

Linux DNS Server Administration

Linux DNS Server 管理指南

本书由 Linux 专家编著 专业人士翻译
讨论 Linux 管理员工作中的最关键主题
献给在 Linux 环境中工作的网络专业人士

[美] Craig Hunt 著
陈圣琳 王欣 卢峰 等译
朱友芹 审校



电子工业出版社
Publishing House of Electronics Industry
URL: <http://www.phei.com.cn>

Linux DNS Server Administration

Linux DNS Server 管理指南

[美] Craig Hunt 著

陈圣琳 王欣 卢峰 等译

朱友芹 审校

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 提 要

域名服务系统DNS在因特网发展过程中起了重大推动作用,而Linux是建立DNS服务器的优秀平台,BIND软件最初将DNS引入网络并且目前仍然是应用最广泛的DNS软件。本书讲解了Linux上的DNS和BIND,主要针对Linux系统。全面介绍了其工作原理、基本配置、高级配置、系统维护等技术。

读者对象: 本书适合Linux系统管理人员阅读。



Copyright©2001 SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501. World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

本书英文版由美国SYBEX公司出版,SYBEX公司已将中文版独家版权授予中国电子工业出版社及北京美迪亚电子信息有限公司。未经许可,不得以任何形式和手段复制或抄袭本书内容。

图书在版编目(CIP)数据

Linux DNS Server管理指南/(美)汉特(Hunt, C.)著;陈圣琳等译.-北京:电子工业出版社, 2001.6

书名原文: Linux DNS Server Administration

ISBN 7-5053-6783-8

I. L… II. ①汉… ②陈… III. Linux操作系统 IV. TP316.89

中国版本图书馆CIP数据核字(2001)第040041号

书 名: **Linux DNS Server管理指南**

著 者: [美] Craig Hunt

译 者: 陈圣琳 王 欣 卢 峰 等

审 校: 朱友芹

责任编辑: 陈 宇

印 刷 者: 北京天竺颖华印刷厂

装 订 者: 三河金马印装有限公司

出版发行: 电子工业出版社 URL:<http://www.phei.com.cn>

北京市海淀区万寿路173信箱 邮编: 100036 电话: 68279077

北京市海淀区翠微东里甲2号 邮编: 100036 电话: 68252397

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 18.25 字数: 470 千字

版 次: 2001年6月第1版 2001年6月第1次印刷

书 号: **ISBN 7-5053-6783-8**

TP·3812

定 价: 32.00元

版权贸易合同登记号 图字: 01-2000-0417

凡购买电子工业出版社的图书,如有缺页、倒页、脱页,请向购买书店调换,若书店售缺,请与本社发行部联系调换。

献给Internet无私的先驱之一：Jon Postel。

致 谢

没有一本书是“个人英雄主义”的结果，从最开始就凝聚着团队的力量。想当初，我提出这套丛书的想法时，正在一家餐馆里与Neil Edde共进一盘烤肉。虽然Neil最终没有被分派到该项目组中，但是他鼓励我向Sybex的出版商提出这个计划，所以他应该得到由衷的感谢。

特别感谢该丛书的责任编辑Maureen Adams。在出版的商务方面，有发行期限的督促，因为业务人员和市场人员的计划都是基于期限来制定的。Maureen取消这些限制，延长了我们的出版期限，使我们有足够的时间来保证本丛书技术上的质量。

Molly Glover的任务是保证我们按计划行事，她做得很出色：提供了足够的动力使我的工作不断进展，但又不至于欲速不达。在较强的出书压力下，身边有Molly这样的保持头脑冷静的指引者，也可以从容不迫。

Sarah Lemaire是本书的文字编辑，从书中较高的文法结构中随处可见她做的润色，同时她很尊重我的写作方式，保持了整体风格。

Will Deutsch (UC Davis) 是我的技术编辑，这是第二次与我合作著书。非常感谢Will诚恳的建议。他以前从事研究工作，因此可以从一种不同的角度来看待问题，对一些系统管理员会感兴趣的课题，Will有很强的洞察力，使我能够在这些方面比其他系统管理员了解的更多、看得更远。

