网络应用系列

网络服务器的建立、调试与管理

李庆明　闫　斌　王春年　张红艳　孙晓鹏　编著

科学出版社

1998

内 容 简 介

本书简要介绍建立、调试和管理网络服务器的基本知识，并对域名系统、WWW服务器、E-Mail服务器、FTP、TELNET、网关、防火墙等实用技术作了详细阐述。

本书可作为初次建立网络服务器的技术指南和培训教材，也可作为从事网络管理和开发的广大技术人员的参考用书。

图书在版编目(CIP)数据

网络服务器的建立、调试与管理/李庆明等编著. -北京：科学出版社，1998.7
（网络应用系列）
ISBN 7-03-006505-0

Ⅰ.Ⅰ… Ⅱ.李… Ⅲ.因特网-服务器 Ⅳ.TP393.4

中国版本图书馆CIP数据核字（98）第01136号

科学出版社出版
北京东黄城根北街16号
邮政编码：100717
北京双青印刷厂印刷
新华书店北京发行所发行 各地新华书店经售
*
1998年7月第 一 版 开本：787×1092 1/16
1998年7月第一次印刷 印张：10 3/4
印数：1—5 000 字数：246 000

定价：17.00元

目　录

1 域名系统

域名系统的建立是接入Internet的一项基础性工作。本章帮助读者理解域名系统的概念，它与IP地址的关系，掌握名字服务器的具体配置，以及DNS的维护和故障排除等方法。

1.1 域名系统

域名系统(Domain Name System)的建立是一项基础性工作，是一项特殊的服务，用于解决计算机名字到 IP 地址的映射。它的实现给用户带来了极大的便利：用户不需再记忆单调、枯燥的数字，不必了解 IP 地址等技术细节，就可以方便地使用 WWW，FTP，E-MAIL，NEWS 等各种信息服务。

1.1.1 域名系统的历史

Internet 起源于 ARPAnet。在 ARPANET 上，主机名字到地址映射最初是由一个单独的文件 HOST.TXT 实现的。70 年代，ARPAnet 仅是一个由数百台主机组成的小规模的网络，一个单独的 HOST.TXT 文件包含了所有连接到 ARPAnet 上每台主机的名字到地址的映射，它由 SRI(the Standford Research Institute) 网络信息中心(NIC)负责维护和向外散发。管理人员将他们的改动通过电子邮件发给 NIC，并定期获取最新的 HOSTS.TXT。SRI- NIC 的工作人员定期将改动编入 HOSTS.TXT。

当网络规模较小时，它是一种简单实用的方法。但是，当网络不断扩展，特别是 ARPAnet 采用 TCP/IP 协议后网络规模爆炸性增长时，这种线性表结构、集中式管理的 HOSTS.TXT 机制遇到了一些无法解决的问题，如：

·巨大的通信量

每增加一台主机就需要在 HOSTS.TXT 文件中增加一行，HOSTS.TXT 文件的大小随主机数的增加而增大，网络的迅速膨胀使 HOSTS.TXT 成为一个巨大的文件；同时所有其他主机都要从 SRI-NIC 中获取更新过的 HOSTS.TXT 文件，这种由更新过程所带来的通信量增长得更快。这种集中式的管理使 SRI-NIC 在网络通信量和处理负载上的开销无法接受。

·名字冲突

在 HOSTS.TXT 中不能有两台相同名字的主机，否则，会引起混乱。在增加一台主机之前必须先获取最新的 HOSTS.TXT 文件，查看该名字是否已经存在。但由于在多个网络中会同时增加多台主机，所以这种线性表机制使名字冲突成为一个无法避免的问题。

·一致性

在一个正在扩充之中的网络上保持文件的一致性越来越困难，甚至于不可能。当最新的 HOSTS.TXT 文件传到一台主机时，可能网络中的另一台主机又改变了地址或者又出现了一

台新主机。定期改变 HOSTS.TXT 这种静态管理机制无法保持文件的一致。

为解决上述问题，USC(University of Southern California)的信息科学研究所的 Paul Mockapetris 设计了新的系统结构。在 1984 年他发布了 RFC 882 和 RFC 883(现已由 RFC 1034 和 RFC 1035 代替)，定义了域名系统(DNS)，为名字系统提供了完善的解决方案。它具有层次式、分布式、动态变化等特点，很好地解决了 HOSTS.TXT 所无法解决的问题：它允许本地数据管理员使其数据具有全局可访问性，这种管理的分散化可消除单台主机所造成瓶颈以及减轻通信拥塞，并且本地管理员可容易地实现使数据保持最新；它采用层次化结构解决了主机名字冲突的问题。

