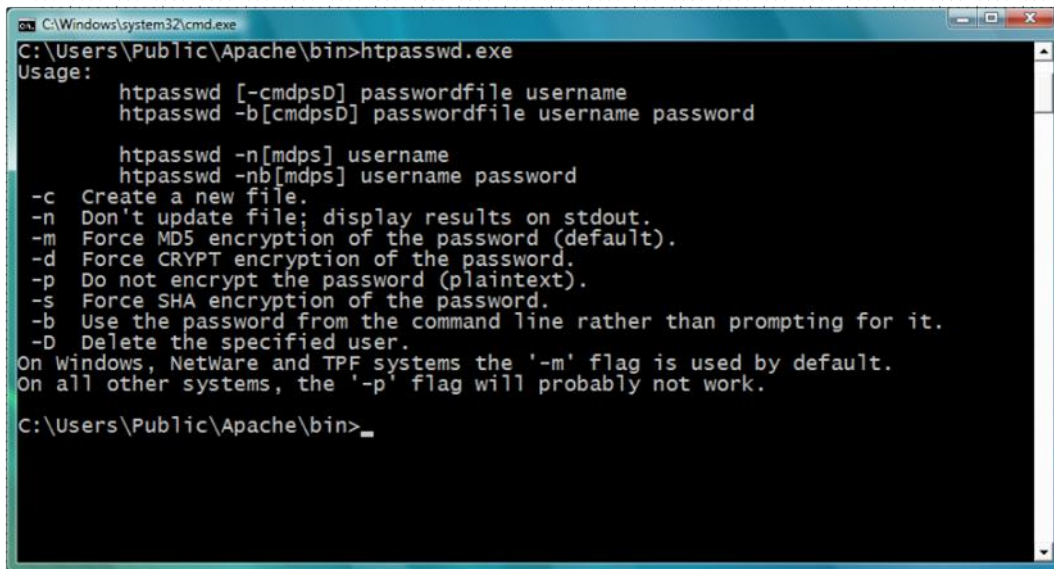
...

Authorization: Basic QWRtaW5pc3RyYXRvcjpwYXNzQHdvcmQx

Administrator:pass@word1

Создание учётной записи

- Файл **.htpasswd**
- Утилита **htpasswd.exe**



```
C:\Windows\system32\cmd.exe
C:\Users\Public\Apache\bin>htpasswd.exe
Usage:
    htpasswd [-cmdpsD] passwordfile username
    htpasswd -b[cmdpsD] passwordfile username password

    htpasswd -n[mdps] username
    htpasswd -nb[mdps] username password
-c Create a new file.
-n Don't update file; display results on stdout.
-m Force MD5 encryption of the password (default).
-d Force CRYPT encryption of the password.
-p Do not encrypt the password (plaintext).
-s Force SHA encryption of the password.
-b Use the password from the command line rather than prompting for it.
-D Delete the specified user.
On Windows, Netware and TPF systems the '-m' flag is used by default.
On all other systems, the '-p' flag will probably not work.
C:\Users\Public\Apache\bin>_
```

- Создание нового файла и первой записи
 - **htpasswd -c c:\secure\.htpasswd admin**
- Добавление пользователя
 - **htpasswd c:\secure\.htpasswd vasya**

