DNS and BIND

Fourth Edition

Paul Albitz and Cricket Liu

O'REILLY®

DNS и BIND

Четвертое издание

Пол Альбитц и Крикет Ли

Санкт-Петербург
2002

Пол Альбитц, Крикет Ли

DNS и BIND, 4-е издание

Перевод М. Зислиса

Главный редактор	*А. Галунов*
Зав. редакцией	*Н. Макарова*
Научный редактор	*А. Маврин*
Редактор	*А. Лосев*
Корректура	*О. Маршкова*
Верстка	*Н. Гриценко*

Альбитц П., Ли К.
DNS и BIND. – Пер. с англ. – СПб: Символ-Плюс, 2002. – 696 с., ил.
ISBN 5-93286-035-9

Книга «DNS и BIND» стала библией для системных администраторов. Она уникальна по полноте изложения материала, что в сочетании с прекрасным авторским стилем делает ее незаменимой и актуальной для каждого, кто хочет наладить эффективную работу DNS. Четвертое издание включает информацию о версии 9 пакета BIND, в которой реализованы новые и очень важные механизмы, а также о версии 8, входящей в состав большинства действующих коммерческих разработок. Пакеты BIND 8 и 9 позволяют значительно повысить безопасность служб DNS.

Рассмотрены следующие темы: функциональность и принципы работы DNS; структура пространства доменных имен; установка и настройка серверов имен; применение MX-записей для маршрутизации почты; настройка узлов на работу с DNS; разделение доменов на поддомены; обеспечение безопасности DNS-сервера; новые возможности BIND 9; расширения системы безопасности DNS (DNSSEC) и подписи транзакций (TSIG); распределение нагрузки между DNS-серверами; динамические обновления, асинхронные уведомления об изменениях зон, пошаговая передача зон; устранение неполадок (применение nslookup и dig, чтение отладочного вывода); DNS-программирование с применением библиотеки DNS-клиента и модуля Perl Net::DNS.

ISBN 5-93286-035-9
ISBN 0-596-00158-4 (англ)

Издательство «Символ-Плюс». 193148, Санкт-Петербург, ул. Пинегина, 4,
тел. (812) 324-5353, edit@symbol.ru. Лицензия ЛП N 000054 от 25.12.98.
Налоговая льгота – общероссийский классификатор продукции
ОК 005-93, том 2; 953000 – книги и брошюры.
Подписано в печать 26.02.2002. Формат 70x100 1/16 . Печать офсетная.
Объем 43,5 печ. л. Тираж 3000 экз. Заказ N
Отпечатано с диапозитивов в Академической типографии «Наука» РАН
199034, Санкт-Петербург, 9 линия, 12.

Оглавление

Предисловие

Возможно, вам не так уж много известно о системе доменных имен (Domain Name System), но работая в Интернете, вы неизбежно ее используете. Всякий раз, отправляя сообщения электронной почты или исследуя просторы World Wide Web, вы полагаетесь на DNS – систему доменных имен.

Дело в том, что люди предпочитают запоминать *имена* компьютеров, а компьютерам больше нравится обращаться друг к другу по числовым адресам. В Интернете этот адрес имеет разрядность 32, то есть может быть числом от нуля до четырех с хвостиком миллиардов.[1] Компьютеры с легкостью запоминают такие вещи, потому что обладают большими объемами памяти, идеально подходящей для хранения чисел, но для людей эта задача не в пример сложнее. Попробуйте случайным образом выбрать из телефонной книги десять номеров и запомнить их. Непросто? Теперь вернитесь к началу телефонной книги и сопоставьте каждому номеру случайный код района. Примерно настолько же сложно будет запомнить 10 произвольных интернет-адресов.

Отчасти именно по этой причине необходима система доменных имен. DNS занимается двунаправленным отображением имен хостов, подходящих для запоминания людьми, и интернет-адресов, с которыми работают компьютеры. По сути дела, DNS в сети Интернет является не только средством работы с адресами, но и стандартным механизмом для предоставления и получения разнообразной информации об узлах сети. DNS нужен практически для каждой программы, обеспечивающей сетевое взаимодействие, в том числе программам для работы с электронной почтой, терминальным клиентам (например, telnet), средствам передачи файлов, таким как FTP, и, разумеется, веб-броузерам, таким как Netscape Navigator и Microsoft Internet Explorer.

Другой важной особенностью DNS является способность системы распространять информацию о хосте по всей сети Интернет. Хранение доступной информации о хосте на единственном компьютере полезно лишь для тех, кто пользуется этим компьютером. Система доменных имен обеспечивает получение информации из любой точки сети.

Более того, DNS позволяет распределять управление информацией о хостах между многочисленными серверами и организациями. Нет необ-

---

[1] А в системе IP-адресации версии 6 адреса имеют колоссальную длину – 128 бит, что позволяет охватить десятичные числа от нуля до 39-значных.

ходимости передавать данные на какой-то центральный сервер или регулярно синхронизировать свою базу данных с «основной». Достаточно убедиться, что ваш раздел, называемый *зоной*, соответствует действительности на ваших *DNS-серверах*. А они, в свою очередь, сделают информацию о зоне доступной всем остальным DNS-серверам сети.

Поскольку база данных DNS является распределенной, в системе должна быть предусмотрена возможность поиска нужной информации путем опроса множества возможных источников ее получения. Система доменных имен наделяет DNS-серверы способностью находить нужные источники информации и получать сведения по любой зоне.

Разумеется, система DNS не лишена недостатков. К примеру, в целях избыточности базы данных, система позволяет хранение зональной информации на более чем одном сервере. При этом возникает опасность десинхронизации копий зональной информации.

Но *самая большая* проблема, связанная с DNS, несмотря на широкое распространение в сети Интернет, – это реальное отсутствие хорошей документации по работе с системой. Большинство администраторов сети Интернет вынуждены обходиться лишь той документацией, которую считают достаточной поставщики используемых программ, а также тем, что им удается выудить из соответствующих списков интернет-рассылок и конференций Usenet.

Такой дефицит документации означает, что понимание предельно важной интернет-службы, одной из монументальных основ сегодняшней сети Интернет, либо передается от администратора к администратору как ревностно хранимая семейная тайна, либо постоянно изучается повторно отдельными программистами и разработчиками. Новые администраторы зон повторяют ошибки, уже бесчисленное число раз сделанные другими.

Цель этой книги – изменить сложившуюся ситуацию. Мы осознаем, что не у каждого читателя есть время и желание становиться специалистом по DNS. У большинства из вас есть достаточно других занятий, помимо управления зонами и DNS-серверами: системное администрирование, разработка сетевых инфраструктур, или разработка программного обеспечения. Заниматься исключительно DNS может только сотрудник безумно большой организации. Мы постарались представить информацию, достаточную для решения основных рабочих задач, будь то управление небольшой зоной или целой международной системой, работа с единственным сервером имен или наблюдение за сотней серверов. Извлеките из книги нужный вам минимум, и возвращайтесь к ней по мере необходимости.

DNS – это сложная тема, настолько сложная, что взяться за нее пришлось не одному, а двум авторам; и мы постарались представить систему настолько прозрачно и доступно, насколько это возможно. В первых двух главах содержится теоретический обзор и достаточный для применения объем практической информации, а в последующих гла-

вах использование системы доменных имен рассмотрено более подробно. С самого начала мы предлагаем читателям нечто вроде дорожной карты, чтобы каждый мог выбрать свой собственный путь изучения книги, соответствующий рабочим задачам или интересам.

Когда речь пойдет о программах, обеспечивающих работу DNS, мы практически целиком сконцентрируемся на инструменте под названием BIND, Berkeley Internet Name Domain, который является наиболее популярной (и наиболее нами изученной) реализацией спецификаций DNS. Мы старались представить в этой книге выжимку из нашего опыта управления и поддержки зон с помощью BIND. (Так получилось, что некоторое время одна из наших зон являлась самой большой зоной сети Интернет; правда, это было очень давно). Где это было возможно, мы включали реальные программы, используемые нами в администрировании; многие из них переписаны на языке Perl с целью достижения большей скорости работы и повышения эффективности.

Надеемся, эта книга поможет вам познакомиться с системой DNS и инструментом BIND, если вы еще новичок, лучше понять их работу, если вы уже знакомы, и приобрести ценное понимание и опыт, даже если вы уже знаете DNS и BIND, как свои пять пальцев.

Версии

Четвертое издание этой книги затрагивает новые версии BIND – 9.1.0 и 8.2.3, а также более старую версию 4.9. Несмотря на то, что на момент написания этой книги версии 9.1.0 и 8.2.3 являются наиболее свежими, они пока не получили широкого распространения в составе Unix-систем, отчасти потому, что обе версии были выпущены недавно, а многие поставщики настороженно относятся к использованию новых программ. Мы время от времени упоминаем и другие версии BIND, в частности 4.8.3, поскольку многие поставщики продолжают распространять программы, содержащие код, основанный на более старых версиях, в составе своих Unix-разработок. Если определенная возможность доступна только в версии 4.9, 8.2.3 или 9.1.0, либо если существуют различия в поведении версий, мы постараемся четко определить, что именно работает и для какой версии BIND.

В наших примерах мы очень часто прибегаем к служебной программе DNS – *nslookup*. Мы пользуемся *nslookup* из комплекта поставки BIND версии 8.2.3. Более старые версии *nslookup* обеспечивают большую часть функциональности (но не всю) *nslookup* версии 8.2.3.[1] В большинстве примеров мы использовали команды, доступные почти во всех версиях *nslookup*; случаи, когда это было невозможно, отмечены отдельно.

[1] Это верно также и для nslookup из комплекта поставки BIND версии 9.1.0. См. более подробно в главе 12 «nslookup и dig».

Что нового в четвертом издании?

Текст книги был обновлен, чтобы соответствовать наиболее поздним версиям BIND; мы также добавили в большом объеме новый материал:

- Более подробное рассмотрение динамических обновлений и механизма NOTIFY, включая и подписываемые динамические обновления (signed dynamic updates), а также описание нового для BIND 9 механизма *update-policy* – в главе 10.
- Поэтапная передача зоны – также в главе 10.
- Зоны ретрансляции, поддерживающие передачу по условию (conditional forwarding), – в главе 10.
- Прямое и обратное отображение адресов в контексте технологии IPv6 с использованием записей новых типов A6 и DNAME, а также бит-строковых меток – в конце главы 10.
- Новый механизм подтверждения подлинности транзакций – транзакционные подписи (transaction signatures, известные также как TSIG) – описан в главе 11.
- Более подробное рассмотрение вопросов обеспечения безопасности DNS-серверов – в главе 11.
- Более подробное рассмотрение работы с брандмауэрами в сети Интернет – в главе 11.
- Описаны расширения DNS, связанные с безопасностью (DNS Security Extensions или DNSSEC), представляющие собой новый механизм цифровой подписи зональных данных, – все в той же 11-ой главе.
- Раздел, посвященный совместной работе клиентов, серверов и контроллеров доменов Windows 2000 и BIND, – в главе 16.

