**Работа с Web-сервером Russian Apache**

**Русский Apache**

Самый распространенный Web-сервер в мире - это Apache. По данным компании Netcraft (http://www.netcraft.com/ Survey/) общее число Web-узлов, работающих под его управлением, к концу 1998 г. достигло 2 млн. (55% общего числа узлов) и постоянно растет. Для сравнения: на долю серверов Microsoft приходится 25%, Netscape -7%. Будучи бесплатной открытой программой, предназначенной для бесплатных же Unix-систем (FreeBSD, Linux и др.), Apache по функциональным возможностям и надежности не уступает коммерческим серверам, а широкие возможности конфигурирования позволяют настроить его для работы практически с любой конкретной системой. Существуют локализации сервера для различных языков, в том числе и для русского.

Исторически сложилось так, что русские тексты в Internet могут быть представлены в разных кодировках, из которых наиболее распространены koi8-r (или просто koi8) и Windows-1251: с первой работает большинство серверов и рабочих станций под управлением Unix, вторая является стандартной для всех версий Windows. Поскольку кодировка Windows-1251, естественно, применяется на подавляющем большинстве клиентских машин, доля тех, кто путешествует по русской части WWW, используя koi8, не превышает сейчас 5%. Однако в этой кодировке хранятся документы на многих Unix-серверах, в ней чаще всего передаются почтовые сообщения и практически всегда - письма в телеконференции, с ней же работают многие русскоязычные каналы IRC (кстати, аббревиатура КОИ расшифровывается как "код обмена информацией"). Чтобы решить проблемы, возникающие при несовпадении кодировок текста на сервере и клиентской машине, и был создан русский модуль Apache-RUS для Web-сервера Apache.

В статье мы рассмотрим процесс установки и настройки как самого сервера, так и механизма перекодирования документов "на лету".

**Установка**

Свежую версию Apache-RUS можно получить по адресу ftp://apache.lexa.ru/pub/apache-rus/ ("старшая" часть номера версии, например 1.3.3, соответствует версии оригинального Apache, "младшая", например PL27.3, - так называемому patch level, т. е. версии русского модуля). Рекомендуется устанавливать те версии, которые зарекомендовали себя как "стабильные". Здесь настройка сервера описывается на примере Apache\_1.3.3rusPL27.3.

Итак, первым делом мы переписываем на свою машину архив (менее 1,5 Мбайт) и распаковываем его:

# ftp ftp://apache.lexa.ru/pub/apache-rus/ apache\_1.3.3rusPL27.3.tar.gz

# tar xvzf apache\_1.3.3rusPL27.3.tar.gz

После этого входим в созданный при распаковке каталог apache\_1.3.3rusPL27.3 и запускаем сценарий configure:

# cd apache\_1.3.3rusPL27.3

# ./configure

При необходимости сценарию можно в явной форме указать аргументы (их список выдается по команде configure -help). Так, если требуется установить сервер в иной каталог, нежели стандартный, нужно выполнить "configure -prefix=<path-to-apache>".

Когда configure отработает, следует, как обычно, дать команды make и make install (эти действия выполняются пользователем root).

# make

# make install

Теперь сервер установлен в каталоге /usr/local/apache, но запускать его пока нельзя - сначала мы должны отредактировать файлы настройки httpd.conf, access.conf и srm.conf в каталоге /usr/local/apache/etc/ (начиная с версии 27.4 - /usr/local/apache/conf).

**Настройка**

Настройка конфигурационных файлов Web-сервера - самый ответственный шаг при его установке. Здесь мы рассмотрим только наиболее распространенные директивы и их параметры, поскольку полный перечень с описанием займет не один десяток страниц. Сервер перечитывает конфигурационные файлы при запуске, а также при получении сигнала -HUP (жесткий рестарт) или -uSR1 (мягкий рестарт). Если сервер находится в рабочем состоянии, то при изменении конфигурации его рекомендуется перезапустить командой

# kill -USR1 `cat /usr/local/apache/logs/httpd.pid`

В этом случае имеющиеся соединения не закрываются принудительно и завершаются обычным образом, а следующие клиенты работают уже с новыми конфигурационными файлами.