1.1.2 域名系统

域名系统的第一个实现称为 JEEVES，它是由 Paul Mockapertis 编写的。BIND 为 Berkeley Internet Name Domain 的简称，是 DNS 实现中最流行的一个。它是由 U. C. Berkeley 的 Kevin J Dunlap 为 Berkeley UNIX 操作系统所写的，现已成为绝大多数商用 UNIX 中的标准部分。

域名系统是一个分布式数据库。它允许本地负责控制整个数据库的部分段，每一段中的数据通过客户-服务器模式在整个网络上进行存取。复制技术和缓存技术的采用，不但保证整个数据库的安全，又使系统拥有了优良的性能。

1.DNS 数据库结构

下面首先介绍关于 DNS 数据库结构的一些概念：域名空间、域、域名、资源记录等。DNS 分布式数据库中的数据单元是按名字进行索引的，这些名字形成了一种倒转的树状结构，该结构称为域名空间。该树的层次结构，如图 1.1 所示。该树在顶端有唯一的根，DNS 简单地称其为“根”(Root) 或“根域”(Root Domain)。在这种树状结构中，每一个节点都表示整个数据库中的一个分区，该分区称之为域。每个域可再进一步地划分成子分区，即子域，该树层次的限制为 127 层。每个域由标签标明了它同其父域的关系，由域名给出了它在数据库中位置。与域名相关的数据则存储在资源记录中。

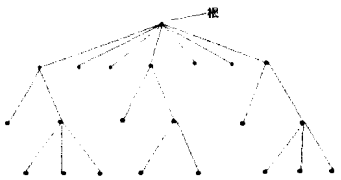

图 1.1 域名空间结构

(1)域名

树中每一节点用一个简单的字符串作标签。该标签最多可包含 63 个字符(不可以是“.”)，为便于记忆，一般不超过 12 个字符。根域为一空(零长度)标签。树中节点到根之间路径上的标签序列构成了域名(Domain Names)。其读取顺序从节点到根，用点号分隔路径中的标签，但点号并不总是标志一个区的划分。为了方便，根域一般写为点“.”，如果它在节点的域名的结尾出现，那么这种域名就被称为绝对域名(absolute domain name)，也被称为全称域名(fully-qualified domain name)，简称为 FQDN。不以点结尾的名字一般解释为相对域名。

为保证域名唯一地标识树中每一个节点，DNS 要求兄弟节点(同一父节点的子节点)的命

名具有唯一性。

(2)域

域(domain)是树状域名空间中的一棵子树。域的名字就是该域中最高层节点的名字。如sxu.edu.cn域的顶端即为sxu.edu.cn的节点。

子域中的任何域名均被认为是域的一部分。由于一个节点可以包含在很多子树中,所以一个域名可以存在于很多域中。例如,域名sxu.edu.cn为域sxu.edu.cn的一部分,它也是cn域的一部分,如图1.2所示。

域名空间中,叶节点的域通常代表主机。它们的域名可以指向网络地址、硬件信息和邮件路由信息。树内的节点,其域名既可命名一台主机,又可指向有关域的子孙或子域的结构信息。域名树中的内部域名并不受唯一性限制。它们既可表示它们所对应的域,又可代表网络中某台特定的主机。例如,sxu.edu.cn既为SXU的域,又为SXU和Internet间转发邮件的主机的域名。

使用主机还是结构,依据域名所使用的环境来决定。查询一个节点子孙的检索操作将返回结构数据,而传送邮件到一个域名将检索主机信息。

一个域是否为另一域的子域?简单的判断方法就是比较它们的域名,子域名一定是以其父域名结尾的。例如,域sxu.edu.cn肯定为edu.cn的子域,因为sxu.edu.cn以edu.cn结尾。同样,edu.cn是cn的子域。