我要感谢所有业务人员和艺术指导，对于该书，他们都付出了辛勤的劳动：电子出版专家Adrian Woolhouse；插图画家Tony Jonick；校对员Dave Nash、Laurie O'Connell、Nancy Riddiough、Suzanne Stein和Nathan Whiteside；索引员Matthew Spence。Sybex全体人员为读者奉献了出众的作品。

我还想感谢那些致力于BIND软件维护、DNS设计和开发的人。不仅仅是因为他们建立了适用于任何人使用的令人赞叹的代码程序，还因为这些程序和DNS系统改变了这个世界的的生活方式。所有使用Internet的人们都应该感谢他们。

最后，我要感谢Kathy。当孩子们都在外面玩的时候，她是我惟一的助手。感谢她对这份工作有如此的耐心。

序

Linux应该有一套自己的系统管理图书，而不能把它看作是其他操作系统的姊妹篇。很多Linux系统管理员依靠Unix系统手册，因为Linux和Unix基本上是一样的。既然如此，对于一个专业的Linux系统管理员，什么时候才能真正的“专业”，在面对自己的老板和用户的时候，不能总是拿Unix来看待Linux的问题。Linux不是业余爱好者的操作系统，而是可以用于关键应用的企业级和部门级操作系统，作为这样系统的管理员，必须要达到很专业的程度，而不能依赖于“基本”上一样的Unix系统。所以，你需要关于Linux操作系统的最准确的管理指南。

当一个系统管理员试图将一些重点讲述Linux的图书集中到一起时，也会发现一些问题：不同出版社的图书，结构和风格截然不同；即使是同一出版社，不同编辑也导致图书结构不同，而这些不一致就导致整体的晦涩难懂。没有统一性，寻找信息就很困难，而且由于完整性方面的缺陷，也有可能根本找不到。为了解决这些问题，我设计了这套丛书，其中的每一本都有统一的结构和风格，而且是绝对Linux的。

作为丛书作者，我对其中的每一本书都字斟句酌，尽我最大的努力保证内容的完整性、准确性和连贯性。如果你阅读过我的某些书，例如《TCP/IP网络管理》或者《Linux网络服务器》，你就会知道我的写作风格是简明和干练的，这种风格贯穿了整个丛书系列。

最终，你也有了可以依赖的Linux书库，这是我的希望。你可以从书中迅速准确地找到所需要的技术信息。

Craig Hunt

简介

在因特网的发展过程中，有两项技术起了重大的推动作用。其中WWW是Internet应用的关键技术之一，而另一门技术就是域名服务系统（DNS）。早在1988年，广泛应用DNS之前，因特网上的主机已经数以千计了，现在已是上百万了。这并不单纯是几个命令就能解决的问题，还涉及到资源的可利用性。

域名服务系统（DNS）是管理因特网主机名的工具。在域名服务系统（DNS）出现之前，所有的因特网主机名都存储在普通文件中，而且网络上的每台机器都要保留一个副本。这种文件的局限性阻碍了网络的发展，域名服务系统（DNS）的出现突破了这种限制，它使用分布式多级数据库系统代替了普通文件的存储方式。因此，用户可以摆脱集中维护主机文件的限制。现在，网络上的每个组织都自行维护自己的域名数据库，系统管理员负责维护系统中的DNS。这本书将会告诉用户如何去做。

Linux是建立DNS服务器的优秀平台。Linux的可靠性带有传奇色彩，曾经有用户使用Linux服务器正常运行一年多没有中断。

运行于系统之上的名称服务器软件的可靠性与操作系统的可靠性同等重要。Linux的优势之一在于它使用了伯克利因特网域名（BIND）软件。BIND是最初将DNS引入了网络的软件，在20世纪80年代后期得以广泛应用，目前，BIND仍然是应用最广泛的DNS软件。BIND已被成千上万的用户验证，证明了其可靠性。