Структура

Порядок следования глав настоящей книги приблизительно соответствует возможному развитию зоны и росту знаний ее администратора. В главах 1 и 2 обсуждается теория системы доменных имен. В главах с 3 по 6 рассматриваются вопросы, связанные с принятием решений по созданию собственных зон, а также действия администратора в случае необходимости создать зону. Следующая часть книги, главы с 7 по 11, посвящена сопровождению зон, настройке хостов для использования DNS-серверов, планированию развития зон, созданию доменов различных уровней и безопасности серверов. Наконец, главы с 12 по 16 посвящены разрешению сложностей, возникающих при работе с различными инструментами, общим проблемам и забытому искусству программирования с применением библиотек DNS-клиента.

Перечислим темы по главам:

Глава 1 «Основы» описывает исторический фон создания системы, и посвящена проблемам, приведшим к созданию DNS, а также собственно обзору теории системы доменных имен.

Глава 2 «Как работает DNS» посвящена более подробному рассмотрению теоретических основ DNS, в частности – организации пространства имен в системе DNS, доменов, зон и DNS-серверов. Там же рассматриваются такие важные понятия, как разрешение адресов и кэширование.

Глава 3 «С чего начать?» посвящена вопросам получения пакета BIND в случае его отсутствия, применениям пакета, когда он уже у вас в руках, определению и выбору доменного имени, а также установления связи с организацией, которая обладает полномочиями делегировать выбранную зону.

Глава 4 «Установка BIND» – это подробное рассмотрение того, как установить два первых DNS-сервера на основе BIND, включая создание базы данных серверов, запуск и диагностику их работы.

Глава 5 «DNS и электронная почта» рассказывает о записи DNS типа MX, которая позволяет администраторам задавать альтернативные узлы, которым передается на обработку почта для определенных адресов. В этой главе описаны стратегии маршрутизации почты для различных типов сетей и узлов, включая сети с интернет-брандмауэрами и узлы, не имеющие прямого подключения к сети Интернет.

Глава 6 «Конфигурирование узлов» рассказывает о том, как настраивать клиентскую часть (*resolver*) BIND, а также об особенностях реализаций клиента – как в составе распространенных Unix-систем, так и применяемых на платформах Windows 95/NT/2000.

Глава 7 «Работа с BIND» посвящена регулярным действиям администратора, выполнение которых необходимо для поддержания устойчивой работы зон, находящихся под его началом, в частности – проверке состояния DNS-сервера и вопросов, касающихся авторитативных серверов зоны.

Глава 8 «Развитие домена» рассказывает о планировании роста и эволюции зон, включая вопросы о том, как вырасти большим, а также о планировании переездов и перебоев в работе.

Глава 9 «Материнство» – о радостях, связанных с обретением потомства. Мы расскажем, когда имеет смысл заводить детей (создавать поддомены), как их называть, *как* их заводить (!) и как присматривать за ними.

Глава 10 «Дополнительные возможности» рассказывает о параметрах настройки сервера имен, которые используются не очень часто, но могут помочь в тонкой настройке DNS-сервера и в упрощении процесса администрирования.

Глава 11 «Безопасность» посвящена обеспечению безопасности и тем настройкам DNS-сервера, которые относятся к работе с интернет-

брандмауэрами, а также двум новым технологиям DNS, связанным с безопасностью: DNS Security Extensions и подписям транзакций (Transaction Signatures).

Глава 12 «nslookup и dig» подробно рассказывает о самых популярных инструментах DNS-отладки, и содержит описания способов извлечения неявной информации из удаленных DNS-серверов.

Глава 13 «Чтение отладочного вывода BIND» – это Розеттский камень[1] отладочной информации BIND. Глава поможет разобраться в таинственной отладочной информации, создаваемой пакетом BIND, а это, в свою очередь, поможет лучше понять, как работает DNS-сервер

Глава 14 «Разрешение проблем DNS и BIND» содержит описания и способы разрешения многих распространенных проблем, связанных с использованием DNS и BIND, а также рассказывает о более редких случаях, связанных с ошибками, диагностика которых может вызывать затруднения.

Глава 15 «Программирование с использованием библиотечных функций» рассказывает о том, как использовать функции библиотеки клиента BIND для опроса DNS-серверов и получения информации в программе на языке C или Perl. Приводится исходный текст полезной (как мы надеемся) программы, которая проверяет работоспособность DNS-серверов и их авторитативность.

Глава 16 «Обо всем понемногу» посвящена незатронутым темам. Она содержит описание использования масок (wildcards) в DNS, принципов работы с хостами и сетями, не имеющими постоянного подключения к сети Интернет, кодировки сетевых имен, экспериментальных типов записей и работы с DNS в Windows 2000.

Приложение A «Формат сообщений DNS и RR-записи (resource records)» содержит предельно подробный справочник по форматам, используемым в запросах и ответах DNS, а также полный перечень определенных в настоящее время типов RR-записей.

Приложение B «Таблица совместимости BIND» – это перечисление наиболее важных особенностей самых распространенных версий BIND.

Приложение C «Сборка и установка BIND на Linux-системах» содержит пошаговые инструкции по сборке BIND версии 8.2.3 в Linux.

---

[1] Розеттский камень – черная базальтовая плита с трехъязычной надписью на египетском иероглифическом, египетском демотическом (разговорном) и древнегреческом языках, обнаруженная в 1799 г. офицером наполеоновских войск Бушаром при сооружении форта Сен-Жюльен на берегу Розеттского рукава Нила. Расшифровка иероглифического текста в 1822 г. стала началом изучения египетской иероглифической письменности. – *Примеч. ред.*

Приложение D «Домены высшего уровня» – это перечисление существующих в настоящее время доменов высшего уровня сети Интернет.

Приложение E «Настройка DNS-сервера и клиента BIND» содержит справочник по синтаксису и семантике каждого из существующих параметров настройки серверов и библиотек клиента.

Для кого эта книга

Прежде всего эта книга предназначена для системных и сетевых администраторов, которым приходится управлять зонами и одним или несколькими DNS-серверами, но она содержит материал, который будет интересен проектировщикам сетей, почтовым администраторам и многим другим людям. Не все главы одинаково интересны для столь разношерстной аудитории, и, конечно же, читателю нет смысла копаться во всех шестнадцати главах, чтобы найти интересующий его материал. Мы надеемся, что следующая карта поможет выстроить правильный путь по главам книги.

Системным администраторам, впервые столкнувшимся с вопросами сопровождения зон, следует прочесть главы 1 и 2, чтобы получить теоретическую подготовку по DNS, главу 3 – в целях получения информации о первых шагах и выборе подходящего доменного имени, главы 4 и 5 – чтобы узнать, как происходит настройка зоны «с нуля». Глава 6 объясняет, как настроить хосты для работы с новыми DNS-серверами. Несколько позже следует обратиться к главе 7, в которой объясняется, как «подкачать» объем, добавляя серверы и данные в зону. Главы 12, 13 и 14 содержат описание инструментов и методов, помогающих в устранении проблем.

Опытным администраторам будет полезно прочитать главу 6, чтобы узнать, как настраивать DNS-клиенты на различных хостах, и главу 7, чтобы получить информацию о том, как грамотно сопровождать зоны. В главе 8 содержатся инструкции, связанные с планированием роста и развития зоны, которые должны быть особенно полезны людям, занятым в администрировании больших зон. Глава 9 рассказывает о том, как можно стать родителем – то есть, о создании поддоменов, и является учебником *этикета*, обязательным к прочтению теми, кто планирует совершить этот трудный шаг. В главе 10 рассмотрены многие новые возможности BIND версий 8.2.3 и 9.1.0. Глава 11 посвящена обеспечению безопасности DNS-серверов и для опытных администраторов может представлять особенный интерес. Главы с 12 по 14 содержат описание действий на случай возникновения проблем и сопутствующих инструментов, эти главы могут оказаться занимательным чтением даже для очень опытных администраторов.

Системным администраторам сетей, не имеющих постоянного подключения к сети Интернет, рекомендуется прочесть главу 5, чтобы изучить процесс настройки маршрутизации почты в таких сетях, и

главу 11, которая содержит описание создания независимой инфраструктуры DNS.

Программистам, в целях освоения теории DNS, предлагается прочесть главы 1 и 2, а затем главу 15, в которой содержится подробное рассмотрение программирования при помощи библиотечных функций BIND.

Сетевым администраторам, которые напрямую не вовлечены в процесс сопровождения зон, рекомендуется прочесть главы 1 и 2, в целях освоения теории DNS, главу 12, чтобы научиться использовать *nslookup* и *dig*, а затем главу 14, чтобы узнать о способах разрешения возникающих сложностей.

Почтовым администраторам следует прочесть главы 1 и 2, в целях освоения теории DNS, главу 5, чтобы узнать, как сосуществуют DNS и электронная почта, и главу 12, в которой описаны инструменты *nslookup* и *dig*, – эта глава научит извлекать информацию о маршрутизации почты из пространства доменных имен.

Заинтересованные пользователи могут прочесть главы 1 и 2, в целях освоения теории DNS, а затем – любые главы, по желанию!

Мы предполагаем, что читатель знаком с основами администрирования Unix-систем, сетевым взаимодействием TCP/IP, а также программированием на уровне простых сценариев командного интерпретатора или языка Perl. При этом никаких других специальных знаний не требуется. При появлении новых терминов и понятий они насколько возможно подробно объясняются в тексте книги. По возможности мы использовали аналогии с системами Unix (и реальным миром), чтобы облегчить читателю восприятие новых для него концепций.

Примеры программ

Исходные тексты программ-примеров, приводимых в книге, доступны для загрузки по протоколу FTP по следующим адресам:

ftp://ftp.uu.net/published/oreilly/nutshell/dnsbind/dns.tar.Z
ftp://ftp.oreilly.com/published/oreilly/nutshell/dnsbind/

В обоих случаях извлечь файлы из архива можно командой:

```
% zcat dns.tar.Z | tar xf -
```

На System V – системах необходимо использовать следующую *tar*-команду:

```
% zcat dns.tar.Z | tar xof -
```

Если команда *zcat* недоступна в системе, следует использовать отдельные команды *uncompress* и *tar*.

Если не удается получить тексты примеров напрямую по сети Интернет, но существует возможность посылать и получать сообщения электронной почты, можно воспользоваться службой *ftpmail*. Чтобы получить справку по использованию службы *ftpmail*, необходимо отправить сообщение на адрес *ftpmail@online.oreilly.com*. Следует оставить пустым поле темы сообщения; тело письма должно содержать единственное слово – «help».