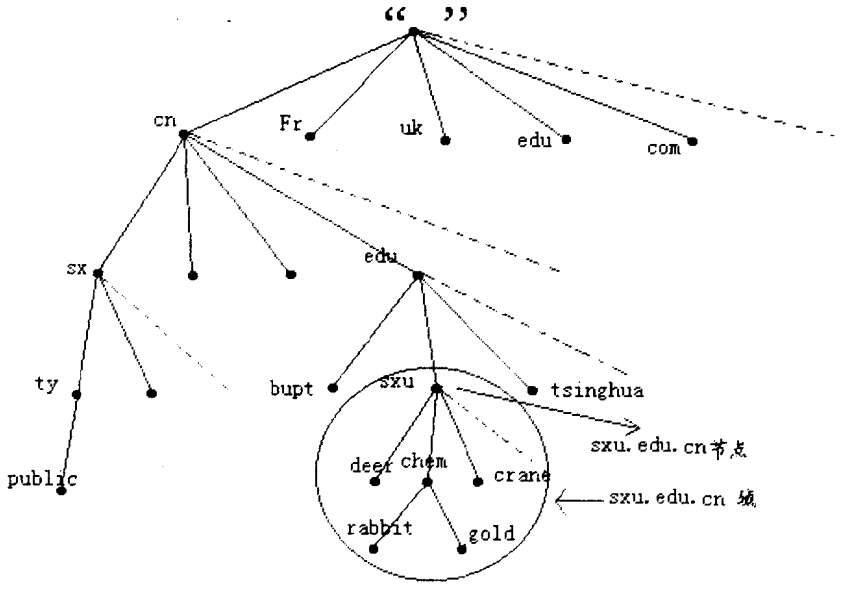

图1.2 sxu.edu.cn域

(3)资源记录

与域名相关的数据存储在资源记录(resource record)中,简称RR。记录划分成不同的类,每类记录都相关于某一类网络或软件。现在共有三类,用于Internet(任何基于TCP/IP的互联网)的IN,以及基于Chaosnet协议的网络的CH和用于Hessiod网络的HS,其中IN是最流行的。

记录分为不同类型。每种类型都对应于一个存储在域名空间中的不同的数据变量。不同的类型还定义了一些不同的记录类型,每一种记录类型都定义了一种特殊记录句法,该句法是

所有该类型的资源记录都必须遵守的。

(4)Internet 高层域

域名空间对域名标签没有任何特殊规定,可以自己决定域名的语义规则。但是对于 Internet,其域名空间有一些约定俗成的要求,特别是在高层域上,域名遵从某些特定传统,这样可使域名不致陷入混乱。

最初的高层域按组织形式划分了 Internet。有七种主要的高层域,如下所示:

com	商业组织	edu	教育组织
gov	政府组织	mil	军事组织
net	网络组织	org	非商业组织
int	国际组织		

其中 com,org,net 为国际通用顶级域名。随着 Internet 在全球的迅猛发展,原有的域已不能满足需要,因此又增加了如下七种通用顶级域名:

firm	商业公司	arts	强调文化和娱乐的团体
nom	用于个人和个体	web	与 WWW 特别相关的实体
rec	强调消遣和娱乐的实体	info	提供信息服务的组织
store	从事商业销售的企业		

DNS 同时也支持地理位置的划分。新的高层域对应各个国家的组织,他们的域名遵从国际标准 ISO 3166。ISO 3166 为世界上的每一国家建立了官方的、两字母的缩写,如 cn 代表中国、ca 代表加拿大、uk 代表英国等等。

2.DNS 工作机制

了解了 DNS 数据库的结构之后,下面介绍一些有关 DNS 工作机制的概念和术语。DNS 采用客户机/服务器模式,利用解析的方式获取相关数据;通过代理区,实现了数据库的分散化管理,减少了不必要的数据;通过复制和缓存,保持了冗余,提高了系统性能。

DNS 的客户机/服务器机制中,服务器部分称为名字服务器(name server),客户方称为解析器(resolver)。名字服务器具有授权区(zone)的完整信息,负责从域名空间中检索数据,并将答案返回解析器。解析器查询名字服务器、解析响应、返回信息给申请程序。解析器发出查询到最后获得相关数据的过程称为解析(resolution)。区与域是两个不同的概念,区中不包括代理(delegation)出去的域。为保持冗余、分担负载,需要建立多个名字服务器,名字服务器的数据采用复制的方法——区传输(zone transfer)进行更新。缓存(cache)的采用缩短了解析过程,减轻了根名字服务器的负载。