Настройка базовой аутентификации

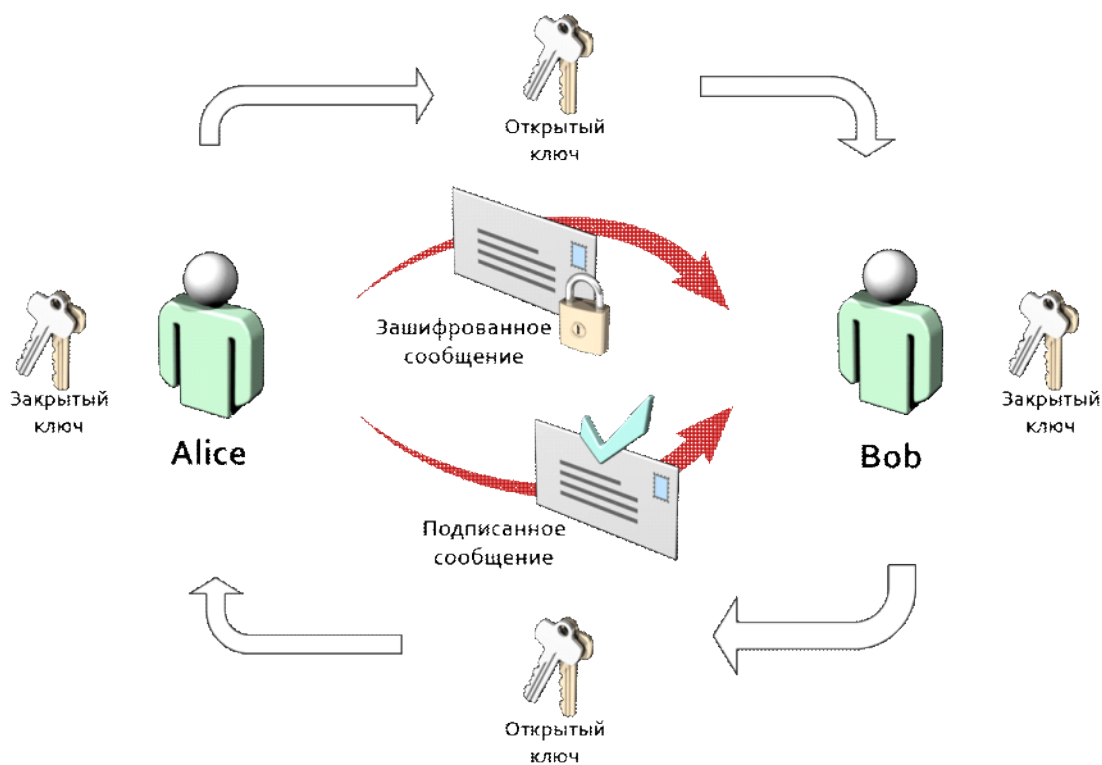
- Тип аутентификации
 - AuthType Basic
- Строка сообщения (realm)
 - AuthName "Запретная зона!"
- Файл с паролями. Полный физический путь!
 - AuthUserFile "c:\secure\.htpasswd"
- Включение защиты
 - Require valid-user

- Допуск конкретных пользователей
 - Require user admin vasya petya

- Файл группы пользователей
 - AuthGroupFile "c:\secure\.htgroup"
- Допуск пользователей группы
 - Require group admins managers

Защита трафика (SSL/TLS)

- Шифрование данных
 - невозможность перехвата
- Подтверждение сервера
 - невозможность подделки сервера
- Подтверждение клиента
 - аутентификация клиента



Сертификат сервера

- Создание самоподписанного сертификата
 - > cd conf
 - > openssl.exe
 - OpenSSL> req -config openssl.cnf -new -out mysite.csr
 - OpenSSL> rsa -in privkey.pem -out mysite.key
 - OpenSSL> x509 -in mysite.csr -out mysite.cert -req -signkey mysite.key -days 365
- LoadModule ssl_module modules/mod_ssl.so
- <VirtualHost 127.0.0.1:443>
 - SSLEngine On
 - SSLCertificateFile ../ssl/mysite.cert
 - SSLCertificateKeyFile ../ssl/mysite.key

Лабораторная работа 3.5

Базовая аутентификация пользователя с использованием SSL

Содержание лабораторной работы 3.5

Базовая аутентификация пользователя с использованием SSL

Упражнение 1: Создание файла пользователей

- Откройте окно командной строки
- Введите команду:
`htpasswd -c c:\.htpasswd admin`
и нажмите клавишу [ENTER]
- Введите пароль **password** и нажмите клавишу [ENTER]
- Подтвердите пароль **password** и нажмите клавишу [ENTER]
- Введите команду:
`htpasswd c:\.htpasswd vasya`
и нажмите клавишу [ENTER]
- Введите пароль **1234** и нажмите клавишу [ENTER]
- Подтвердите пароль **1234** и нажмите клавишу [ENTER]
- На диске **C:** у вас появился файл **.htpasswd**. Можете открыть его в текстовом редакторе и посмотреть его содержимое
- Перенесите файл **.htpasswd** в директорию **mysite/conf**

Упражнение 2: Включение аутентификации для отдельной папки

- Создайте в папке **mysite/htdocs** папку **secure**
- В текстовом редакторе создайте новый файл и наберите следующий текст:
`AuthType Basic
AuthName "Secure Folder"
AuthUserFile "c:\...\mysite\conf\.htpasswd"
Require valid-user`
- Сохраните файл в папке **mysite/htdocs/secure** под именем **.htaccess**
- Откройте браузер и наберите в адресной строке <http://www.mysite.local/secure/>
- Проверьте работу аутентификации пользователей

Упражнение 3: Ограничение доступа к файлам

- В папке **mysite/htdocs** создайте два архива с именами **test.zip** и **test.rar**
- В текстовом редакторе создайте новый файл и наберите следующий текст:
`# Запрет доступа к архивам zip и rar
<FilesMatch "\.(zip|rar)$">
 AuthType Basic
 AuthName "Secure File"
 AuthUserFile "c:\...\mysite\conf\.htpasswd"
 Require valid-user
</FilesMatch>`