这本书介绍了Linux上的DNS和BIND知识，主要针对Linux系统。有些书中囊括了各种版本的Unix，其中肯定包括各种版本的BIND，甚至包括那些过时的版本，之所以这样是因为有些Unix商业版仍在广泛应用中，这些系统安装了旧版BIND软件。Linux做了大量的工作来提供最新的软件并使之易于更新。本书中的例子都是基于最新版BIND的，这些Linux的实例都可以应用到用户的系统中，用户无需深入研究各种Unix系统中不相关的例子。这些BIND的实例彼此连贯且易理解。鉴于这些考虑，本书进行了合理的组织，帮助用户轻松阅读。

哪些读者适合阅读本书

这本书适合那些在Linux上运行DNS的用户，或任何想了解DNS工作原理的人。如果读者仔细阅读以下序言部分，或许会发现这本书也是你所需要的。

对于Linux系统管理员，本书是他们获得DNS知识的主要来源。书中详细说明了如何在Linux平台上建立DNS服务器；还提供了在Linux上编译、安装和配置BIND的实例；讨论了Linux安全特征以及其他参考书中可能忽略的有关Linux的信息。

甚至Unix和Windows NT的系统管理员也会发现这是本有用的参考书。因为书中详细说明了DNS协议以及配置DNS的具体参考值。本书组织清晰有序，对于DNS工作原理和特定配置参数的独到见解会对任何运行DNS的用户有所帮助——即使读者不使用Linux系统。

阅读本书不需要读者具备大量的DNS知识，但需要用户对Linux和系统管理较为熟悉。如果读者需要了解这些知识，参考其他提供所需知识背景的书籍。

这本书有效地强调了用户掌握DNS知识的重要性。本书采用面向任务的结构，内容详实，循序渐进，向读者一步步介绍如何操作以及何时进行操作。此外，还解释了这样操作的原因。书中对DNS工作原理的解释将帮助用户做出服务器的最佳配置方案。

这本书如何组织

本书分五部分：工作原理、基本配置、高级配置、系统维护和附录。这五个部分包括12个章节及4个附录。

本书采用整体设计，使用户在阅读时具有连贯性。该书首先介绍了DNS的基本知识，接着是每个管理员应该掌握的基本BIND配置技巧，随后又讨论了特殊情况下进行的专门配置选项，最后给出了维护和运行DNS服务器的相关信息。

很多章节涉及到其他章节的知识，书中都给出了相关章节的指针。已经理解基本DNS协议和结构的读者可以直接阅读基本配置部分。经验丰富的管理员如果熟悉所有的BIND基本配置就可以直接跳到高级配置部分。但对大多数系统管理员来说，通读本书都会受益匪浅。

尽管这本书需要通读，但笔者可以理解很多系统管理员没有时间阅读全书。他们必须直接解决所遇问题、找到如何解决的答案。为方便这些用户阅读，书中在涉及某些主体时给出了必要的背景知识以及需要深入参考的相关章节指针。

第一部分：工作原理

这一部分帮助读者理解名称服务是如何工作的。这里介绍了分布式多级数据库系统、DNS协议以及在Linux系统中运行DNS的BIND软件。

第1章 DNS体系结构 第1章介绍了DNS的体系结构。其中阐述了DNS促进Internet发展的特征，解释了域名结构、域名获得方法、顶级域名以及域名委托等。此外，还讨论了根服务器的作用、查询过程、信息的动态分配以及服务器之间的关系。

第2章 DNS协议 第2章详细介绍了查询过程。其中解释了如何交换数据报，完成主机名到IP地址的解析，描述了保持分布式DNS数据库同步的协议。

第3章 BIND软件 DNS是通过BIND软件实现的。本章介绍了BIND的构件，BIND在不同服务器上的配置，BIND服务器如何进行DNS架构，named和ndc命令的作用等。同时，还为用户的Linux提供了下载和编译最新版BIND的指令。