(1)名字服务器

名字服务器(Name Server)通常具有区(zone)的完整信息,对该区具有权限(authority)。名字服务器可同时具有多个区的授权。

区和域是两个不同的概念。区包括了域中除代理出去的子域外,所有的域名和数据。然而,如果域的子域没有被代理出去,域和区就包含同样的数据。名字服务器装载区而不是域,是因为域中可能含有对该名字服务器不必要的信息——已经代理出去的域。

设计域名系统的一个主要目的就是让管理分散化。通过代理(delegation)实现了这一目的。管理域的组织将域划分成子域,每一子域由其他组织代理,即将子域的权限分配给不同的名字服务器。用户的数据包括了指向具有子域权限的名字服务器的指针,而不再是子域的具

体数据。如果名字服务器被询问有关该子域的信息,它会应答一个可与之会话的相应名字服务器的列表。例如:sxu.edu.cn 被代理给了山西大学负责运营校园网的人员,如果要访问 SXU 域中的数据,它会指向 SXU 中的名字服务器。

DNS 定义了两种类型的名字服务器:主名字服务器和辅名字服务器。主名字服务器从它所运行的主机上的文件获得它负责的区的数据。辅名字服务器从其他的具有该区权限的名字服务器上获得它的区数据。当辅名字服务器启动时,它同相关的名字服务联系并传输区数据,这一过程被称为区传输(zone transfer)。

DNS 提供这两种类型的名字服务器是为了保持冗余、方便管理。在创建好区数据并设置好了主名字服务器后,就不必将这些数据从一台主机拷贝到另一台主机,只需简单地设置从主名字服务器装载区数据的辅名字服务器就可以了。一旦它们已被安装好,辅名字服务器就会定期的查询主名字服务器来保持区数据为最新版本。同时,当主名字服务器发生故障时,辅名字服务器可以立即提供名字服务,保证了系统的可靠性。

主名字服务器用来装载其区数据的文件被称为数据文件(data file)。可将辅名字服务器配置为在传输区数据文件的同时进行备份。如果辅名字服务器被中断后又重新启动,它首先读取备份的数据文件,然后检查该数据文件中的数据是否最新。数据文件包含描述区的资源记录。资源记录定义了区中的所有主机,并标明每一子域的代理。

(2) 解析

解析器(resolver)是访问名字服务器的客户。它所做的工作为:汇集查询,发送查询并等待应答,未得到应答时重发查询。由于解析器的功能十分简单,名字服务器不但要给出有关它们负责的区的数据,而且还要能够搜索域名空间来找到它们没有权限的区的数据,这一过程称为名字解析(name resolution),简称为解析(resolution)。

名字空间的结构是通过名字进行索引的一棵倒转的树,所以由根开始,给出完整的名字索引就能使名字服务器找到树中任何一个节点。先查找根名字服务器的名字和地址,然后由根名字服务器启动在该名字路径上的名字服务器。

根名字服务器知道所有高层域的名字服务器的位置。一旦收到域名查询后,根名字服务器就提供具有该域名所在高层域的权限范围之内的名字服务器的地址和名字。再由高层域提供该域名所在的第二层域的权限范围之内的名字服务器列表。这样,每一次名字服务器的查询不是得到如何更靠近答案的信息就是得到答案本身。

显然,根名字服务器在解析过程中起着非常关键的作用。尽管 DNS 提供缓存机制,减轻了根名字服务器的负载,但在缺乏其他信息的情况下,解析还是只能从根名字服务器开始,这使得根名字服务器成为 DNS 操作的核心。为了不致因为根名字服务器的失败而使所有解析过程崩溃,也为了分担负载,Internet 上有多台根名字服务器在运行,跨越了整个网络的不同部分。但是作为查询的汇集点,每台根名字服务器上的信息流量仍非常大。

查询有两种方式:递归(recursion)和重复(iteration)(或简称非递归)。递归查询将绝大部分工作放在单一的名字服务器上,递归是指当名字服务器收到递归查询时所使用的解析过程,重复是指当名字服务器收到重复查询时所使用的解析过程。