- Сохраните файл в папке **mysite/htdocs** под именем **.htaccess**
- Откройте браузер и наберите в адресной строке <http://www.mysite.local/test.zip>
- Убедитесь, что на обращение к архивам требуется ввод пароля

Упражнение 4: Создание сертификата сервера

- Откройте окно командной строки и с помощью команды **cd** перейдите в папку **conf** сервера **Apache**
- В командной строке введите команду **openssl** и нажмите клавишу **[ENTER]** (Если появится предупреждение - не обращайте внимания)
- Введите следующую команду:
`req -config openssl.cnf -new -out mysite.csr`
и нажмите клавишу **[ENTER]**
- Далее вводите последовательно:
password нажмите клавишу **[ENTER]**
password нажмите клавишу **[ENTER]**
RU нажмите клавишу **[ENTER]**
нажмите клавишу **[ENTER]**
нажмите клавишу **[ENTER]**
нажмите клавишу **[ENTER]**
нажмите клавишу **[ENTER]**
mysite.local и нажмите клавишу **[ENTER]**
admin@mysite.local и нажмите клавишу **[ENTER]**
password нажмите клавишу **[ENTER]**
нажмите клавишу **[ENTER]**
- Введите следующую команду:
`rsa -in privkey.pem -out mysite.key`
и нажмите клавишу **[ENTER]**
- Введите **password** нажмите клавишу **[ENTER]**
- Введите следующую команду:
`x509 -in mysite.csr -out mysite.cert -req -signkey mysite.key -days 365`
и нажмите клавишу **[ENTER]**
- Скопируйте файлы **mysite.cert** и **mysite.key** из папки **Apache24/conf** в папку **mysite/conf**

Упражнение 5: Включение безопасного соединения

- Откройте конфигурационный файл сервера Apache **httpd.conf**
- Раскомментируйте строку подключающую модуль **ssl_module**
- Сохраните файл **httpd.conf**
- Откройте конфигурационный файл сайта **mysite/conf/mysite.conf**
- Добавьте директиву **Listen** со значением **443**
- Создайте новый виртуальный хост из уже имеющегося (Ctrl+C, Ctrl+V)
- В новом хосте вместо **127.0.0.1:80** укажите **127.0.0.1:443**
- Также в новом хосте добавьте:

```
SSLEngine On
SSLCertificateFile c:/.../mysite/conf/mysite.cert
SSLCertificateKeyFile c:/.../mysite/conf/mysite.key
```

- Сохраните файл
- Откройте браузер и наберите в адресной строке <https://mysite.local>
- На появившейся странице предупреждения об ошибке сертификата (не забывайте, сертификат самоподписанный, то есть выдан недоверенной организацией!) выберите (в зависимости от используемого браузера) действие, которое продолжит открытие этого веб-узла.
- Обратите внимание, браузер явно показывает, что используется безопасное соединение
- Проверьте работу SSL в других браузерах. Сравните предупреждения. Сравните способы указания на то, что используется безопасное соединение