Как связаться с издательством O'Reilly

Комментарии и вопросы, связанные с этой книгой, можно направлять непосредственно издателю:

O'Reilly & Associates, Inc.
101 Morris Street
Sebastopol, CA 95472
(800) 998-9938 (в США или Канаде)
(707) 829-0515 (международный/местный)
(707) 829-0104 (факс)

Издательством O'Reilly создана веб-страница, посвященная этой книге, на которой доступна информация о найденных ошибках и будут появляться разнообразные дополнительные сведения. Страница доступна по адресу:

http://www.oreilly.com/catalog/dns4

Если у вас есть технический вопрос или комментарий, связанный с этой книгой, задайте его, отправив сообщение по адресу:

bookquestions@oreilly.com

На веб-сайте издательства O'Reilly доступна дополнительная информация о книгах, конференциях, программном обеспечении, источниках информации и сети O'Reilly (O'Reilly Network):

http://www.oreilly.com

Типографские соглашения

Использованы следующие соглашения по шрифту и формату для команд, инструментов и системных вызовов Unix:

- Выдержки из сценариев или конфигурационных файлов оформлены моноширинным шрифтом:

```
if test -x /usr/sbin/named -a -f /etc/named.conf
then
    /usr/sbin/named
fi
```

- Примеры диалоговых сеансов, отображающие ввод в командной строке и соответствующую реакцию системы, оформлены непропорциональным шрифтом, причем ввод пользователя отмечен жирным выделением:

```
% cat /var/run/named.pid
78
```

- Если команда должна вводиться суперпользователем (администратором системы, или пользователем root), она предваряется символом диеза (#):

```
# /usr/sbin/named
```

- Заменяемые элементы кода оформлены моноширинным курсивом.
- Имена доменов, файлов, функций, команд, названия страниц руководства Unix, фрагменты кода оформлены курсивом, если они расположены внутри параграфа.

Цитаты

Цитаты из Льюиса Кэррола в каждой из глав приводятся по версии 2.9 издания Millenium Fulcrum электронного текста «Алисы в Стране чудес» из библиотеки Проекта Гутенберга (Project Gutenberg) и по изданию 1.7 текста «Алиса в Зазеркалье». Цитаты в главах 1, 2, 5, 5, 8 и 14 из «Алисы в стране чудес», а цитаты в главах 3, 4, 7, 9, 10, 11, 12, 13, 15 и 16 – из «Алисы в зазеркалье».[1]

Благодарности

Авторы выражают благодарность Кену Стоуну (Ken Stone), Джерри МакКоллому (Jerry McCollom), Питеру Джеффу (Peter Jeffe), Хэлу Стерну (Hal Stern), Кристоферу Дарему (Christopher Durham), Биллу Уизнеру (Bill Wisner), Дэйву Керри (Dave Curry), Джеффу Окамото (Jeff Okamoto), Брэду Ноулзу (Brad Knowles), Роберту Эльцу (K. Robert Elz), а также Полу Викси (Paul Vixie) за их бесценный вклад в написание этой книги. Мы также хотели бы поблагодарить наших рецензентов – Эрика Пирса (Eric Pearce), Джека Репенинга (Jack Repenning), Эндрю Черенсона (Andrew Cherenson), Дэна Тринкла (Dan Trinkle), Билла Лефевра (Bill LeFebvre) и Джона Секреста (John Sechrest) за их критику и предложения. Без помощи этих людей эта книга была бы совсем не такой (а была бы она гораздо короче!).

За второе издание этой книги авторы выражают благодарность безупречной команде рецензентов: Дэйву Бэрру (Dave Barr), Найджелу

[1] В русском издании для цитат используется перевод Нины Демуровой (М.:, ПРЕССА, 1992). – *Примеч. ред.*

Кэмпбеллу (Nigel Campbell), Биллу Лефевру, Майку Миллигану (Mike Milligan) и Дэну Тринклу.

За третье издание книги авторы отдают честь команде мечты технических рецензентов: Бобу Хэлли (Bob Halley), Барри Марголину (Barry Margolin) и Полу Викси.

Долг благодарности за четвертое издание причитается Кевину Данлэпу (Kevin Dunlap), Эдварду Льюису (Edward Lewis) и Брайану Веллингтону (Brian Wellington), первоклассной команде рецензентов.

Крикет хотел бы отдельно поблагодарить своего бывшего руководителя, Рика Норденстена (Rick Nordensten), образцового современного высокопроизводительного менеджера, под присмотром которого была написана первая версия этой книги; своих соседей, которые терпели его эпизодическую раздражительность в течение многих месяцев, и, конечно же, свою жену Пэйдж за постоянную поддержку и за то, что она мирилась с непрекращающимся, даже во время ее сна, стуком клавиш. Что касается второго издания, Крикет хотел бы добавить слова благодарности в адрес своих бывших руководителей Регины Кершнер (Regina Kershner) и Пола Клоуда (Paul Klouda) за их поддержку работы Крикета с сетью Интернет. За помощь в работе над третьим изданием Крикет считает своим долгом поблагодарить своего партнера, Мэтта Ларсона (Matt Larson), который участвовал в разработке Acme Razor; за четвертое он благодарит своих преданных пушистиков Дакоту и Энни – за их поцелуи и участие, а также замечательного Уолтера Б. (Walter B), который время от времени заглядывал в кабинет и проверял, как у Папы дела. Пол благодарит свою жену Катерину за ее терпение, за многочисленные разборы полетов и за доказательство того, что она в свободное время может гораздо быстрее сшить стеганое одеяло, чем ее муж напишет свою половину книги.

Мы хотим сказать спасибо ребятам из O'Reilly & Associates, за их тяжелый труд и терпение. В особенности этой благодарности заслуживают наши редакторы – Майк Лукидес (Mike Loukides) (издания с первого по третье) и Дебра Кэмерон (Debra Cameron) (четвертое издание), а также огромное количество других людей, которые работали над различными изданиями этой книги: Нэнси Котари (Nancy Kotary), Элли Фонтэйн Мэйден (Ellie Fountain Maden), Роберт Романо (Robert Romano), Стивен Абрамс (Steven Abrams), Кишмет МакДоноу-Чен (Kismet McDonough-Chan), Сет Мэйслин (Seth Maislin), Элли Катлер (Ellie Cutler), Майк Сьерра (Mike Sierra), Ленни Мельнер (Lenny Muellner), Крис Райли (Chris Reilley), Эмили Куилл (Emily Quill), Анна-Мария Вадува (Anne-Marie Vaduva) и Брэнда Миллер (Brenda Miller). Также спасибо Джерри Пику (Jerry Peek) за самую разнообразную поддержку, и Тиму О'Рейли за то, что он вдохновил нас на написание этой книги.

И спасибо Эди за сверчка[1] на обложке!

[1] Cricket (фамилия одного из авторов) переводится с англ. как «сверчок». – *Примеч. перев.*

1

Основы

- *(Очень) краткая история сети Интернет*
- *Интернет и интернет-сети*
- *Система доменных имен в двух словах*
- *История пакета BIND*
- *Надо ли мне использовать DNS?*

Кролик надел очки.
– С чего начинать, Ваше Величество? – спросил он.
– Начни с начала, – важно ответил Король, – продолжай, пока не дойдешь до конца. Как дойдешь – кончай!

Чтобы понять DNS, необходимо обратиться к истории сети ARPAnet. Система DNS была создана с целью решения конкретных проблем этой сети, а сеть Интернет, выросшая из ARPAnet, сейчас является главным потребителем этих решений.

Опытные пользователи сети Интернет, вероятно, могут пропустить эту главу. Все прочие, мы надеемся, найдут здесь достаточно информации для понимания причин, которые привели к появлению DNS.

(Очень) краткая история сети Интернет

В конце шестидесятых Управление передовых исследований Министерства обороны США (Department of Defense's Advanced Research Agency, или ARPA) – позднее DARPA – открыло финансирование ARPAnet, экспериментальной глобальной компьютерной сети, объединившей важные исследовательские организации страны. Первоначальной целью создания этой сети было разделение дорогостоящих или дефицитных компьютерных ресурсов между государственными подрядчиками. Но с самого начала пользователи ARPAnet использовали сеть и для совместной работы: они обменивались файлами и программами, сообщениями электронной почты (получившей теперь повсеместное распространение), объединяли усилия по разработке и исследованиям, используя разделяемые ресурсы компьютеров сети.

Набор протоколов TCP/IP (Transmission Control Protocol/Internet Protocol) был разработан в начале восьмидесятых годов и быстро получил

статус стандарта для сетевого обмена информацией между узлами ARPAnet. Включение этого набора протоколов в популярную операционную систему BSD Unix, разработанную в Калифорнийском университете Беркли, сыграло определяющую роль в процессе демократизации и объединения сетей. Система BSD Unix университетам была доступна практически бесплатно. Так работа с сетями и доступ к ARPAnet стали внезапно доступными и очень дешевыми для большого числа организаций, которые ранее никак не были связаны с ARPAnet. Машины, подключавшиеся к ARPAnet, входили также в состав локальных сетей, и в итоге это привело к объединению многочисленных разрозненных локальных сетей посредством ARPAnet.

Сеть разрослась с очень небольшого числа узлов до десятков тысяч. Первоначальная сеть ARPAnet стала основой объединения локальных и региональных сетей, работающих по протоколам TCP/IP. Это объединение носит название *Интернет*.

Однако в 1988 году организация DARPA пришла к заключению, что эксперимент окончен. Министерство обороны приступило к демонтажу сети ARPAnet. В этот момент несущим остовом для сети Интернет стала другая сеть, которая финансировалась национальным научным фондом (National Science Foundation) и носила название NSFNET.

Уже не так давно, весной 1995 года, сеть Интернет в очередной раз сменила главную магистраль; финансируемая обществом NSFNET уступила место целому ряду коммерческих магистралей, управляемых операторами дальней связи, такими как MCI и Sprint, а также такими опытными коммерческими сетевыми организациями как PSINet и UUNET.

Сегодня сеть Интернет объединяет миллионы узлов по всему миру. Большая часть не-PC машин подключена к сети Интернет. Некоторые из новых коммерческих информационных магистралей имеют пропускную способность, измеряемую гигабитами в секунду, что в десятки тысяч раз превышает пропускную способность когда-то существовавшей ARPAnet. Ежедневно десятки миллионов людей используют сеть для общения и совместной работы.