第二部分：基本配置

本部分提供了每个域名管理员都需要掌握的基本配置信息。它包括Linux中DNS系统配置的每一步骤，从客户端配置到主服务器配置。

第4章 配置解析器 每个网络系统都有一个解析器负责处理用户向名称服务器发出的请求，客户端和服务端都需要配置解析器。本章就给出了解析器的缺省配置信息，描述了每个解析器配置选项，便于用户用来创建自定义配置。文中举出实例以供参考，还给出了host.conf和nsswitch.conf文件的说明，从而配置各种名称服务的使用顺序。

第5章 高速缓存服务器和从服务器的配置 大多数Linux名称服务器都是高速缓存服务器或从服务器。本章就介绍了这两种服务器的配置。其中包括named.conf配置文件，根缓存文件和本地主机文件。

第6章 创建主服务器 每个域都有一个主服务器，存储域数据库信息。本章就介绍了主服务器的配置和DNS数据库文件的基本结构。其中涉及了域数据库和反转数据库，还解释了DNS数据库记录，称为资源记录。

第三部分：高级配置

这一部分讲述了超越基本知识的高级配置，给出了子域授权过程以及处理特殊情况配置特征。

第7章 建立子域 一旦用户拥有了一个域，就可以在这个域内划分子域。文章介绍了在域内进行子域授权的过程以及子域的维护。

第8章 特殊BIND配置 一般配置并不能满足所有站点的要求，第8章就针对这种情况介绍了转发器、首选服务器、性能调节、通配符、专用根目录等，同时也讨论了这些特性的利弊。

第9章 动态DNS 使用DHCP的站点需要为DHCP服务器找到一种方法，使之可以将主机名到IP地址的映射通知给DNS服务器。动态DNS（DDNS）就是完成这项任务的技术。该章讨论了DDNS的优势及其风险，同时给出了DDNS的配置方法和使用信息。

第四部分：系统维护

本部分集中讨论了维护服务器安全、可靠性的基本任务，包括安全性的维护、测试和监视等。

第10章 DNS安全 因特网上的每个系统都潜在着被入侵的危险，DNS服务器也不例外。本章讨论了通用的Linux安全程序以及用于保护系统的BIND特性。

第11章 测试DNS Linux系统的各种测试工具可以帮助用户测试自己的服务器或远程服务器。第11章描述了nslookup、dig和host等测试工具，给出了使用这些工具测试特殊问题的实例。

第12章 BIND日志文件 BIND软件创建了日志文件，使用户可以检查客户机和服务器之间的DNS协议实施情况。其中，包括BIND日志文件（named.run）、废弃文件（named_dump.db）、统计文件（named.stat），建立日志文件的配置选项也做了介绍。

附录

这本书包括四个附录。其中两个附录详细介绍了BIND的配置命令和BIND支持的资源记录。另外两个附录给出了BIND第九版和NIS的相关信息，这些与配置Linux DNS服务器并不直接相关，但对域管理员来说都是很有帮助的。

附录A BIND 9 附录A以BIND 9为例讨论了BIND软件的特性以及未来可能的变化。其中对BIND9的beta版功能也做了描述。

附录B named.conf命令参考 所有Linux BIND服务器配置都存放在named.conf文件中。这一部分给出了named.conf配置命令的实例，并提供了所有named.conf命令的详细语法参考。