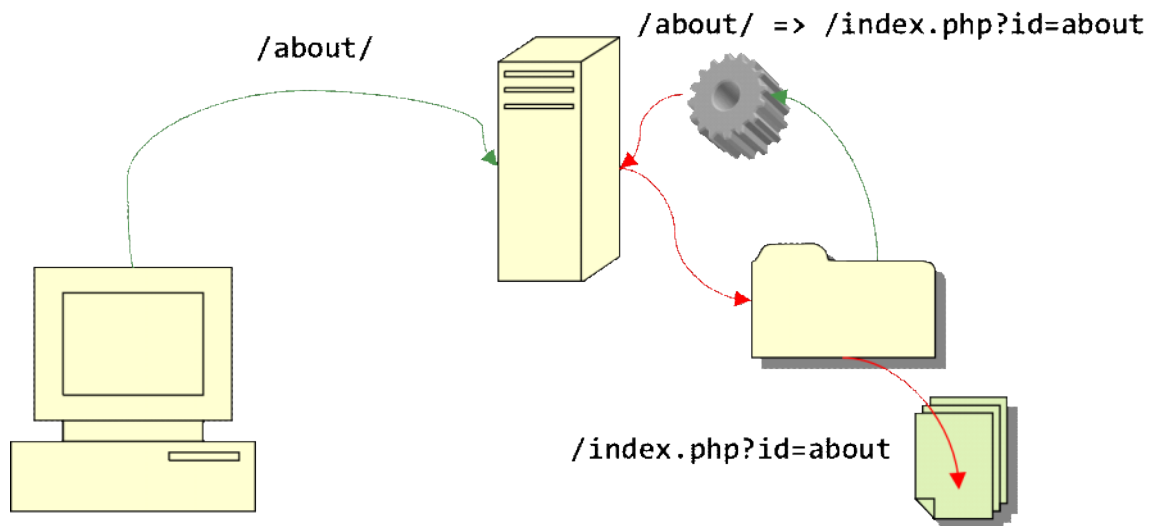
Регулярные выражения

- <FilesMatch "\.(zip|rar)\$">
- RedirectMatch 301 "/(.*)\.html" "[http://mysite.local/\\$1.php](http://mysite.local/$1.php)"
- Модификаторы
 - .
 - любой одиночный символ
 - [chars]
 - класс символов, один из
 - [^chars]
 - класс символов, ни один из
 - text|text
 - альтернатива: или
- Квантификаторы
 - {n,m}
 - от n до m
 - {n,}
 - от n до бесконечности
 - {n}
 - точно n
 - ?
 - 0 или 1
 - +
 - от 1 до бесконечности
 - *
 - 0 или сколько угодно
- Маркеры
 - ^
 - начало строки
 - \$

- конец строки
- Экранирование
 - `\char`

Модуль mod_rewrite

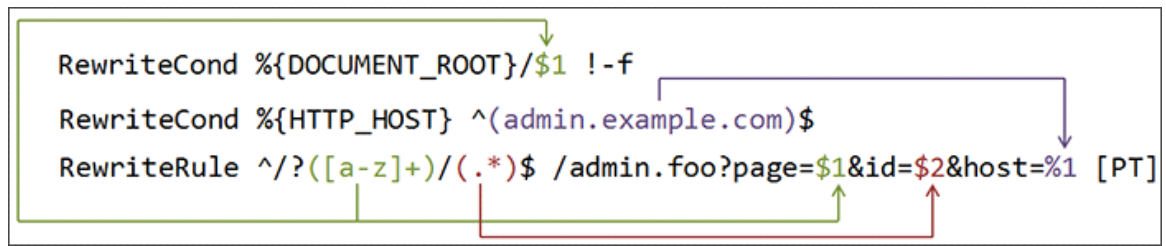
- LoadModule rewrite_module modules/mod_rewrite.so



Использование модуля mod_rewrite

- Используем модуль mod_rewrite в папке
 - RewriteEngine On
- Правила модуля mod_rewrite
 - RewriteRule Шаблон Замена/Подстановка [Флаги]
- Условия модуля mod_rewrite
 - RewriteCond Тест Условие [Флаги]
- Флаги
- NC
 - не учитывать регистр
- OR
 - либо следующее условие
- L
 - последнее правило
- R
 - перенаправление
- F
 - возвращает статус 403
- G
 - возвращает статус 410
- Варианты
 - RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)\$ index.php?q=\$1
 - RewriteCond %{HTTP_HOST} ^www\.site\.ru [NC]

RewriteRule ^(.*)\$ [http://site.ru/\\$1](http://site.ru/$1) [R=301]



Что мы изучили?

- Сервер Apache – самый распространенный сервер
- Возможность создавать виртуальные хосты и виртуальные директории
- Возможность настраивать сообщения об ошибках
- Возможность обрабатывать любые типы файлов
- Сервер Apache имеет большое число модулей
- Поддержка аутентификации пользователей
- SSL обеспечивает защиту коммуникаций
- Получение сертификата в доверенном центре сертификации гарантирует правильность данных в сертификате
- Расширение стандартного функционала с помощью модулей