Интернет и интернет-сети

Следует сказать пару слов об Интернете и интернет-сетях. В тексте различие между названиями кажется незначительным: одно название всегда пишется с прописной буквы, второе всегда нет. Тем не менее, разница в значении – *существенна*. Интернет с заглавной буквы «И» – обозначение сети, которая началась с ARPAnet и существует сегодня, грубо говоря, как объединение всех TCP/IP-сетей, прямо или косвенно связанных с коммерческими информационными магистралями США. При внимательном рассмотрении – это целый ряд различ-

ных сетей – коммерческих опорных сетей TCP/IP, корпоративных и правительственных сетей США, а также TCP/IP-сетей других стран. Сети объединены маршрутизаторами и высокоскоростными цифровыми каналами.

Начинающийся со строчной буквы интернет – это просто любая сеть, объединяющая несколько сетей масштабом поменьше, причем с использованием все тех же протоколов межсетевого взаимодействия. Сеть интернет не всегда связана с сетью Интернет и не обязательно основана на сетевых протоколах TCP/IP. Существуют изолированные корпоративные интернет-сети, а также сети на основе протоколов Xerox XNS или DECnet.

Относительно новый термин «intranet», по сути дела, – это всего лишь рекламное название для интернет-сетей, которое используется, чтобы подчеркнуть применение технологий, обкатанных в Интернете, в рамках внутренних корпоративных сетей. С другой стороны, extranet-сети – это сети интернет, объединяющие сотрудничающие компании, либо компании с их агентами по продаже, поставщиками и клиентами.

История системы доменных имен

В семидесятых годах сеть ARPAnet представляла собой тесное сообщество из нескольких сотен узлов. Всю жизненно-важную информацию по узлам, в частности, необходимую для взаимных преобразований имен и адресов узлов ARPAnet, содержал единственный файл *HOSTS.TXT*. Известная Unix-таблица узлов, */etc/hosts*, – прямо унаследовала свою структуру от файла *HOSTS.TXT* (в основном с помощью удаления ненужных на Unix-системах полей).

За файл *HOSTS.TXT* отвечал Сетевой информационный центр (NIC, Network Information Center) Стэнфордского исследовательского института (SRI, Stanford Research Insitute). В тот период времени единственным источником, распространявшим файл, являлся узел SRI-NIC.[1] Администраторы ARPAnet, как правило, просто посылали изменения электронной почтой в NIC и периодически синхронизировали свои файлы *HOSTS.TXT* с копией на узле SRI-NIC с помощью протокола FTP. Присылаемые ими изменения добавлялись в файл *HOSTS.TXT* один или два раза в неделю. Однако по мере роста сети ARPAnet эта схема стала неработоспособной. Размер файла рос пропорционально количеству узлов ARPAnet. Еще быстрее рос информационный поток, связанный с необходимостью обновления файла на хостах: появление одного нового узла приводило не только к добавлению строки в *HOSTS.TXT*,

[1] Организация SRI International в настоящее время уже не связана жестко со Стэнфордским исследовательским институтом, расположенном в Менло-Парк (в Калифорнии); она проводит исследования во многих областях, включая и компьютерные сети.

но и к потенциальной необходимости синхронизации данных каждого узла с данными SRI-NIC.

После перехода ARPAnet на протоколы TCP/IP рост сети стал взрывным. Появился гордиев узел проблем, связанных с файлом *HOSTS.TXT*:

Информационные потоки и нагрузка

Нагрузка на SRI-NIC в смысле сетевого трафика и работы процессора, связанных с раздачей файла, приближалась к предельной.

Конфликты имен

Никакие два хоста, описанные в файле *HOSTS.TXT*, не могли носить одинаковые имена. Организация NIC могла контролировать присваивание адресов способом, гарантирующим их уникальность, но не имела никакого влияния на имена узлов. Не было способа предотвратить добавление узла с уже существующим именем, при том, что такое действие нарушало работу всей схемы. Так, добавление узла с именем, идентичным имени одного из крупных почтовых концентраторов, могло привести к нарушению работы почтовых служб большей части сети ARPAnet.

Синхронизация

Синхронизация файлов в масштабах быстро растущей сети становилась все более сложной задачей. К тому моменту, когда обновленный файл *HOSTS.TXT* достигал самых далеких берегов выросшей ARPAnet, адреса отдельных узлов успевали измениться, а также появлялись новые узлы, доступ к которым был необходим пользователям.

Основная проблема заключалась в том, что схема с файлом *HOSTS.TXT* не поддавалась масштабированию. По иронии судьбы, успех ARPAnet как эксперимента вел к моральному устареванию и провалу механизма *HOSTS.TXT*.

Административные советы ARPAnet начали исследование, которое должно было привести к созданию замены *HOSTS.TXT*. Целью его было создание системы, которая решила бы проблемы, присущие сводной таблице узлов. Новая система должна позволять производить локальное управление данными, но делать эти данные доступными всем. Децентрализация администрирования решила бы проблемы с трафиком и нагрузкой, непосильными для единственного узла. Локальное управление данными упростило бы задачу обновления и синхронизации информации. В новой системе следует использовать иерархическое пространство имен для присваивания идентификаторов узлам, что позволит гарантировать уникальность каждого отдельного имени.

Ответственным за разработку архитектуры новой системы стал Пол Мокапетрис, работавший тогда в Институте информационных наук (Information Sciences Institute). В 1984 году он издал документы RFC

882 и 883, в которых описывалась система доменных имен (Domain Name System, или DNS). Эти RFC-документы были обновлены документами RFC 1034 и 1035, которые и являются в настоящее время действующей спецификацией DNS.[1] RFC 1034 и 1035 к настоящему времени дополняются многими другими подобными документами, в которых описаны потенциальные проблемы DNS с точки зрения сетевой безопасности, возможные трудности реализации, проблемы административного плана, механизмы динамического обновления DNS-серверов, обеспечение безопасности зональных данных и многое другое.

Система доменных имен в двух словах

DNS – это *распределенная* база данных. Это свойство дает возможность локально управлять отдельными сегментами общей базы, а также позволяет сделать данные каждого сегмента доступными всей сети – посредством использования механизма клиент-сервер. Надежность и адекватная производительность основаны на репликации и кэшировании.

Серверная часть клиент-серверного механизма DNS представлена программами, которые называются *DNS-серверами* (*name servers*, дословно – серверами имен). DNS-серверы владеют информацией о некоторых сегментах базы данных и делают ее доступной клиентам, которые называются *поисковыми анализаторами* (*resolvers*).[2] Как правило, DNS-клиент – это просто набор библиотечных функций, которые создают запросы и посылают их по сети серверу имен.

Структура базы данных DNS очень похожа на структуру файловой системы Unix (рис. 1.1). Вся база данных (или файловая система) представлена в виде перевернутого дерева, корень (корневой узел) которого расположен на самом верху. Каждый узел дерева имеет прикрепленную текстовую метку, которая идентифицирует его относительно родительского узла, по аналогии с «относительным путевым именем» в файловой системе (например, *bin*). Одна из меток, пустая, (она обозначается как "") закреплена за корневым узлом дерева. В тексте корневой узел обозначается точкой (.). В файловой системе Unix корень обозначается символом «слэш» (/).

Каждый узел является корнем новой ветви дерева. Каждая из ветвей (поддеревьев) является разделом базы данных – «каталогом» в ин-

---

[1] Документы RFC (Request for Comments, запрос комментариев) являются частью относительно неформальной процедуры введения новых технологий в сети Интернет. RFC-документы обычно свободно распространяются и содержат описания технического плана технологий, предназначенные, во многих случаях, для разработчиков.

[2] Далее в тексте будет использоваться как термин «поисковый анализатор», так и «клиентская часть DNS», или просто «DNS-клиент», в зависимости от контекста. – *Примеч. науч. ред.*

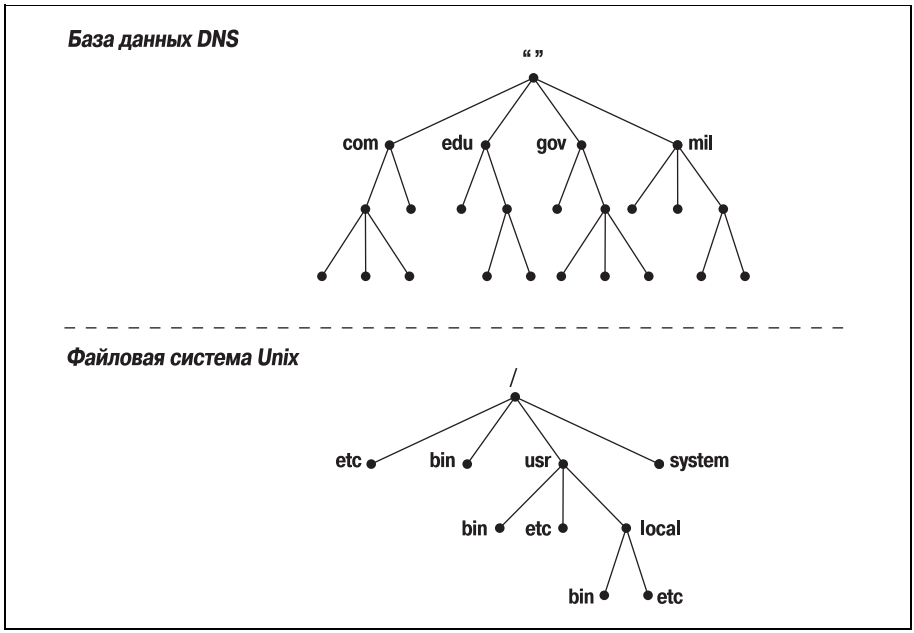

Рис. 1.1. *База данных DNS и файловая система Unix*

терпретации файловой системы Unix, или *доменом* в интерпретации системы доменных имен. Каждый домен или каталог может быть разбит на еще более мелкие подразделы, которые в DNS называются *поддоменами*, а в файловых системах – «подкаталогами». Поддомены, как и подкаталоги, изображаются как потомки соответствующих родительских доменов.

Имя домена, как и имя любого каталога, уникально. *Имя домена* определяет его расположение в базе данных, так же, как «абсолютный путь к каталогу» однозначно определяет его расположение в файловой системе. Имя домена в DNS – это последовательность меток от узла, корневого для данного домена, до корня всего дерева; метки в имени домена разделяются точками. В файловой системе Unix абсолютное путевое имя каталога – это последовательность относительных имен, начиная от корня дерева до конкретного узла (то есть чтение происходит в направлении, противоположном направлению чтения имен DNS; рис. 1.2), при этом имена разделяются символом «прямая наклонная черта» («слэш»).