在递归中,解析器将有关特定域名信息的递归查询发送给名字服务器。这时,被查询的名字服务器只有以所查询的数据结果、指出所要查询的数据类型不存在的错误信息或者要查询的域名不存在的错误信息三种情况来回答。因为查询是递归类型的,名字服务器不能传递回别的名字服务器地址。

如果被查询的名字服务器不具备所要查询数据的权限，这时，它将必然查询其他的名字服务器来获取答案。它可以发送递归查询给那些名字服务器，从而迫使它们找到答案并将答案返回。或者，它发送重复查询，然后可能返回别的更靠近要寻找的域名的名字服务器的地址。

在重复解析中，名字服务器仅将它所知道的最好答案返回给查询者，不作额外的查询。被查询的名字服务器检索其本地数据(包括它的缓存中的数据)来找寻所要求的数据。如果找不到所需的数据，它将尽力返回给查询者可以帮助查询者继续解析过程的答案。

解析器查询本地名字服务器，然后本地名字服务器查询一组别的名字服务器来获取答案。它所查询的每一名字服务器都传递回沿着名字空间更靠近所要数据的别的名字服务器地址，最终本地名字服务器查询到具有权限的名字服务器，从而得到所需数据。

地址到名字的映射是用来生成便于人来阅读和解释的输出(例如在登录文件中)。在DNS中，域名空间中的数据(包括地址)，是按名字进行索引的。找到一给定域名的地址相对容易，但是找到映射到给定地址的名字则就要在树中的每一个域名空间中进行穷尽搜索。

in-addr.arpa以地址做为索引的域名空间，给出了解决地址到名字的映射的一个清晰有效的办法。in-addr.arpa域中的节点以Doted-ocet(将32 bit IP地址表示为由点"."分隔开的4个8位的十进制数字形式的方法)形式表示IP地址。例如in-addr.arpa域可以有256个子域，对应于第二个字节的可能值。最后，在第四层内，有一链接在最后一字节上的资源记录给出了具有该IP地址的主机或网络的全称。这样就生成了一个巨大的域——in-addr.arpa域，容纳了Internet中的每一个IP地址，见图1.3。