附录C 资源记录参考 附录C提供了BIND支持的所有资源记录参考。

附录D 配置网络信息服务 NIS提供对统一维护的主机表进行访问的一项服务，如同对其他系统管理数据库的访问一样。附录D给出了这一可选服务的安装、配置信息。

目 录

第一部分 工作原理	1
第1章 DNS体系结构	1
/etc/hosts文件	1
DNS层次结构	5
域搜索	8
解析一个查询	9
缓存的重要性	11
获得域名	11
总结	13
第2章 DNS协议	14
Internet协议集	14
DNS消息	15
同步数据库	22
总结	24
第3章 BIND软件	25
解析器	25
名称服务器	27
浏览ISC Web站点	40
编译最新发布的BIND	42
安装BIND的其他方法	44
BIND配置的几种可能方案	49
建立DNS分级结构	50
总结	51
第二部分 基本配置	53
第4章 配置解析器	53
缺省配置	53
配置resolv.conf文件	54
解析器的环境变量	62
host.conf文件	65

nsswitch.conf文件	67
总结	70
第5章 高速缓存服务器和从服务器的配置	71
named.conf文件	71
根提示 (Root Hint) 文件	79
本地主机文件	82
总结	83
第6章 创建主服务器	84
named.conf文件	84
域数据库记录	85
区文件指令	89
区文件	93
反向域文件	102
总结	105
第三部分 高级配置	107
第7章 创建子域	107
为什么要授权子域	107
对域进行授权	111
给反向域授权	118
总结	123
第8章 特殊BIND配置	124
通知未知服务器	124
控制递归	126
性能调整	127
特殊的外部服务器	134
定义附加信息	138
使用通配符	144
总结	145
第9章 动态DNS	146
DDNS协议	146
计划、策略和问题	149
预配置DNS	152
激活动态更新	155

使用动态更新	156
总结	160
第四部分 系统维护	161
第10章 DNS安全	161
基本的安全措施	162
DNS服务器的保护	171
DNS的安全性协议	182
总结	192
第11章 DNS的测试	193
host命令	193
dig命令	199
nslookup	208
总结	217
第12章 BIND日志文件	218
转储文件	218
统计文件	224
跟踪文件	229
系统的记录消息	231
记录的配置	234
总结	237
第五部分 附录	239
附录A Bind 9	239
附录B Named.conf命令参考	243
附录C 资源记录参考	255
附录D 配置网络信息服务	268

第一部分 工作原理

第1章 DNS体系结构

在解释为什么需要将计算机名字翻译成数字地址的问题上，已经说得太多了。系统管理员们都知道其中的原因：用户习惯用名字，而网络使用数字。而系统管理员的工作就是让用户和网络都能够正常、愉快地工作！

在这一章，你将学习到名称如何翻译成使用IP协议的网络上的数字。本章解释了后续章节将要出现的一些术语，并且为理解为什么创建一个运行的域名服务器需要某些特定任务作了铺垫。

首先，我们会研究一下最初解决这一问题所使用的文件。正是由于这一文件的种种局限性才导致了今天域名系统（DNS）的分布式多层数据库系统结构的产生。我们将讨论域名的结构、它所表达的域的层次、服务器在层次中的位置以及根服务器的作用。有些读者初次接触DNS，我们还讲解了如何获得自己的域名。我们将从回顾过去的历史开始。

/etc/hosts文件

最初，主机表是将主机名映射为Internet地址所能使用的惟一工具。所有的Linux系统都有一个主机表，存放在/etc/hosts中。实际上，所有的IP网络服务器都有一个主机表。如果你使用过Unix，那么一定对/etc/hosts很熟悉，因为这个文件就是Unix系统用来存放主机表的。如果具有Windows NT背景，那么你一定知道主机表是%SystemRoot%\System32\Drivers\etc\hosts文件。所有这些系统上的hosts文件的结构和内容都是相同的。清单1.1是一个来自Linux系统的例子，它描述了hosts文件的结构。

清单1.1 /etc/hosts文件的结构

```
$ cat /etc/hosts
#
# Table of IP addresses and host names
#
127.0.0.1          localhost
172.16.5.5        crow
```

```

172.16.5.1          wren ns1
172.16.5.4          hawk gw5
172.16.5.20         kestrel kestral
172.16.5.2          robin redbreast bob
172.16.5.6          eagle www
172.16.5.7          bluebird blue news