В DNS каждый домен может быть разбит на поддомены, и ответственность за эти поддомены может распределяться между различными организациями. Допустим, компания Network Solutions сопровождает домен *edu* (*educational*, то есть образовательный), но делегирует ответственность за поддомен *berkeley.edu* Калифорнийскому университету Беркли (рис. 1.3). это похоже на удаленное монтирование файловой системы: определенные каталоги файловой системы могут в действи-

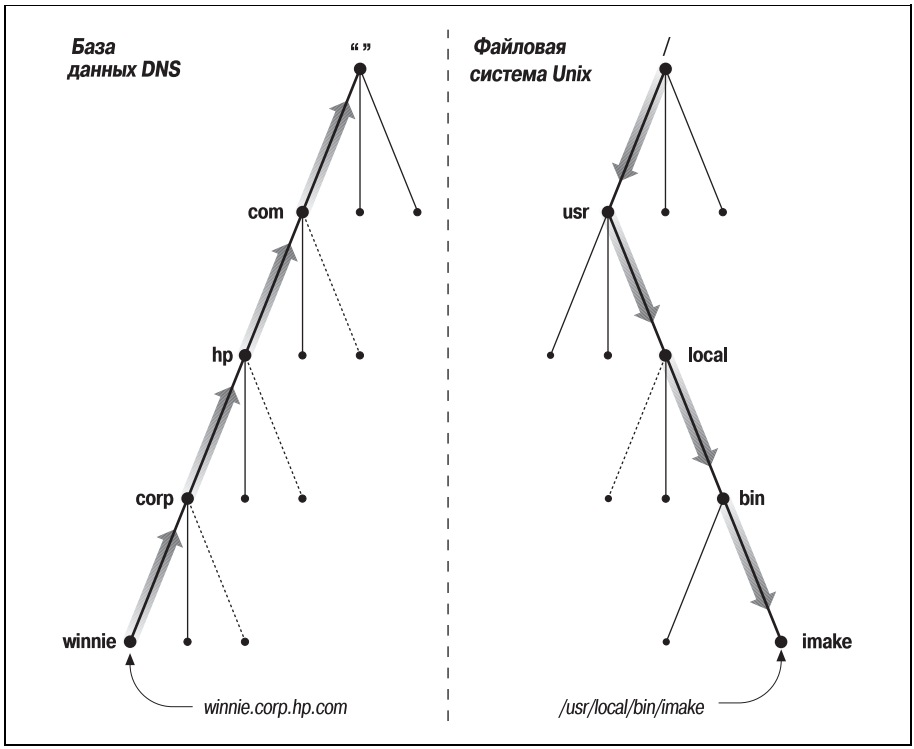

Рис. 1.2. *Чтение имен DNS и файловой системы Unix*

тельности являться файловыми системами, расположенными на других узлах и смонтированными удаленно. К примеру, администратор узла *winken* (рис. 1.3) отвечает за файловую систему, которая на локальном узле выглядит как содержимое каталога */usr/nfs/winken*.

Делегирование управления поддоменом *berkeley.edu* Калифорнийскому университету Беркли приводит к созданию новой *зоны* – независимо администрируемой части пространства имен. Зона *berkeley.edu* теперь не зависит от *edu* и содержит все доменные имена, которые заканчиваются на *berkeley.edu*. С другой стороны, зона *edu* содержит только доменные имена, оканчивающиеся на *edu*, но не входящие в делегированные зоны, такие, например, как *berkeley.edu*. *berkeley.edu* может быть поделен на поддомены с именами вроде *cs.berkeley.edu*, и некоторые из этих поддоменов могут быть выделены в самостоятельные зоны, если администраторы *berkeley.edu* делегируют ответственность за них другим организациям. Если *cs.berkeley.edu* является самостоятельной зоной, зона *berkeley.edu* не содержит доменные имена, которые заканчиваются на *cs.berkeley.edu* (рис. 1.4).

Доменные имена используются в качестве индексов базы данных DNS. Данные DNS можно считать «привязанными» к доменному имени. В файловой системе каталоги содержат файлы и подкаталоги. Анало-

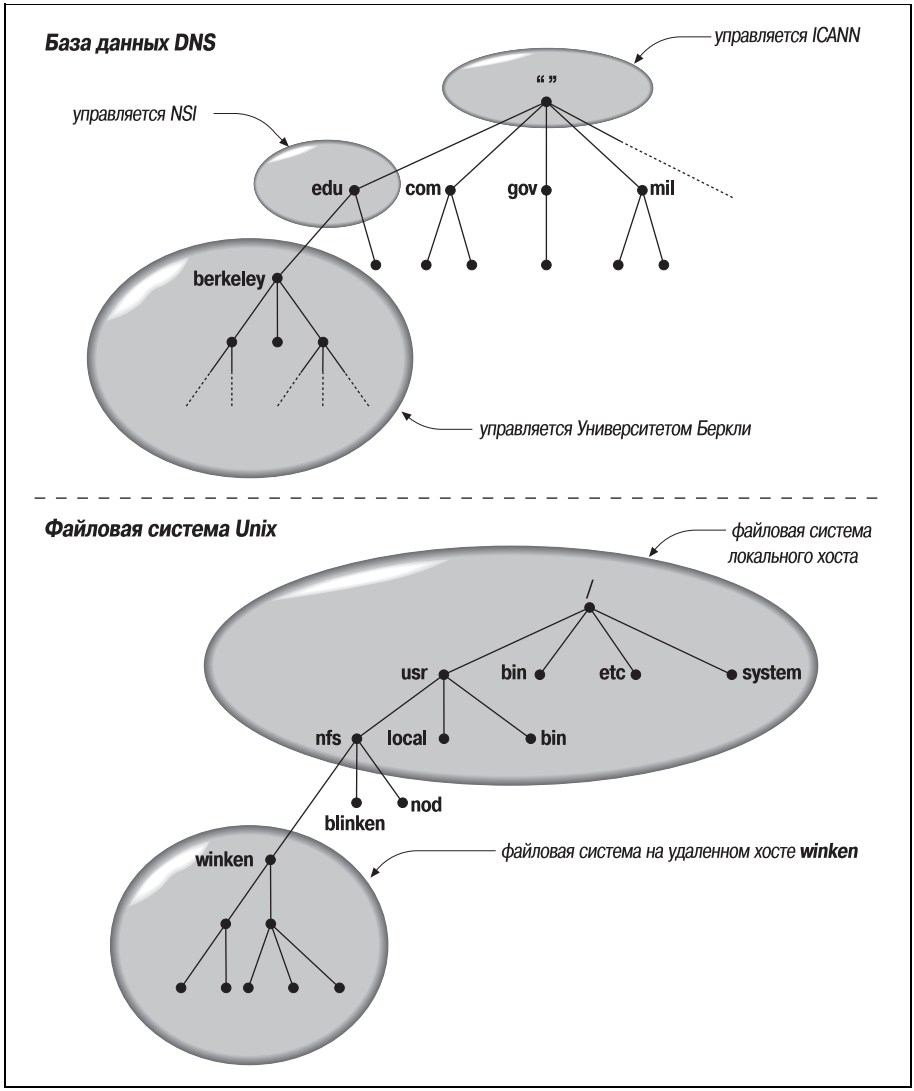

Рис. 1.3. *Удаленное управление доменами разных уровней и файловыми системами*

гичным образом домены могут содержать узлы и поддомены. Домен включает в себя те узлы и поддомены, доменные имена которых принадлежат этому домену.

У каждого узла в сети есть доменное имя, которое является указателем на информацию об узле (рис. 1.5). Эта информация может включать IP-адрес, информацию о маршрутизации почтовых сообщений и другие данные. Узел может также иметь один или несколько *псевдонимов доменного имени*, которые являются просто указателями на ос-

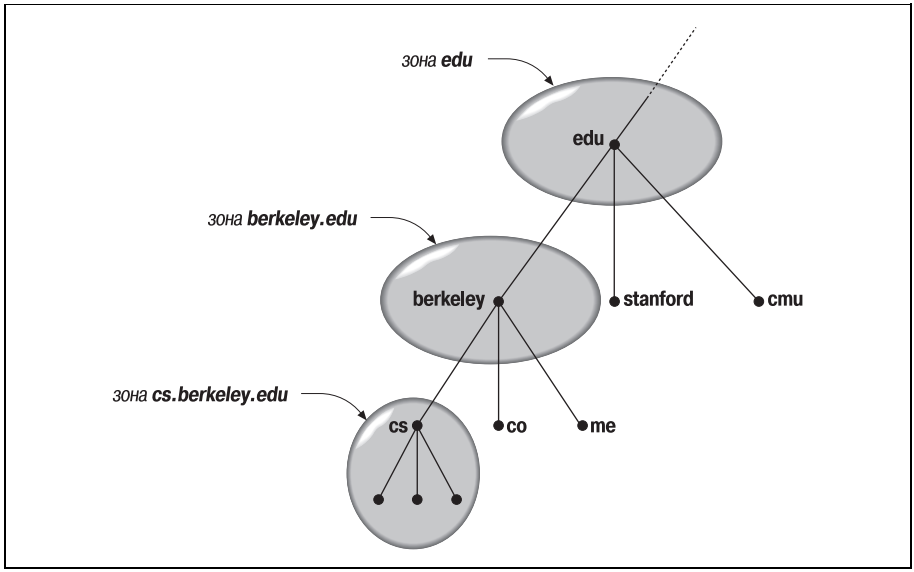

Рис. 1.4. *Зоны edu, berkeley.edu и cs.berkeley.edu*

новное (официальное, или *каноническое*) доменное имя. На рис. 1.5 *mailhub.nv...* является псевдонимом канонического имени *rincon.ba.ca...*

Для чего нужна столь сложная структура? Чтобы решить проблемы, существовавшие при использовании *HOSTS.TXT*. К примеру, строгая иерархичность доменных имен устраняет угрозу конфликтов имен.