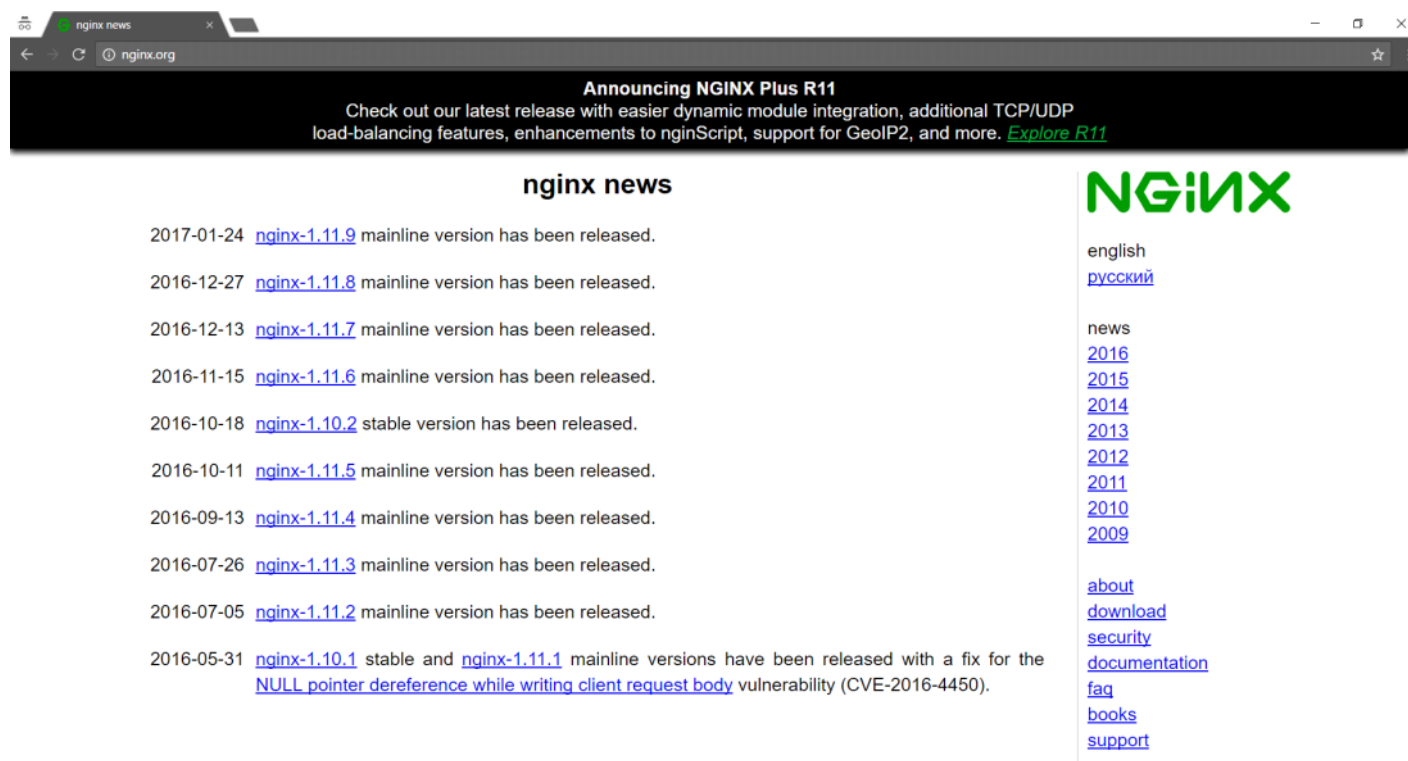
Модуль 4

Веб-мастеринг. Веб-сервер Nginx

Темы модуля

- Обзор и установка сервера Nginx
- Основные директивы настройки сервера Nginx
- Файлы конфигурации сервера
- Настройка директорий
- Создание виртуального хоста
- Настройка Nginx как кэширующего сервера
- Настройка Nginx как балансировщика нагрузки

Веб-сервер Nginx



Announcing NGINX Plus R11
Check out our latest release with easier dynamic module integration, additional TCP/UDP load-balancing features, enhancements to nginScript, support for GeolIP2, and more. [Explore R11](#)

nginx news

2017-01-24 [nginx-1.11.9](#) mainline version has been released.

2016-12-27 [nginx-1.11.8](#) mainline version has been released.

2016-12-13 [nginx-1.11.7](#) mainline version has been released.

2016-11-15 [nginx-1.11.6](#) mainline version has been released.

2016-10-18 [nginx-1.10.2](#) stable version has been released.

2016-10-11 [nginx-1.11.5](#) mainline version has been released.

2016-09-13 [nginx-1.11.4](#) mainline version has been released.

2016-07-26 [nginx-1.11.3](#) mainline version has been released.

2016-07-05 [nginx-1.11.2](#) mainline version has been released.

2016-05-31 [nginx-1.10.1](#) stable and [nginx-1.11.1](#) mainline versions have been released with a fix for the [NULL pointer dereference while writing client request body](#) vulnerability (CVE-2016-4450).