```

以 (#) 开头的行为注释。其他所有行定义了主机表的数据，并且这些行的格式完全一样：数据行以IP地址开始，随后是一串映射到这些地址的名称。其中第一个名称是指定给那一地址的基本名称，其他名称则是别名或缩写。

主机别名则用来为某一主机名提供一些短小、历史性、通用的名称以及替代拼写。当然这些完全是为了方便用户使用。短名称的便利之处不言而喻。历史性名称则用来简化名称转化。当主机名改变后，旧的历史名称可能仍然被许多用户使用，或者嵌入在许多老的本脚中。通用名称则是这样一些名称，用户希望通过它们来定位某些服务。例如，**www**代表Web服务器，**mail**代表邮件服务器，**news**代表新闻服务器，用户一般希望通过这些名称找到相应的服务。当主机名很难拼写时替代拼写就帮上忙了。

提示：如果必须创建替代拼写或者经常需要更换主机名，那么你极有可能在一开始就选择了错误的主机名。主机名应当易于拼写并且不依赖于地点、用户或任务这样易变的事物。文档RFC 1178 “Naming Your Computer”中给出了关于主机命名的建议。

使用清单1.1中的**hosts**文件为例，用户可以指定名称**bluebird**、**blue**或**news**，系统都将返回IP地址172.16.5.7。从系统的角度看来，这些名称都是相同的，因为它们指向同一个IP地址。这样你可能会奇怪，为什么列表中的第一个名称被称为基本名称呢？基本名称是系统进行反向搜索，将数字地址转换为名称时使用的名称。将数字地址转换为名称可以让显示结果更具可读性。例如，**netstat**命令在确定网络状态时获取IP地址，但是缺省情况下，它只显示那些IP地址对应的主机名，如下所示：

```

$ netstat --inet
Active Internet connections (w/o servers)
Proto R-Q S-Q Local Address Foreign Address  State
tcp    1  0 robin:1967 eagle:80      CLOSE_WAIT
tcp    1  0 robin:1966 eagle:80      CLOSE_WAIT
tcp    1  0 robin:1964 eagle:80      CLOSE_WAIT
tcp    1  0 robin:1963 eagle:80      CLOSE_WAIT
tcp    0 126 robin:23    hawk:1449    ESTABLISHED

```

如本例所示，**netstat**命令在显示结果时使用了基本名称来代表每一个主机。**netstat -- inet**命令显示TCP/IP网络连接。每一行分别列出了主机以及连接涉及的端口。如果本地计算机只能访问一个主机表进行域名解析，那么代表每一个主机的名称就是主机表中与那个主机地址相关联的基本名称。

分析主机表

对清单1.1中的主机表进行逐行分析，就可以解释各种类型的主机表条目，包括那些你在自己的主机表中遇到的条目。前两行是包含在任何一个主机表中的：

```
127.0.0.1    localhost
172.16.5.5   crow
```

第一行定义了回送地址127.0.0.1，并将其分配给主机名localhost。回送地址是一个软件结构，它允许系统通过TCP/IP栈给自己发送数据，而不必真正在网络上传输数据。回送功能简化了测试，这样在两个本地进程之间传送数据时，系统可以使用用作网络通信的同一段代码，大大减少了网络上的通信量。

第二行定义了本地计算机的名称与地址。在本例中，本地计算机的名字为crow，分配的地址为172.16.5.5。每一台计算机的主机表中都有它自己的名称与地址。

接下来的两行是本地DNS服务器与本地缺省路由器的对应信息：

```
172.16.5.1   wren ns1
172.16.5.4   hawk gw5
```