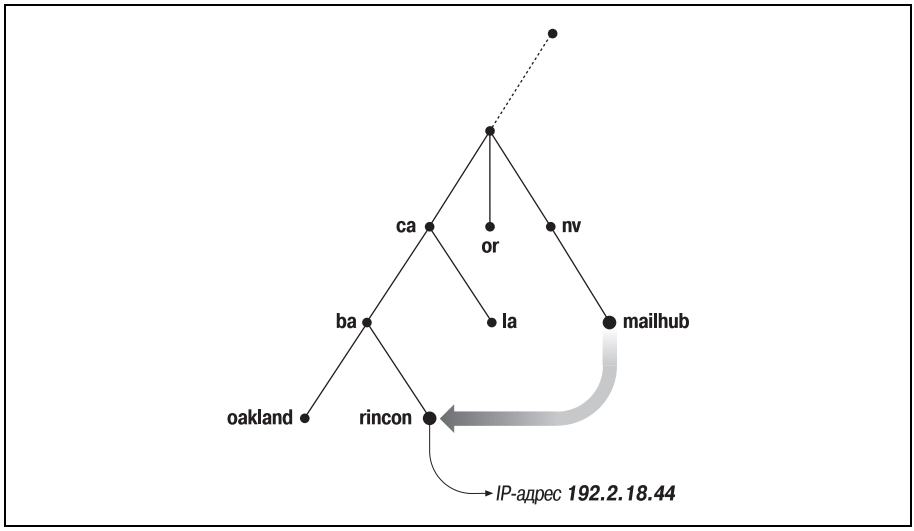

Рис. 1.5. *Псевдоним в DNS, ссылающийся на каноническое имя*

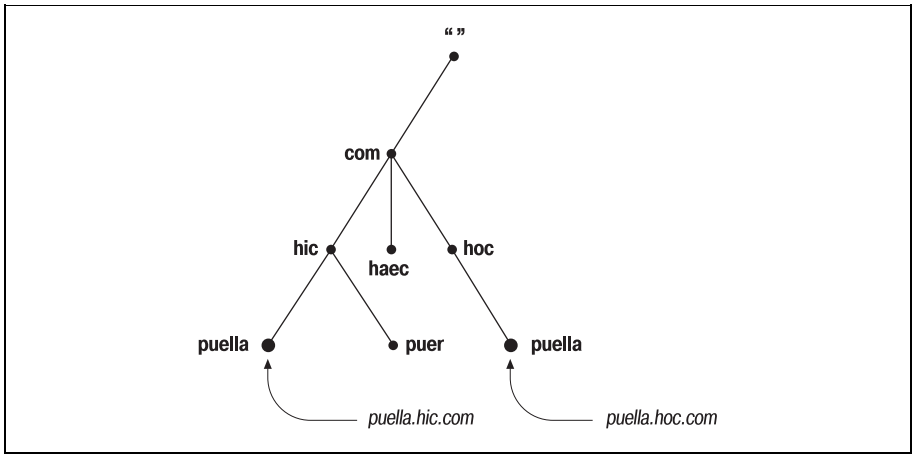

Рис. 1.6. *Решение проблемы конфликтов имен*

Имя каждого домена уникально, так что организация, управляющая доменом, вольна придумывать имена поддоменов, входящих в этот домен, самостоятельно. Независимо от выбранных имен, имена эти не будут конфликтовать с другими доменными именами, поскольку заканчиваются уникальным именем домена, сопровождаемого только этой организацией. Так, организация, ответственная за домен *hic.com* может дать узлу имя *puella* (рис. 1.6), поскольку известно, что доменное имя узла будет заканчиваться уникальным доменным именем *hic.com*.

История пакета BIND

Первая реализация системы доменных имен называлась JEEVES, и была разработана самим Полом Мокапетрисом. Более поздняя реализация носит название BIND, это аббревиатура от *Berkeley Internet Name Domain*, и была разработана для операционной системы 4.3 BSD Unix (Беркли) Кевином Данлэпом. В настоящее время развитием и сопровождением пакета BIND занимается Internet Software Consortium.[1]

На пакете BIND мы и сосредоточимся в данной книге, поскольку именно BIND является сегодня наиболее популярной и распространенной реализацией DNS. Пакет доступен на большинстве разновидностей системы Unix и входит в стандартную конфигурацию систем от большинства поставщиков Unix. BIND также был портирован на платформу Microsoft Windows NT.

[1] Более подробно об организации Internet Software Consortium и ее разработках в области BIND можно узнать по адресу *http://www.isc.org/bind.html*

Надо ли мне использовать DNS?

Несмотря на очевидную пользу DNS, существуют случаи, в которых ее применение неоправданно. Помимо DNS существуют и другие механизмы разрешения имен, некоторые из которых могут быть составной частью операционной системы. Иногда затраты сил и времени, связанные с сопровождением зон и DNS-серверов, превышают все возможные выгоды. С другой стороны, возможна ситуация, когда нет другого выбора, кроме как установить и поддерживать DNS-серверы. Вот некоторые указания, которые помогут сориентироваться и принять решение:

Если вы подключены к сети Интернет...

...DNS является жизненной необходимостью. DNS можно считать общепринятым языком сети Интернет: почти все сетевые службы в Интернете, включая World Wide Web, электронную почту, удаленный терминальный доступ и передачу файлов, используют DNS.

С другой стороны, подключение к сети Интернет вовсе не означает, что не удастся избежать самостоятельной установки и сопровождения нужных пользователю зон. В случае ограниченного числа узлов всегда можно найти уже существующую зону и стать ее частью (подробнее – в главе 3 «С чего начать?»). Как вариант, можно найти кого-то, кто позаботится о размещении зоны. Если пользователь платит интернет-провайдеру за подключение, обычно существует возможность разместить свою зону на технологических мощностях этого провайдера. Существуют также компании, предоставляющие подобную услугу за отдельную плату.

Если же узлов много или очень много, то, скорее всего, понадобится самостоятельная зона. Если вы хотите иметь непосредственный контроль над зоной и серверами имен, то вступаете на путь администрирования и сопровождения. Читайте дальше!

Если у вас интернет-сеть на основе протоколов TCP/IP...

...то DNS, вероятно, не помешает. В данном случае под интернет-сетью мы не подразумеваем простую сеть из одного сегмента Ethernet и нескольких рабочих станций, построенную на протоколах TCP/IP (такой вариант описан в следующем разделе), но достаточно сложную «сеть сетей». Например – множество Appletalk-сегментов плюс несколько сетей Apollo token ring, объединенных вместе.

Если интернет-сеть является преимущественно гомогенной, и узлы не нуждаются в службе DNS (скажем, в случае крупной интернет-сети DECnet или OSI), вполне возможно, что можно обойтись без нее. Но в случае разнородных хостов, в особенности, если некоторые из них работают под управлением Unix, DNS пригодится. Система упростит распространение информации об узлах и избавит ад-

министратора от необходимости выдумывания своей схемы распространения таблиц хостов.

Если у вас собственная локальная сеть...

...и эта сеть не соединена с большей сетью, вполне возможно обойтись без DNS. Можно попробовать использовать службу Windows Internet Name Service (WINS) от Microsoft, таблицы хостов, или Network Information Service (NIS) от Sun.

В случаях, когда требуется распределенное администрирование, либо присутствуют сложности с синхронизацией данных в сети, использование DNS может иметь смысл. И если планируется подключение вашей сети к другой, скажем, к корпоративной интернет-сети, либо к Интернету, стоит заранее заняться настройкой собственных зон.

2

- *Пространство доменных имен*
- *Пространство доменных имен сети Интернет*
- *Делегирование*
- *DNS-серверы и зоны*
- *DNS-клиенты*
- *Разрешение имен*
- *Кэширование*

Как работает DNS

– Что толку в книжке, – подумала Алиса, – если в ней нет ни картинок, ни разговоров?

Система доменных имен – это, прежде всего, база данных, содержащая информацию об узлах сети. Да, вкупе с этим вы получаете целый набор всякой всячины: чудные имена с точками, серверы, которые подключаются к сети, загадочное «пространство имен». И все же следует помнить, что, в конечном итоге, услуга, предоставляемая DNS, сводится к получению информации об узлах сети.

Мы уже рассмотрели некоторые важные аспекты работы DNS, включая архитектуру «клиент-сервер» и структуру базы данных. Однако мы не особенно вдавались в детали и не объясняли работу основных механизмов DNS.

В этой главе мы объясним и проиллюстрируем процессы, на которых построена работа системы доменных имен. Будет представлена терминология, которая позволит прочесть и понять оставшуюся часть книги (а также вести интеллектуальные беседы с друзьями – администраторами DNS).

Но сначала все-таки взглянем чуть внимательнее на концепции, представленные в предшествующей главе. Попробуем углубиться в детали и придать им особый ракурс.

Пространство доменных имен

Распределенная база данных системы доменных имен индексируется по именам узлов. Каждое доменное имя является просто путем в ог-

ромном перевернутом дереве, которое носит название *пространства доменных имен*. Иерархическая структура дерева, отображенная на рис. 2.1, похожа на структуру файловой системы Unix. Единственный корень дерева расположен наверху.[1] В файловых системах Unix эта точка называется корневым каталогом и представлена символом «слэш» (/). В DNS же это просто «корень» («root»). Как и файловая система, дерево DNS может иметь любое количество ответвлений в любой точке пересечения, или *узле*. Глубина дерева ограничена и может достигать 127 уровней (предел, до которого вы вряд ли когда-нибудь доберетесь).

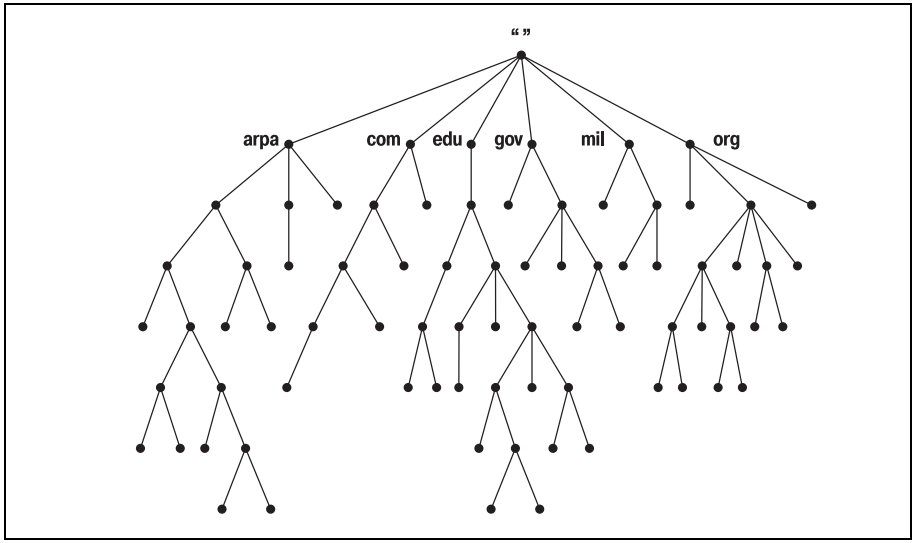

Рис. 2.1. *Структура пространства имен DNS*

Доменные имена

Каждому узлу дерева соответствует текстовая метка, длина которой не может превышать 63 символов, причем использование символа точки недопустимо. Пустая (нулевой длины) метка зарезервирована для корня. Полное *доменное имя* произвольного узла дерева – это последовательность меток в пути от этого узла до корня. Доменные имена всегда читаются от собственно узла к корню («вверх» по дереву), причем метки разделяются точкой.

Если метка корневого узла должна быть отображена в доменном имени, она записывается как символ точки, например, так: «www.oreilly.com.». (На самом деле имя заканчивается точкой-разделителем и пустой меткой корневого узла.) Сама по себе метка корневого узла за-

[1] Понятно, что это дерево компьютерщика, а не ботаника.