NGINX

[english](#)
[русский](#)

news
[2016](#)
[2015](#)
[2014](#)
[2013](#)
[2012](#)
[2011](#)
[2010](#)
[2009](#)

[about](#)
[download](#)
[security](#)
[documentation](#)
[faq](#)
[books](#)
[support](#)

Основные файлы и директории

- nginx.exe
- html
 - корневая папка по умолчанию
- conf
 - конфигурация сервера
 - nginx.conf
 - mime.types

Управление сервером

- Запуск из командной строки
 - nginx.exe
- Управление запущенным сервером
 - nginx -s stop
 - nginx -s quit
 - nginx -s reload

Правила конфигурации

- # Однострочный комментарий
- Однострочные директивы
 - ИмяДирективы Значение [Значение2 [...]];
- Блочные директивы
 - ИмяДирективы [Значение]{
ИмяДирективы Значение
}
- Физические пути в ОС Windows прописываются с прямым слэшем:
 - C:/Users/Public/Folder

Обзор основных директив

- Журналирование
 - `error_log logs/error.log;`
 - `log_format main '$remote_addr - [$time_local] "$request"';`
 - `access_log logs/access.log main;`
- Подключение файлов
 - `include mime.types;`
- Доступ
 - `deny all;`
 - `allow 127.0.0.1;`
- Документы
 - `root document_root;`
 - `index index.html index.htm;`
 - `autoindex off;`
 - `default_type application/octet-stream;`
- Файлы ошибок
 - `error_page 404 /404.html;`
- Заголовки
 - `add_header Cache-Control no-store;`
 - `expires 24h;`
 - `expires modified +24h;`

Настройка виртуального хоста

```
server{
    listen 80; # 127.0.0.1:80
    server_name localhost;
    location / {
        root www; # Корневая папка
        index index.html # Файл по умолчанию
    }
    error_page 500 502 503 504 /50x.html;
    # Виртуальная директория
    location /foo/{
        alias /some/path/
    }
    # Перенаправление
    location /bar/{
        rewrite ^(.*)$ http://some-site.com redirect #302
    }
    location /baz/{
        rewrite ^(.*)$ http://other-site.com permanent #301
    }
    location /deleted.html{
        return 410;
    }
    location /file.html{ # для /file.html?lock=1
        if($query_string ~ ^lock=1){
            return 403;
        }
        if($arg_lock ~ "0"){
            return 402;
        }
    }
}
```

```
}  
}  
}
```

Описание директорий

- `location = / { # точное совпадение
конфигурация А
}`
- `location / {
конфигурация Б
}`
- `location /documents/ {
конфигурация В
}`
- `location ^~ /images/ { # регулярные выражения не проверяются
конфигурация Г
}`
- `location ~* \.(gif|jpg|jpeg)$ { # регулярные выражения
конфигурация Д
}`
- `/`
 - А
- `/index.html`
 - Б
- `/documents/doc.html`
 - В
- `/images/1.gif`
 - Г
- `/documents/1.jpg`
 - Д
- `error_page 405 =200 @dir`
- `location @dir{
root /some_root`

} ...

Лабораторная работа 4.1

Создание и настройка виртуального хоста

Содержание лабораторной работы 4.1

Создание и настройка виртуального хоста

Упражнение 1: Запуск веб-сервера Nginx

- С помощью сетевой утилиты **netstat** убедитесь, что порт **80** никем не занят. В противном случае - освободите его
- Из командной строки с помощью команды **nginx** запустите веб-сервер
- Для удобства запуска можете добавить путь к файлу **nginx.exe** в системную переменную **PATH** (см. Лабораторную работу 3.1)
- Откройте браузер, наберите в адресной строке: <http://localhost> и убедитесь, что веб-сервер запущен

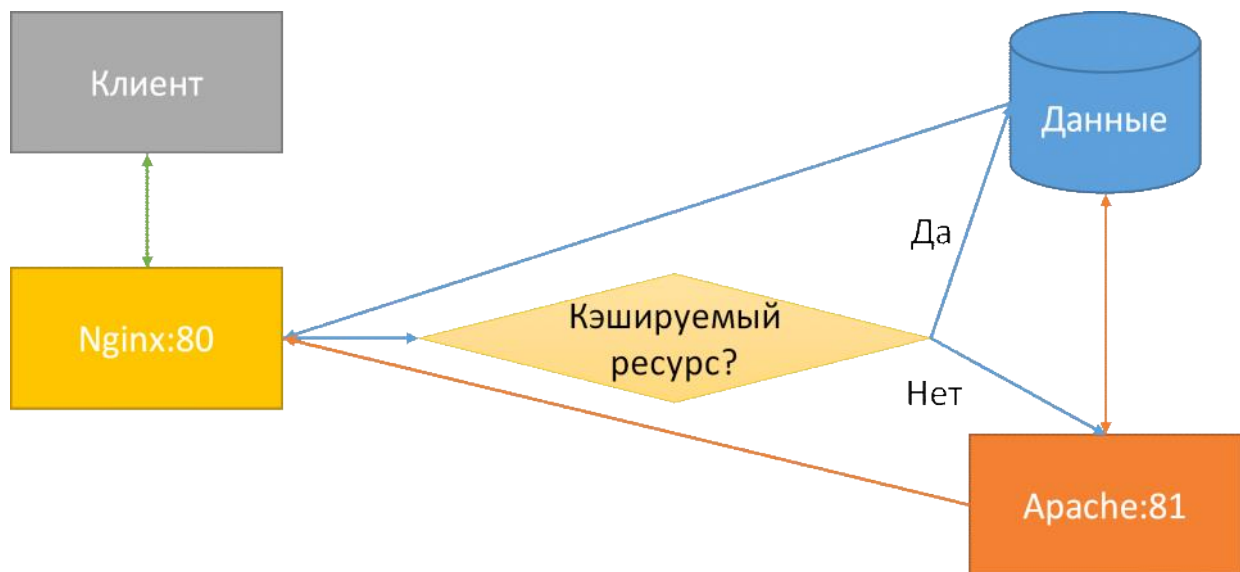
Упражнение 2: Создание виртуального хоста