这些条目将在系统引导时发挥作用。在本例中，wren是网络中的名称服务器，缩写为ns1，即“name server 1”的意思。hawk则是网络中的缺省网关，它的缩写是gw5，即“gateway 5”的意思。这些条目说明了缩写的用途，并且为读者描述了他们的主机表中可能出现的条目类型。

下面两行表示的条目类型可能用于捕捉拼写错误以及处理历史名称：

```
172.16.5.20  kestrel kestral
172.16.5.2   robin redbreast bob
```

第一行中，系统管理员增加了一个别名将打字错误kestral也认定为合法名称，因为用户很可能无法正确拼写kestrel。第二行包括了历史名字bob的一个别名，而bob则是本地网络的鸟类名字标准化以前指定给地址172.16.5.2的。这些行是由系统管理员添加的，以处理用户遇到的一些具体问题。你的主机表很可能不需要这些东西。

最后两行说明了通用名称的用法，例如www和news：

```
172.16.5.6   eagle www
172.16.5.7   bluebird blue news
```

一般说来，并不需要将类似这样的名字放到主机表中，因为，如果网络中确实有Web服务器或新闻服务器的话，那么当然会运行DNS服务。在DNS服务运行时，所有的计算机都可以从DNS服务器上获取这些名称，而不必从本地主机表中获得。

示例中的/etc/hosts文件包含八行，可能是你的计算机中hosts文件长度的两倍。在大多数计算机上，主机表的用途有限，因为系统依赖名称服务器进行名称—地址的解析。即便是规模较小、相对孤立的网络通常也依靠某种服务，例如网络信息服务（NIS，Network Information Service），来创建集中式的主机表，因为维护一个单独的服务器要比为系统中的每一个主机表维护一个单独副本容易得多。因此，即使主机表是解决名称解析的主要手段，那么也只需要一台服务器拥有非常庞大的主机表，而其他计算机拥有较小的主机表（参考一下附录D“配置网络信息服务”，其中描述了在Linux下如何配置NIS服务器）。

用于主机表-DNS转换的脚本

可以通过一些脚本将主机表转换为DNS数据库。使用这些脚本的一个问题是，它们假设你的系统拥有一个足够大的主机表以值得转换。就我个人而言，从1987年至今，我也未曾拥有一个主机表足够大的系统！这些脚本简单易用，但是在运行DNS的Linux系统上创建一个庞大的主机表毫无意义！第6章“创建主服务器”，阐明了将主机信息存入DNS数据库是多么简单。你可以放心大胆地忽略所阅读的任何关于主机表-DNS转换脚本的信息。

主机表用途

尽管用途有限，主机表还是非常重要。主机表用来在DNS不可用时，例如初始引导时，解析关键地址（例如缺省网关的地址）。缺省网关就是一个非常好的例子。名称服务器很有可能位于缺省网关的远端。那么，系统就无法与远程的名称服务器通信，直至加载缺省网关设置。在这种情况下，将缺省网关的地址放入本地主机表，就可以在名称服务器不可用时，不必查询名称服务器而配置网关。

除这些有限的功能之外，主机表在绝大多数系统中没有什么实际用途。所有访问Internet的系统都要使用DNS进行名称—地址的解析。当然，也有可能你的系统位于一个孤立的小网络上，既没有接入Internet，也没有必要与远程系统通信。在这种情况下，主机表就能够满足所有的需求了，尤其与NIS提供的集中维护功能结合起来。但是这是一个非常罕见的例子，对于其他绝大多数情况，主机表的任务就是在引导时对系统提供支持以及处理一些本地别名。其他的一切任务都由DNS完成。

主机表局限性

讨论到现在，我们还没有给出一个采用DNS的充足理由。毕竟，每一个系统都已经拥有并且在使用主机表，来完成连接到IP网络的系统所需要的名称—地址和地址—名称的映射。实际上，主机表最初是Internet雏形阶段能够提供名称服务的唯一的工具。那么为什么说，每一个连接到Internet的系统都只需要一个小的主机表而且应当主要依赖于DNS呢？主机表的显著缺陷就是，它缺少今天Internet所需要的规模扩展能力。