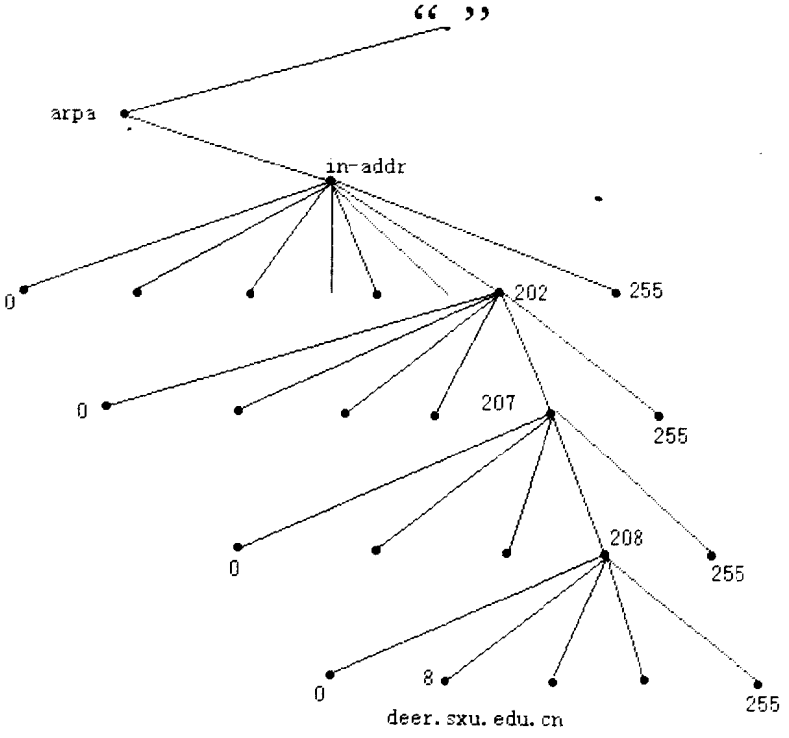

图1.3 in-addr.arpa域

IP地址在域名空间中以相反的方向表示，第一个IP地址字节在in-addr.arpa的底部。这样，域名中的IP地址就可以被正确地读取。

然而，IP地址同域名一样是分层的。网络编号用点隔开的方式很像域名，网络管理员就

可以据此划分地址空间,进一步代理出编号空间。IP 地址同域名区别在于 IP 地址由低层到高层是自左至右,如图 1.4 所示。

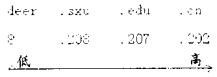

图 1.4 IP 地址与域名的关系

使 IP 地址中的第一个字节出现在树的最高层使管理员有能力沿网络连接将 in-addr.arpa 域的权限代理出去。例如,202.in-addr.arpa 可以被代理给网络 202 的管理员。若该字节以反方向出现,那么这样做是不可能的。如果 IP 地址以相反方向表示,则 202.in-addr.arpa 将由所有以 IP 地址 202 结尾的主机组成——这就不是一个可代理出去的域。

in-addr.arpa 名字空间显然对由 IP 地址到域名的映射有用。在域名空间中搜寻一个用非地址形式数据索引的域名,除了使用像 in-addr.arpa 这样特殊名字空间外,就只能使用穷尽搜索,即反向查询(inverse queries)。反向查询是搜寻一个用给定数据索引的域名。它由收到该查询的名字服务器单独进行处理。

该名字服务器搜索其所有的本地数据来查询该数据项,如果找到则返回该索引相关域名。如果它找不到数据,就放弃。该名字服务器并不会再去查询别的名字服务器。

(3)缓存

解析过程与主机表相比较,显得十分费解和庸肿。但在实际应用中,解析过程是很快的。其原因就在于采用了缓存(caching)机制。它一方面使解析加速,另一方面使本地名字服务器不必过分依赖根名字服务器,减轻了根名字服务器的负载。

名字服务器在处理递归查询时,要进行多次查询才能找到答案。但是在这一过程中,它可以获得很多有关域名空间的信息。名字服务器在每次获得新的名字服务器列表时,都获得了那些名字服务器所具有权限的区的信息和地址。然后,在名字服务器最终找到最初查询所需的数据后,它可将这些数据保存起来以备将来使用。下次若名字服务器存有与所收到的解析器的所查询的域名相关的信息,则整个查询过程就会缩短。名字服务器可能也将答案缓存起来,这时它只将答案返回给解析器就可以了,即使未将答案缓存,它也可能知道哪些名字服务器具有所查询域名的权限,从而直接查询它们就可以了。

假设我们的名字服务器已经找到了 www.microsoft.com 的地址。在这一过程中,它将 www.microsoft.com 和 microsoft.com 的名字和地址缓存了起来。当我们要查询 ftp.microsoft.com 的地址时,我们的名字服务器就不必查询根名字服务器,直接由 microsoft.com 名字服务器开始查询。

但是,名字服务器不能、也不应该永远地缓存数据。这样会使对授权名字服务器上数据的改变永远无法传递到网络中的其他部分,远程的名字服务器就只能使用缓存的数据。因而,区管理员需要确定该数据的生存期(TTL),即名字服务器对数据缓存的允许时间长度。超过生存期,名字服务器就丢弃缓存数据并从授权名字服务器中进行区传输,重新获得新的数据。

数据的生存期的确定必须在性能和一致性间进行权衡。短的 TTL 可以有利于确保你的域数据在整个网络上的一致性,但另一方面,它将增加名字服务器的负载并延长有关域中信息的平均解析时间。长的 TTL 会缩短有关你的域中信息的平均解析时间,但数据信息具有不一致性的周期也变长了。

1.2 建立域名服务器

1.2.1 准备工作

在我们对域名系统的理论有了初步了解之后,我们开始建立名字服务器。首先你需要做以下几件准备工作:申请 IP 地址,获取 BIND 软件包,确定域名等。这里,我们假定你已经申请到 IP 地址,对 IP 地址的申请就不再介绍。