**Файл access.conf**

В access.conf содержатся директивы, описывающие права доступа к каталогам и файлам Web-сервера. Прежде всего решите, в каком каталоге будут храниться документы. По умолчанию это /usr/local/apache/share/htdocs, однако многие администраторы предпочитают размещать документы начиная с каталога /www/<имя\_сервера>/, поскольку при такой организации проще ориентироваться в структуре файлов. Пусть, например, мы создали каталоги:

/www/rmt.ru/

/www/radio-msu.net/

/www/people.radio-msu.net/

Они будут корневыми для соответствующих виртуальных серверов.

Файл access.conf может содержать секции Directory, Location и Files, которые ограничены одноименными директивами. В параметрах этих директив могут использоваться символы "?" и "\*" , а также регулярные выражения, предваряемые тильдой, например <Directory ~"^/www.+/a?server">. В секции Directory помещаются инструкции, относящиеся к определенному каталогу на диске, в секции Location - относящиеся к виртуальному пути, в секции Files - относящиеся к файлу или группе файлов.

<Directory /www/rmt.ru>

# директивы, относящиеся ко всем документам, хранящимся в

каталоге /www/rmt.ru и вложенных в него

</Directory>

<Location /cgi-bin>

# директивы, относящиеся ко всем документам, доступным по

адресу http://<имя\_сервера>/cgi-bin/ <путь\_к\_файлу>

</Location>

<Files /www/rmt.ru/form.html>

# директивы, относящиеся к файлу form.html из каталога

/www/rmt.ru

</Files>

Различие между секциями Directory и Location состоит в том, что первая относится к каталогам на диске, вторая - к виртуальному пути (URL), который браузер запрашивает у Web-сервера. И в той, и в другой могут присутствовать директивы order, allow и deny, которые позволяют ограничить доступ к каталогу или URL с различных машин.

Следующие две директивы относятся к секции <Directory>.

Options [options ...]

Возможные значения параметров:

ExecCGI - разрешить выполнение CGI-сценариев в данном каталоге и его поддереве;

FollowSymLinks - разрешить переходы по символическим ссылкам (создаваемым командой ln);

Includes - разрешить SSI (Server Side Includes);

Indexes - разрешить выдачу листинга каталога, если в нем нет файла index.html (или файла индекса, заданного директивой DirectoryIndex);

MultiViews - разрешить поддержку многих языков; по умолчанию она отключена, и включать ее, как правило, не нужно; поддержка перекодирования "на лету" для русского языка устанавливается с помощью других директив, которые мы рассмотрим позже;

All - установить сразу все перечисленные режимы кроме MultiViews.

При отсутствии специальных требований к безопасности вполне допустимо указать "Options All" в секции <Directory /www>; в противном случае нужно описать параметры каждого каталога отдельно.

AllowOverride [options ...]

Большинство директив могут задаваться не только в конфигурационных файлах сервера, но и в файлах .htaccess в каталогах сервера. Директива AllowOverride определяет набор директив, допустимых в файлах .htaccess. Параметры могут быть указаны следующие:

AuthConfig - разрешить установку авторизации по имени пользователя и паролю;

FileInfo - разрешить директивы, отвечающие за типы документов;

Indexes - разрешить директивы, связанные с листингом каталогов;

Limit - разрешить команды allow и deny, которые ограничивают доступ к файлам в зависимости от адреса клиентского компьютера;

Options - разрешить описанную выше директиву Options.

Учтите, что при включении последнего режима пользователи получают возможность создавать собственные файлы .htaccess и разрешать в них выполнение CGI-сценариев. Поэтому если нужно контролировать CGI-сценарии пользователей, не следует распространять на пользовательские каталоги действие директивы AllowOverride Options.

Однако во многих случаях (в частности, когда права на изменение содержимого сервера есть только у администратора) файл access.conf может выглядеть так, как в листинге 1.