писывается исключительно из соображений удобства, как самостоятельная точка (.). Как следствие, некоторые программы интерпретируют имена доменов, заканчивающиеся точкой, как *абсолютные*. Абсолютное доменное имя записывается относительно корня и однозначно определяет расположение узла в иерархии. Абсолютное доменное имя известно также под названием *полного доменного имени*, обозначаемого аббревиатурой *FQDN* (*fully qualified domain name*). Имена без завершающей точки иногда интерпретируются относительно некоторого доменного имени (не обязательно корневого) точно так же, как имена каталогов, не начинающиеся с символа «/» (слэш), часто интерпретируются относительно текущего каталога.

В DNS «братские» узлы, то есть узлы, имеющие общего родителя, должны иметь разные метки. Такое ограничение гарантирует, что доменное имя единственно возможным образом идентифицирует отдельный узел дерева. Это ограничение на практике не является ограничением, поскольку метки должны быть уникальными только для братских узлов одного уровня, но не для всех узлов дерева. То же ограничение существует в файловых системах Unix: двум «единоутробным» каталогам или двум файлам в одном каталоге не могут быть присвоены одинаковые имена. Невозможно создать два узла *hobbes.pa.ca.us* в пространстве доменных имен, и невозможно создать два каталога */usr/bin* (рис. 2.2). Тем не менее, можно создать пару узлов с именами *hobbes.pa.ca.us* и *hobbes.lg.ca.us*, точно так же, как можно создать пару каталогов с именами */bin* и */usr/bin*.

Домены

Домен – это просто поддерево в пространстве доменных имен. Доменное имя домена идентично доменному имени узла на вершине домена. Так, к примеру, вершиной домена *purdue.edu* является узел с именем *purdue.edu* (рис. 2.3).

Аналогичным образом, в корне файловой системы */usr* мы ожидаем увидеть каталог с именем */usr* (рис. 2.4).

Каждое доменное имя в поддереве считается принадлежащим домену. Поскольку доменное имя может входить в несколько поддеревьев, оно также может входить в несколько доменов. К примеру, доменное имя *pa.ca.us* входит в домен *ca.us* и при этом является также частью домена *us* (рис 2.5).

Так что теоретически домен – это просто сегмент пространства доменных имен. Но если домен состоит только из доменных имен и других доменов, где узлы сети – хосты? Ведь домены-то – это группы хостов, верно?

Узлы сети, разумеется, присутствуют, и представлены они доменными именами. Следует помнить, что доменные имена являются просто указателями в базе данных DNS. «Хосты» – это доменные имена, которые указывают на информацию по каждому конкретному хосту. Домен со-

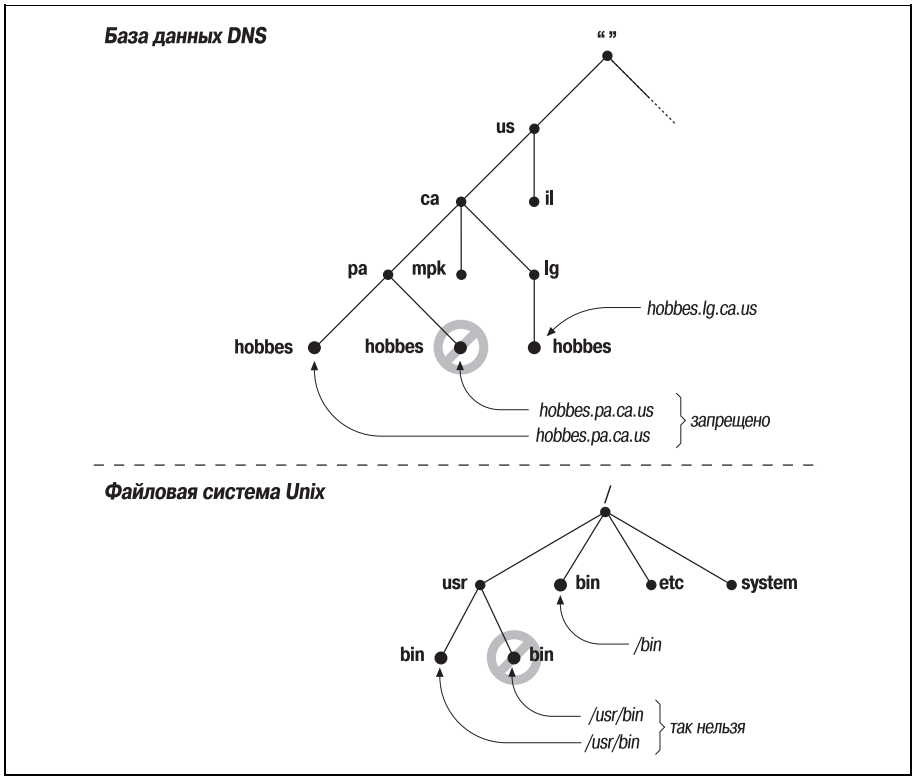

Рис. 2.2. *Обеспечение уникальности доменных имен и путевых имен в файловых системах Unix*

держит все узлы сети, доменные имена которых в него входят. Узлы сети связаны *логически*, зачастую по географическому или организационному признаку, и совсем необязательно – сетью, адресом или типом используемого оборудования. Десяток узлов, входящих в разные сети, возможно, даже расположенных в разных странах, может принадлежать одному-единственному домену.[1]

Доменные имена, соответствующие листьям дерева, как правило, относятся к отдельным узлам сети и могут указывать на сетевые адреса,

[1] Предостережение: не стоит путать домены DNS с доменами в службе NIS, Network Information Service от Sun. Несмотря на то, что домен NIS – это тоже группа узлов, а доменные имена в обеих службах имеют сходную организацию, концептуальные различия достаточно велики. В NIS используется иерархическая организация имен, но иерархия на этом и кончается: узлы сети, входящие в один домен NIS, разделяют определенную информацию об узлах и пользователях, но не могут опрашивать пространство имен NIS с целью поиска информации в других доменах NIS. Домены NT, обеспечивающие управление доступом и службами безопасности, также не имеют никакого отношения к доменам DNS.

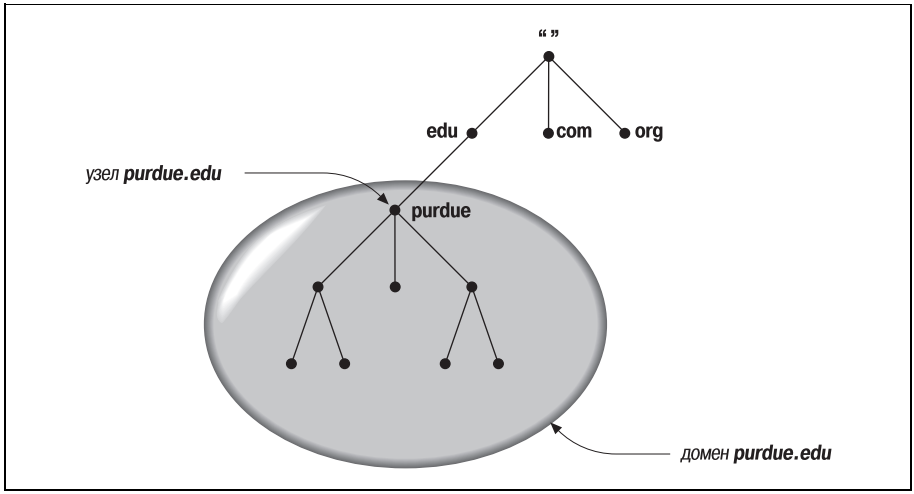

Рис. 2.3. *Домен purdue.edu*

информацию об оборудовании и о маршрутизации почты. Доменные имена внутри дерева могут идентифицировать отдельные узлы, *а также* могут указывать на информацию об этом домене. Имена доменов внутри дерева не привязаны жестко к тому или другому варианту. Они могут представлять как домен (имени которого соответствуют), так и отдельный узел в сети. Так, *hp.com* является именем домена компании Hewlett-Packard и доменным именем узлов, на которых расположен главный веб-сервер Hewlett-Packard.

Тип информации, получаемой при использовании доменного имени, зависит от контекста применения имени. Посылка почтовых сообщений кому-то в домен *hp.com* приводит к получению информации о маршру-

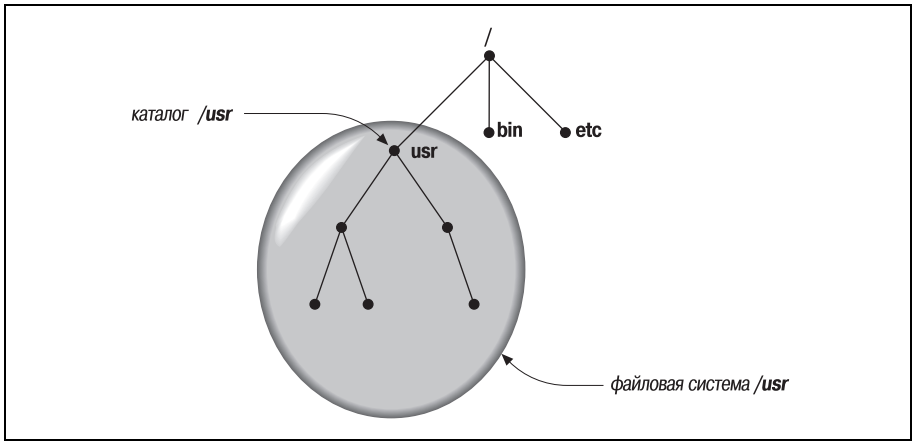

Рис. 2.4. *Каталог /usr*

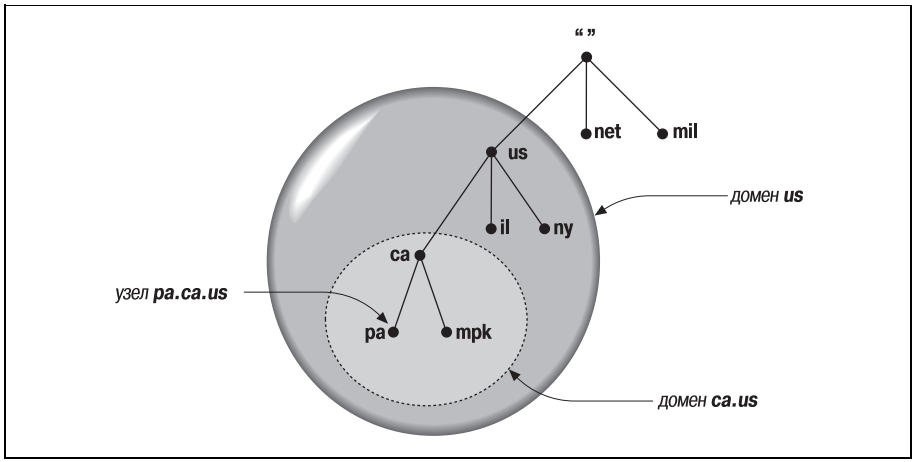

Рис. 2.5. *Узел, входящий в несколько доменов*

тизации почты, открытие telnet-сеанса связи с этим доменом приводит к поиску информации об узле (на рис. 2.6, к примеру, это IP-адрес узла *hp.com*).