- В текстовом редакторе создайте новый файл и напишите в нём:
Виртуальный хост **mysite.local** для Nginx
- Сохраните файл под именем **mysite.nginx.conf** в папке **mysite\conf**
- В данном файле виртуальный хост для **127.0.0.1:80**
- Задайте **server_name** для хостов **mysite.local**, www.mysite.local и **info.mysite.local**
- Определите файлы для **error_log** и **access_log**. Файлы должны находиться в папке **mysite\logs**
- Определите корневую папку сайта **mysite\htdocs** и опишите её
- Откройте конфигурационный файл сервера Nginx **nginx.conf** в текстовом редакторе и в самом конце этого файла (перед последней закрывающей фигурной скобкой) подключите файл **mysite\conf\mysite.nginx.conf** с помощью директивы **include**
- Также в файле **nginx.conf** найдите директиву **log_format** (где-то около 20-й строки) и раскомментируйте её. Обратите внимание, что директива занимает несколько строк
- Не забудьте сохранить все файлы

Упражнение 3: Проверка работы виртуального хоста

- Перезапустите сервер Nginx с помощью сигнала **reload**
- Откройте браузер, наберите в адресной строке: <http://mysite.local> и убедитесь, что изменения применились корректно
- Обращайте внимание на сообщения сервера: если где-то в конфигурации ошибка, то найдите её и исправьте

Кэширующий сервер



```
server{  
    listen 80;  
    server_name mysite.local;  
    location ~* \.(jpg|jpeg|png|gif|ico|css|js)$ {  
        root /apache/htdocs;  
    }  
  
    location / {  
        proxy_pass http://127.0.0.1:81/;  
        proxy_set_header Host $host  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header X-Forwarded-For $remote_addr;  
    }  
}
```

Лабораторная работа 4.2

Использование Nginx в качестве кэширующего сервера

Содержание лабораторной работы 4.2

Использование Nginx в качестве кэширующего сервера

Упражнение 1: Настройка веб-сервера Apache

- Откройте конфигурационный файл сервера Apache **httpd.conf** в текстовом редакторе
- В самом низу файла прокомментируйте строку, которая подключает файл **mysite.conf**
- Поменяйте значение директивы **Listen** на **81** (находится около строки 58)
- Поменяйте значение директивы **DocumentRoot** на **mysite/htdocs** (находится около строки 243)
- Поменяйте значение директивы **Directory** на **mysite/htdocs** (находится около строки 244)
- Сохраните файл
- Запустите Apache из командной строки с помощью команды **httpd**
- Откройте браузер, наберите в адресной строке: <http://localhost:81> и убедитесь, что веб-сервер запущен