Файл srm.conf

Файл srm.conf содержит директивы, связанные с общими настройками структуры каталогов сервера. Как правило, в нем достаточно изменить лишь несколько строк.

DocumentRoot <первый каталог сервера>

Путь к каталогу по умолчанию, индексный файл которого пользователь получит при обращении к серверу (http://<имя\_сервера>/). Эту директиву следует задать и для каждого из виртуальных серверов (в секции <VirtualHost> файла httpd.conf).

UserDir <имя пользовательского каталога>

Каталог, в котором пользователи должны размещать свои файлы, чтобы они были доступны по адресу http://<имя\_сервера>/~<имя\_пользователя>/. Стандартно public\_html. Иногда, чтобы облегчить жизнь пользователям, администраторы дают директиву "UserDir www".

DirectoryIndex <список файлов индекса>

Файл индекса - это тот файл, который будет передан клиенту при обращении к каталогу. Если указать несколько имен, сервер будет искать подходящий файл "слева направо". По умолчанию список содержит всего одно имя - index.html, но принято добавлять в него и другие распространенные имена индексных файлов. Например, директива может иметь вид: DirectoryIndex .index.html index.html index.htm index.cgi index.shtml home.html home.htm default htm default html

Чтобы включить на сервере поддержку CGI-сценариев, следует убрать знак комментария перед директивами ScriptAlias и AddHandler cgi-script .cgi. Первая задает каталог на диске, в котором будут храниться исполняемые программы, а вторая определяет, что все файлы с расширением .cgi должны обрабатываться как сценарии.

Директива ErrorDocument позволяет заменять стандартные сообщения сервера об ошибках на свои. Например, в случае самой распространенной ошибки - 404 (файл не найден) - считается хорошим тоном выдавать пользователю страницу с предложением продолжить свой путь по серверу или форму для поиска по узлу. Реализуется это достаточно просто: в настройках сервера мы убираем знак комментария со строки

ErrorDocument 404 /missing.html

B корневом каталоге каждого виртуального сервера создаем файл missing.html. Рекомендуется дать в нем ссылки на основные разделы сервера - и для удобства пользователей, и для того, чтобы предоставить необходимую информацию поисковым роботам, индексирующим серверы.

Файл httpd.conf

Конфигурационный файл httpd.conf является основным и содержит настройки, связанные с работой Web-сервера, виртуальных серверов, а также всех его программных модулей. Кроме того, именно в нем настраивается перекодирование русских букв при передаче от сервера к клиенту и обратно.

Директива Port, помещенная в самом начале файла, определяет номер порта для http-сервера; по умолчанию это 80. При необходимости можно приписать серверу другой порт или несколько портов, для чего служит директива Listen.

Директива HostnameLookups с параметром on или off включает или, соответственно, отключает преобразование численных IP-адресов клиентов, получивших документы с сервера, в доменные имена. Такое преобразование несколько замедляет работу сервера, но при числе посещений менее 10 000 в сутки это, как правило, практически не заметно.

Директивы User и Group задают пользователя, который будет администрировать сервер. С точки зрения безопасности нежелательно указывать здесь существующего пользователя, имеющего доступ к каким-либо другим ресурсам или файлам. Лучше создать отдельного пользователя и группу специально для http-сервера, например:

User www

Group www

Директивы ServerRoot, ErrorLog, CustomLog определяют соответственно корневой каталог http-сервера, путь к журналу регистрации ошибок (error\_log) и путь к общему журналу обращений к серверу (access\_log).

Директива CacheNegotiatedDocs разрешает кэширование документов, полученных с сервера. По умолчанию этот режим отключен, но, поскольку пропускная способность отечественных Internet-каналов еще долго будет оставлять желать лучшего, хорошо бы его включить: тогда пользователю не придется ждать загрузки картинок при каждом обращении к вашей странице.