主机表是平面文件 主机表是一个简单的文本文件，容易编辑但难于查找。对于每一个主机—地址映射，都需要对该文件顺序查找，文件的每一行中的主机名都需要查找一遍。这对于清单1.1中给出的主机表来说没什么问题，用户个人的Linux系统中的此文件可能也不大。但是，即便拿出当今Internet的名字集的一小部分来，检索这样一个平面文件的任务也是难于承担的。

主机表需要集中维护 当新的主机加入Internet时，它们必须向Internet网络信息中心（NIC）注册。NIC将它们加入HOSTS.TXT文件，该文件是用来生成主机表的源文件。世界上任何一个组织会将其每一台接入Internet的计算机名输入到此文件中，真是难以想像！如果所构建网络中的每一台计算机都需要向政府机构注册，那么构建这样一个网络需要耗费多长时间！如果主机表仍然是将名字翻译为地址的唯一方式，今天大家所熟知的Internet恐怕不会存在了。

主机表没有自动分发机制 因为Internet主机表是集中维护的，就需要某种机制将主机表从NIC移植到Internet上的每一台计算机。Unix系统有几个命令可以用来下载表单并将其转换成正确的格式。然而，至于主机表是否已经更新或何时更新却无从得知，当然也无法获知何时需要下载更新版本。

每个系统都需要自己的主机表拷贝 没有一个系统能够使用主机表来解析名字，除非那个名字就位于系统的本地主机表中。即便是一个系统被设计成能够自动分发主机表，将主机表分发至Internet上的每一台主机也面临着难以超越的困难。

鉴于上述局限性，开发一种解决主机名与IP地址之间映射问题的新方法，对于Internet的发展是非常重要的。在20世纪80年代晚期，出现了DNS，为发展中的Internet提供名称服务。

DNS层次结构

DNS解决了主机表系统固有的所有问题。它不是平面结构，而是一个分布式、分层的数据库系统。DNS仅需要很少的集中维护。它有多种方法将数据库信息自动分发至需要这些信息的服务器上。本地系统无需维护数据库的完整拷贝，而仅仅在需要时了解一下就可以了。这些优势来自域数据库系统的分布式特性、将其链接起来的DNS层次结构以及定位正确的主机—地址映射的方式。

域名结构

DNS是一个有根的分层结构。Linux系统管理员肯定非常熟悉这个结构，因为它与Linux文件系统结构相同。一个域名就显示了DNS的有根分层结构，正如一个Linux文件的绝对路径名显示了目录结构一样。一个Linux绝对路径名从根开始，经过一系列目录，最后以文件名结束，例如/usr/lib/zsh/zed。绝对路径名的各个部分以斜杠分隔。根由一个空的名字——单独的斜杠 (/) 后面不带任何名字来表示。文件的路径名是从最概括（根）到最具体（文件）。

完全合格域名（FQDN）有几点很相似。域名的每一部分都由一个字符隔开，在域名中该字符是一个圆点。根由一个空的名字——圆点（.）不带任何后缀名字来表示。但在这里，顺序正好相反。域名是从最具体（主机）到最概括（根）。例如，在域名royal.terns.foobirds.org中：

- royal为主机名
- terns为子域
- foobirds是该组织所购买的二级域名
- org是该组织注册其二级域名的顶级域
- .为根

这一结构的一个重要特征是，它使得为Internet规模的网络创建足够多的惟一主机名成为可能。同样地，Linux文件系统可以让两个文件同名，只要其绝对路径名惟一就可以了。DNS允许很多计算机拥有相同的主机名，只要完全合格域名各不相同就行了。正如，/home/craig/bashrc与/home/sara/bashrc是两个不同的文件，cow.mammals.org与cow.dairies.com是不同的主机。只有从根域到各个主机的完全路径才惟一界定了主机名称。