Домен может содержать несколько поддеревьев, которые носят название *поддоменов*.[1]

Самый простой способ выяснить, является ли домен поддоменом другого домена, – сравнить их доменные имена. Доменное имя поддомена заканчивается доменным именем родительского домена. К примеру,

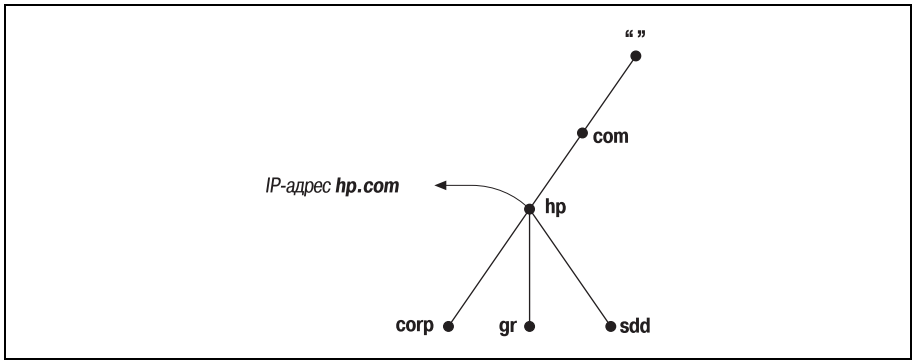

Рис. 2.6. *Внутренний узел дерева, связанный как с информацией о конкретном узле сети, так и с иерархической*

[1] Термины *домен* и *поддомен* в документации по DNS и BIND зачастую используются в качестве взаимозаменяемых. В настоящей книге мы используем термин «поддомен» в качестве относительного: домен является поддоменом другого домена, если корень поддомена принадлежит включающему домену.

домен *la.tyrell.com* должен являться поддоменом домена *tyrell.com*, поскольку имя *la.tyrell.com* заканчивается именем *tyrell.com*. Аналогичным образом этот домен является поддоменом домена *com*, как и домен *tyrell.com*.

Помимо относительных степеней в идентификации доменов в качестве входящих в другие домены, используется также *уровневая* классификация доменов. В списках рассылки и конференциях Usenet можно встретить термины *домен высшего уровня* и *домен второго уровня*. Эти термины просто определяют положение домена в пространстве доменных имен:

- Домен высшего уровня в качестве непосредственного родителя имеет корневой домен.
- Домен первого уровня в качестве непосредственного родителя (также) имеет корневой домен (то есть он является доменом высшего уровня).
- Домен второго уровня в качестве непосредственного родителя имеет домен первого уровня, и т. д.

Записи ресурсов

Информация, связанная с доменными именами содержится в *записях ресурсов* (RRs, resource records).[1] Записи разделяются на классы, каждый из которых определяет тип сети или программного обеспечения. В настоящее время существуют классы для интернет-сетей (на основе семейства протоколов TCP/IP), сетей на основе проколов Chaosnet, а также сетей, которые построены на основе программного обеспечения Hesiod. (Chaosnet – старая сеть, имеющая преимущественно историческое значение).

Популярность класса интернет-сетей значительно превосходит популярность остальных классов. (Мы не уверены, что где-то до сих пор используется класс Chaosnet, а использование класса Hesiod в основном ограничивается пределами Массачусетского технологического института – MIT). В настоящей книге мы сосредоточимся на классе интернет-сетей.

В пределах класса записи делятся на типы, которые соответствуют различным видам данных, хранимых в пространстве доменных имен. В различных классах определяются различные типы записей, хотя некоторые типы могут являться общими для нескольких классов. Так, практически в каждом классе определен тип *адрес*. Каждый тип записи в конкретном классе определяет формат, который должны соблюдать все RR-записи, принадлежащие этому классу и имеющие данный

[1] В дальнейшем мы будем использовать выражение RR-запись, или просто RR. – *Примеч. науч. ред.*

тип. (Подробно типы и форматы RR-записей для интернет-сетей описаны в приложении A «Формат сообщений DNS и RR-записей».)

Не беспокойтесь, если информация кажется излишне схематичной: записи класса интернет будут более подробно рассмотрены позже. Наиболее часто используемые RR-записи описаны в главе 4 «Установка BIND», а более полный перечень приводится в приложении A.

Пространство доменных имен сети Интернет

До сих пор мы говорили о теоретической структуре пространства доменных имен и о том, какого сорта данные могут в нем содержаться, и даже обозначили – в наших (порой выдуманных) примерах – типы имен, которые можно встретить в этом пространстве. Но это ничем не поможет в расшифровке доменных имен, которые ежедневно встречаются пользователям в сети Интернет.

Система доменных имен довольно либеральна в отношении меток, составляющих доменные имена, и не определяет *конкретные* значения для меток определенного уровня. Если администратор управляет сегментом пространства имен, он и определяет семантику доменных имен, входящих в сегмент. Черт возьми, администратор может присвоить поддоменам в качестве имен буквы алфавита от A до Z, и никто его не остановит (хотя сильно будут советовать этого не делать).

При этом существующее пространство доменных имен сети Интернет обладает некоторой сложившейся структурой. Это в особенности касается доменов верхних уровней, доменные имена в которых подчиняются определенным традициям (которые не являются правилами, поскольку их можно нарушать, и так не раз бывало). Эти традиции вносят в доменные имена некоторую упорядоченность. Понимание традиций – это большое подспорье для попыток расшифровки доменных имен.

Домены высшего уровня

Изначально домены высшего уровня делили пространство имен сети Интернет на семь доменов:

com

Коммерческие организации, такие как Hewlett-Packard (*hp.com*), Sun Microsystems (*sun.com*) или IBM (*ibm.com*).

edu

Образовательные организации, такие как Калифорнийский университет Беркли (*berkeley.edu*) и университет Пердью (*purdue.edu*).

gov

Правительственные организации, такие как NASA (*nasa.gov*) и Национальный научный фонд (*nsf.gov*).

mil

Военные организации, такие как армия (*army.mil*) и флот (*navy.mil*) США.

net

В прошлом организации, обеспечивающие работу сетевой инфраструктуры, такие как NSFNET (*nsf.net*) и UUNET (*uu.net*). В 1996 году домен *net*, как и *com*, стал доступен для всех коммерческих организаций.

org

В прошлом некоммерческие организации, такие как Фонд электронной границы (Electronic Frontier Foundation) (*eff.org*). Как и в случае с доменом *net*, ограничения были сняты в 1996 году.

int

Международные организации, такие как NATO (*nato.int*).

Существовал еще один домен высшего уровня, который назывался *arpa* и использовался в процессе перехода сети ARPAnet от таблиц узлов к системе доменных имен. Изначально все узлы ARPAnet имели доменные имена, принадлежавшие домену *arpa*, так что их было несложно отличать. Позже они разбрелись по различным поддоменам организационных доменов высшего уровня. При этом домен *arpa* все еще используется, и чуть позже мы расскажем, как именно.

Можно заметить некоторую националистическую предрасположенность в примерах – прежде всего речь идет об организациях США. Это легче понять – и простить – если вспомнить, что сеть Интернет началась с сети ARPAnet, исследовательского проекта, который финансировался правительством США. Никто и не предполагал, что создание ARPAnet увенчается подобным успехом и что этот успех в итоге приведет к созданию международной сети Интернет.

В наши дни эти домены называются *родовыми доменами высшего уровня* (generic top-level domains или gTLDs). В начале 2001 года их список расширится, и будет включать домены *name*, *biz*, *info*, а также *pro*, которые создаются, чтобы приспособить иерархию к взрывному росту сети Интернет и удовлетворить потребность в доменном «пространстве». Организация, ответственная за управление системой доменных имен сети Интернет, – Internet Corporation for Assigned Names and Numbers (ICANN) – утвердила добавление новых gTLD, а также явно конкретизированных *aero*, *coop* и *museum* в конце 2000 года. Информация о работе организации ICANN и новых доменах высшего уровня доступна по адресу *http://www.icann.org*.

Чтобы справиться с потребностями быстро растущих сегментов сети Интернет, расположенных во многих странах мира, пришлось пойти на определенные компромиссы, связанные с пространством доменных имен Интернета. Было решено не придерживаться схемы организаци-

онного деления доменов высшего уровня, но разрешить использование географических обозначений. Новые домены высшего уровня были зарезервированы (но не во всех случаях созданы) для каждой страны. Эти доменные суффиксы соответствуют существующему международному стандарту ISO 3166.[1] Стандарт ISO 3166 определяет официальные двухбуквенные сокращения для каждой страны мира. Текущий список доменов высшего уровня приведен в приложении D «Домены высшего уровня».

По нисходящей

В пределах упомянутых доменов верхних уровней существующие традиции и степень их соблюдения начинает варьироваться. Некоторые из доменов высшего уровня, упомянутых в стандарте ISO 3166, в общем и целом следуют исходной организационной схеме США. К примеру, в домен высшего уровня Австралии, *au*, входят такие поддомены, как *edu.au* и *com.au*. Некоторые из прочих доменов ISO 3166 следуют примеру домена *uk* и порождают поддомены организационного деления, например *co.uk*, для корпоративного использования, а скажем *ac.uk* – для нужд академического сообщества. Тем не менее в большинстве случаев географические домены высшего уровня имеют организационное деление.

Однако это не относится к домену высшего уровня *us*. В домен *us* входит 50 поддоменов, которые соответствуют – попробуйте угадать! – пятидесяти штатам.[2] Имя каждого из этих поддоменов соответствует стандартному двухбуквенному сокращению названия штата, именно эти сокращения стандартизированы почтовой службой США. В пределах каждого поддомена деление в основном такое же, географическое: большинство поддоменов соответствуют отдельным городам. В пределах поддомена города все прочие поддомены обычно соответствуют отдельным узлам.

Чтение доменных имен

Теперь, познакомившись со свойствами имен доменов высшего уровня и структурой пространства доменных имен, читатели, вероятно, смогут гораздо быстрее определять кроющийся в именах доменов смысл. Попрактикуемся на следующих примерах:

---

[1] За исключением Великобритании. В соответствии со стандартом ISO 3166 и традициями Интернета доменный суффикс высшего уровня для Великобритании должен быть *gb*. На деле же большинство организаций Великобритании и Северной Ирландии (то есть Соединенного Королевства) используют суффикс *uk*. А еще они ездят по неправильной стороне дороги.

[2] В действительности поддоменов в домене *us* чуть больше: один для Вашингтона (округ Колумбия), один для острова Гуам, и т. д.