**Настройка виртуальных серверов в файле httpd.conf**

В большинстве случаев один http-сервер способен обрабатывать запросы, поступающие на различные, так называемые виртуальные, Web-серверы. Виртуальные серверы могут иметь как один и тот же IP-адрес, но разные доменные имена, так и разные IP-адреса. С точки зрения пользователя второй вариант чуть более предпочтителен, поскольку запрос к серверу, отличающемуся от основного только доменным именем, должен содержать его имя, а некоторые старые браузеры, не поддерживающие протокол HTTP/1.1 (например, Microsoft Internet Explorer 2.0), не включают в запрос эту информацию. Однако такие браузеры выходят из употребления (сейчас их уже менее 0,5% общего числа); с другой стороны, выделение собственного IP-адреса каждому виртуальному серверу может быть неоправданной растратой адресного пространства компании.

Для описания адресов и доменных имен виртуальных серверов служат директивы ServerName, ServerAlias, NameVirtualHost и VirtualHost. Они необходимы, только если вам нужно установить более одного виртуального сервера.

В листинге 2 приведен фрагмент конфигурационного файла для случая виртуальных серверов с различными IP-адресами, в листинге 3 - аналогичный фрагмент для случая, когда серверы различаются только доменным именем.

Директива ServerName, находящаяся вне секций VirtualHost, определяет имя основного сервера, т. е. сервера, корневой каталог которого задан директивой DocumentRoot в файле srm.conf. Виртуальные серверы наследуют настройки основного; при необходимости специальной настройки соответствующие директивы помещаются в секции VirtualHost, относящейся к данному серверу. Допустимы любые директивы, которые могут встретиться в файлах httpd.conf и srm.conf, например DocumentRoot, ErrorLog, CustomLog, Location, ServerAdmin.

Из листинга 3 видно, как используется директива ServerAlias, если необходимо создать несколько виртуальных серверов с одинаковым содержанием. После того как вы занесете в конфигурационные файлы информацию об имеющихся на диске виртуальных серверах (разумеется, они должны быть описаны и в конфигурационных файлах DNS), можно приступить к последнему шагу настройки Apache-RUS.

Настройка перекодирования русскоязычных документов

Модуль поддержки русских кодировок был разработан в 1996 г. Дмитрием Крюковым (dvk@stack.net), а с февраля 1997 г. поддерживается рабочей группой Apache-RUS Team во главе с Алексеем Тутубалиным (lexa@ lexa.ru). За время своего развития модуль претерпел множество изменений и теперь обладает практически неограниченными возможностями настройки для любой конкретной конфигурации.

Инструкции, отвечающие за перекодирование, разделяются естественным образом на три группы. К первой относятся две директивы, указывающие, в какой кодировке хранятся файлы на диске: CharsetSourceEnc <кодировка> и CharsetByExtension <кодировка> <расширение1> <расширение2>...

Например, файл httpd.conf может содержать строки:

CharsetSourceEnc koi8-r

CharsetByExtension windows-1251 .txt

Такая запись означает, что все файлы хранятся на диске в кодировке koi8-r; исключение составляют текстовые файлы с расширением txt, для которых используется Windows-1251.

Если кодировок более одной и документы в каждой кодировке хранятся в своем каталоге, директивы CharsetSourceEnc помещаются в соответствующие секции <Location> либо в файлы .htaccsess внутри каталогов.

Вторую группу составляют директивы CharsetDecl, CharsetAlias CharsetRecodeTable и CharsetWideRecode Table, которые определяют названия кодировок, их синонимы и таблицы перекодирования. Все они размещаются в секции <IfModule mod\_charset.c> - </IfModule> и в большинстве случаев не нуждаются в изменении.

В третью, самую многочисленную группу входят директивы, задающие порядок перекодирования символов от сервера клиенту и обратно.

Принято, чтобы при попадании на русскоязычный сервер пользователь получал страницу в "своей" кодировке, определяемой автоматически на основе той информации об операционной системе, которую передает серверу браузер: например, установив, что пользователь работает в Windows, сервер выдает ему страницу в кодировке Windows-1251, а установив, что он работает в Unix, выдает страницу в koi8. Если выбранная таким образом страница не подходит, клиент может сменить кодировку вручную. Основных схем выбора три: по префиксу каталога, по имени виртуального сервера и по номеру порта. У каждой из них есть свои преимущества и свои недостатки.