Упражнение 2: Настройка веб-сервера Nginx

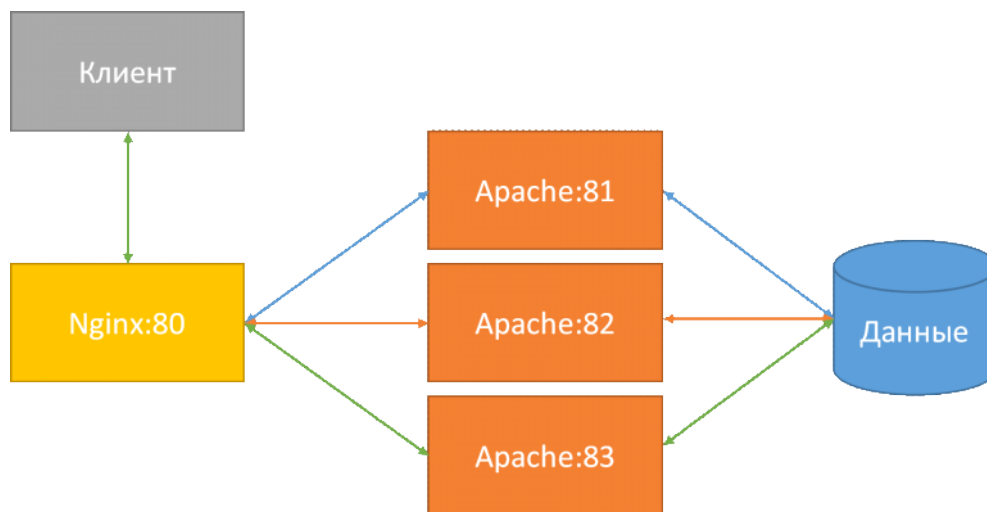
- В текстовом редакторе откройте файл **mysite/mysite.nginx.conf**
- Удалите имеющуюся блочную директиву **location**
- В блочной директиве **server** пропишите:

```
location ~* \.(jpg|jpeg|gif|png|ico|css|js)$ {  
    root    c:\...\mysite\htdocs;  
}  
  
location / {  
    proxy_pass http://127.0.0.1:81;  
    proxy_set_header Host $host  
    proxy_set_header X-Real-IP $remote_addr;  
    proxy_set_header X-Forwarded-For $remote_addr;  
}
```
- Сохраните файл

Упражнение 3: Проверка работы серверов

- Перезапустите сервер Nginx с помощью сигнала **reload**
- Откройте браузер, наберите в адресной строке: <http://mysite.local> и убедитесь, что изменения применились корректно
- Положите в папку **mysite/htdocs** изображение и запросите его через браузер.
- Проверьте логи сервера Apache (**access.log** в папке **Apache24/logs**). Вы должны увидеть, что Nginx берёт картинку не запрашивая её у Apache

Балансировщик нагрузки



```
upstream apache{
    server 127.0.0.1:81 weight=5;
    server 127.0.0.1:82;
    server 127.0.0.1:83;
    server 127.0.0.1:84 down;
}
server{
    listen 80;
    server_name mysite.local;
    location / {
        proxy_pass http://apache;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $remote_addr;
    }
}
```

Лабораторная работа 4.3

Использование Nginx в качестве балансировщика нагрузки

Содержание лабораторной работы 4.3

Использование Nginx в качестве балансировщика нагрузки

Упражнение 1: Установка и настройка второго экземпляра веб-сервера Apache

- Остановите сервер Apache (Ctrl+C)
- Сделайте копию папки **Apache24** и переименуйте её в **Apache24_alternate**
- Откройте конфигурационный файл сервера **Apache24_alternate httpd.conf** в текстовом редакторе
- Поменяйте все вхождения подстроки "**Apache24**" на "**Apache24_alternate**"
- Поменяйте значение директивы **Listen** на **82** (находится около строки 58)
- Сохраните файл
- Запустите Apache из командной строки с помощью команды **httpd**
- Откройте браузер, наберите в адресной строке: <http://localhost:82> и убедитесь, что веб-сервер запущен
- Запустите **первый** Apache из командной строки с помощью команды **httpd**
- Откройте браузер, наберите в адресной строке: <http://localhost:81> и убедитесь, что веб-сервер запущен

Упражнение 2: Настройка веб-сервера Nginx

- В текстовом редакторе откройте файл **mysite/mysite.nginx.conf**
- Перед блочной директивой **server** пропишите:

```
upstream apache {  
    server 127.0.0.1:81 weight=2;  
    server 127.0.0.1:82;  
}
```
- В блочной директиве **location** поменяйте значение директивы **proxy_pass** на <http://apache>
- Сохраните файл

Упражнение 3: Проверка работы серверов

- Перезапустите сервер Nginx с помощью сигнала **reload**
- Откройте браузер, наберите в адресной строке: <http://mysite.local> и убедитесь, что изменения применились корректно
- Проверьте логи обоих серверов Apache. Вы должны увидеть, что Nginx посылает запросы на оба сервера

Что мы изучили?

- Сервер Apache – самый распространенный сервер после сервера Apache
- Сервер Nginx очень быстрый
- Сервер Nginx поддерживает весь функционал стандартного веб-сервера
- Nginx может выступать в роли кэширующего сервера
- Nginx может выступать в роли балансировщика нагрузки
- Самая распространенная связка "frontend/backend" - Nginx/Apache

Что почитать?

- RFC 7231
- Документация веб-сервера Apache
- Документация веб-сервера Nginx

Что дальше?

- Курсы направления веб-технологий