1) http://www.rmt.ru/koi/document.html

http://www.rmt.ru/win/document.html - выбор кодировки по префиксу каталога,

2) http://koi.www.rmt.ru/document.html

http://win.www.rmt.ru/document.html - выбор кодировки по имени сервера,

3) http://www.rmt.ru:8000/document.html

http://www.rmt.ru:8001/document.html - выбор кодировки по порту.

Для организации выбора кодировки по префиксу каталога нужно либо внести в секцию VirtualHost строку вида

Alias /koi /www/rmt

либо создать в соответствующем каталоге символическую ссылку на себя:

# cd /www/rmt

# ln -s . koi

Усилия, затрачиваемые на первоначальное конфигурирование, невелики, но для крупных серверов с разветвленной структурой такая схема не очень подходит: вряд ли удастся проконтролировать корректность ссылок на разные страницы узла с внешних серверов, да и за внутренними ссылками проследить не так-то просто (в большинстве случаев они должны быть относительными).

При выборе кодировки по имени сервера необходимо, чтобы информация о соответствующих именах была задана в настройках DNS-сервера, обслуживающего данный домен, а в файл httpd.conf в секцию VirtualHost вносятся строки:

<VirtualHost 193.124.134.3>

ServerName www.rmt.ru

ServerAlias \*.www.rmt.ru

...

</VirtualHost>

Если в качестве имени поддомена выступает один из синонимов названия кодировки (CharsetAlias), то эта кодировка считается кодировкой клиента. При таком подходе ссылки внутри сервера могут быть любыми, и единственный недостаток данной схемы в том, что перекодирование не выполняется для браузеров, не указывающих в запросе имя сервера, - впрочем, их, как уже говорилось, осталось крайне мало. Если же совместимость со старыми браузерами категорически необходима, можно назначить каждому поддомену свой IP-адрес.

Чтобы применить выбор по номеру порта, необходимо в файле httpd.conf удалить директиву Port и снять комментарии со строк

Listen 80

Listen 8100

Listen 8101

Listen 8102

Listen 8103

CharsetByPort koi8-r 8100

CharsetByPort windows-1251 8101

CharsetByPort ibm866 8102

CharsetByPort iso-8859-5 8103

Номера портов не очень важны. В стандартной настройке Apache-RUS нумерация, как видим, начинается с 8100, но чаще ее начинают с 8000 или 8080.

Данная схема не требует внесения дополнительных записей в DNS и позволяет работать с виртуальными серверами даже клиентам, которые не поддерживают протокол HTTP/1.1, - ведь кодировка выбирается исходя из числа, указывающего на номер порта Web-сервера (по умолчанию это 80). Однако сетевые брандмауэры иногда запрещают работу с определенными портами, и если таким брандмауэром защищена сеть клиента, он не сможет установить соединение с вашим сервером. К сожалению, подобная ситуация возникает чаще, чем хотелось бы.

Схема выбора кодировки задается директивой CharsetSelectionOrder. Ее параметры определяют порядок применения правил выбора. Так, выбору по префиксу каталога соответствует строка

CharsetSelectionOrder Dirprefix Useragent Portnumber Hostname UriHostname

Выбору по имени домена - строка

CharsetSelectionOrder Hostname UriHostname Useragent Portnumber Dirprefix

Для выбора по номеру порта следует записать

CharsetSelectionOrder Portnumber Useragent Hostname UriHostname Dirprefix

Замечания

Чтобы документы, кодировка которых была выбрана автоматически, не оседали в кэшах прокси-серверов, Apache-RUS дает им специальный HTTP-заголовок, запрещающий кэширование. В результате при возврате на страницу (например, по кнопке Back) она считывается с сервера заново, что, во-первых, замедляет работу, а во-вторых (и это более серьезная проблема) очищает все текстовые формы, которые были на странице (то же происходит при использовании JavaScript). Разрешить кэширование позволяет директива CharsetDisableForcedExpires On, которая задается в секции <Location> для данного виртуального пути или в соответствующем файле .htaccess, но тогда возникает риск, что пользователи иногда будут получать страницы в "чужой" кодировке. Существуют и промежуточные варианты: например, можно установить CharsetDisableForcedExpires On (в секции <Files>) только для тех документов, которые содержат формы, окна или JavaScript-сценарии.

Для полного отключения перекодирования в каталоге или на виртуальном сервере служит директива Charset Disable On.

При выборе кодировки по имени сервера или по префиксу каталога хорошим тоном является использование для графических файлов абсолютных ссылок с указанием имени сервера (например, <img src="http://images.rmt.ru/ picture.jpg">). Тогда при переходе клиента от основного сервера к выбранной кодировке изображения будут браться из локального кэша браузера, а не перечитываться заново. Это особенно актуально при большом объеме графической информации на сервере.

**Запуск сервера**

По окончании процедуры настройки следует запустить httpd-сервер. Для этого нужно войти в систему с привилегиями пользователя root и дать команду

# /usr/local/apache/sbin/apachectl start

(начиная с версии 27.4 - # /usr/local/apache/bin/apachectl start)

Если в конфигурационных файлах есть серьезные ошибки, сервер не запустится, а на экран будет выведено соответствующее сообщение. В любом случае после запуска сервера имеет смысл просмотреть файлы error\_log и access\_log, которые находятся в каталоге logs. Для проверки работоспособности сервера достаточно создать в его корневом каталоге файл index.html и обратиться из браузера по адресу сервера. Правильную установку режимов перекодирования следует проверять с помощью браузеров для различных операционных систем. Не забудьте добавить Apache в список программ, запускаемых при старте системы. Успехов вам в пополнении русского Web-пространства!

**Об авторе**

Артем Подстрешный - программист, работает в компании "Радио-МГУ". В "Мире ПК" опубликована его статья "Имена Internet". E-mail: art@radio-msu.net; http://www.radio-msu.net/

**Ссылки**

http://www.apache.org/ - официальный сервер разработчиков Apache

http://apache.lexa.ru/ - сервер группы разработчиков русского модуля Apache

ЛИСТИНГ 1 Фрагмент простого файла access.conf

## access.conf - Apache HTTP server configuration file

##

# access.conf: Global access configuration

# Online docs at http://www.apache.org/

<Directory />

Options FollowSymLinks

AllowOverride None

</Directory>

<Directory /www>

Options All

AllowOverride All

order allow,deny

allow from all

</Directory>

# You may place any other directories or locations you wish

to have access information for after this one.

ЛИСТИНГ 2 Описание виртуальных серверов с различными IP-адресами

...

ServerName www.radio-msu.net

<VirtualHost 193.124.134.2>

DocumentRoot /www/radio-msu.net

ServerName www.radio-msu.net

ErrorLog /var/log/error\_log.radio-msu.net

CustomLog /var/log/access\_log.radio-msu.net combined

...

</VirtualHost>

<VirtualHost 193.124.134.3>

DocumentRoot /www/rmt.ru

ServerName www.rmt.ru

ErrorLog /var/log/error\_log.radio-msu.net

CustomLog /var/log/access\_log.radio-msu.net combined

...

</VirtualHost>

ЛИСТИНГ 3 Описание виртуальных серверов, различающихся только доменным именем

...

ServerName www.radio-msu.net

NameVirtualHost 193.124.134.2

<VirtualHost 193.124.134.2>

DocumentRoot /www/radio-msu.net

ServerName www.radio-msu.net

ErrorLog /var/log/error\_log.radio-msu.net

CustomLog /var/log/access\_log.radio-msu.net combined

...

</VirtualHost>

<VirtualHost 193.124.134.2>

DocumentRoot /www/people.radio-msu.net

ServerName people.radio-msu.net

ServerAlias \*.people.radio-msu.net

ErrorLog /var/log/error\_log.people.radio-msu.net

CustomLog /var/log/access\_log.people.radio-msu.net combined

...

</VirtualHost>

**Список литературы**

Артем Подстрешный. Работа с Web-сервером Russian Apache.