1.获取 BIND

BIND 为 Berkeley Internet Name Domain 的简称,是 DNS 实现中最流行的一个。它通常是 UNIX 操作系统的一部分,但也有例外情况。另外,如果你想使用最新的、功能最完善的版本,你可以自己获取程序的源代码。目前最新的版本是 BIND 4.8.3,它是自由发布的。你可以通过匿名 ftp 从 ftp.uu.net 中取得文件/networking/ip/dns/bin/bind.4.8.3.tar.Z。编译的难易取决于你的 UNIX 的版本与 BSD UNIX 的相近程度,越相近编译就越容易。SunOS 是基于 BSD 的一种 UNIX,所以,编译相对容易。关于如何编译不同 UNIX 的 BIND,你可以通过邮件列表 bind@vangogh.sc.berkeley.edu 和 namedropers@nic.ddn.mil 来获得进一步的帮助。

2.选择域名

选择域名并不像想像中那样简单。因为它不仅需要选择一个域名,还需找到相应的父域。换句话说,你需在 Internet 域名空间中找到相应位置,并占据域名空间的一角。选择域名的第一步是确定你所属的域名位于域名空间的何处。最容易的方法就是从顶层开始,向下查找:首先找到你属于哪一个高层域,然后找到相应的子域。

如果你在美国,顶层域的选择只与你拥有主机的多少有关:如果仅有一台或两台主机,或者你只想在 Internet 域名空间中注册几台主机,你可以加入 us 域。us 域下的第二层子层是相应的州名的缩写,第三层子域以城市名命名。所以,如果你所需要的仅是将在 Colorado Sprins 的两台主机注册到因特网上,你仅需将它们加到 colospgs.co.us 域。

但是,在 us 域下,并不代理子域。这一方面是由于 us 域的结构,另一方面是由于 us 域的管理策略。你的主机成为已经存在的 Internet 域的一部分,us 域管理员会为你的主机增加相应的通信地址和 mail 地址,但是他们并不将子域代理给你。一方面他们减轻了你管理你主机域名服务器的负担,但同时你也不能创建子域。

如果你想创建自己的子域,你不得不申请一个顶层域的子域,如 edu 和 com。只要你不是要求一个特别长的域名(不超过 12 个字符),或该域名已经存在,你总是可以得到你想要的。

目前,我国 cn 高层域分为两类,共有 40 个。其中类别域名 6 个,即 ac:适用于科研机构;com:适用于工、商、金融等企业;edu:适用于教育机构;gov:适用于政府部门;net:适用于互联网络,接入网络的信息中心(NIC)和运行中心(NOC);org:适用于各种非盈利性的组织。

行政区域名 34 个,适用于我国的各省、自治区、直辖市,主要以省市名称的汉语拼音的缩写构成,如 bj 代表北京,sh 代表上海,sx 代表山西等,详见表 1.1。

表 1.1 省、自治区、直辖市字母代码表

拼音全名	区域代码	省市名称	拼音全名	区域代码	省市名称
Anhui	AH	安徽	Jiangxi	JX	江西
Beijing	BJ	北京	Liaoning	LN	辽宁
Chongqing	CQ	重庆	Macau	MO	澳门
Fujian	FJ	福建	Neimenggu	NM	内蒙古
Guangdong	GD	广东	Ningxia	NX	宁夏
Ganshu	GS	甘肃	Qinghai	QH	青海
Guangxi	GX	广西	Sichuan	SC	四川
Guizhou	GZ	贵州	Shandong	SD	山东
Henan	HA	河南	Shanghai	SH	上海
Hubei	HB	湖北	Shanxi	SN	陕西
Hebei	HE	河北	Shanxi	SX	山西
Hainan	HI	海南	Tianjin	TJ	天津
Hongkong	HK	香港	Taiwan	TW	台湾
Heilongjiang	HL	黑龙江	Xinjiang	XJ	新疆
Hunan	HN	湖南	Xizang	XZ	西藏
Jinlin	JL	吉林	Yunnan	YN	云南
Jiangsu	JS	江苏	Zhejiang	ZJ	浙江

中国互联网络信息中心(China Internet Network Center:CNNIC) 是 1997 年 6 月 3 日由国务院信息化领导小组办公室授权中国科学院计算机网络信息中心组织建立的非赢利的服务管理机构。它负责中国的域名管理,为中国互联网络用户提供域名注册服务,并代表中国各互联单位与国际互联网络信息中心、亚太互联网络信息中心及其他的互联网络信息中心进行双边业务联系。中国互联网络信息中心 CNNIC 工作委员会制定了《中国互联网络域名注册暂行管理办法》和《中国互联网络域名注册实施细则》,规定了我国互联网络的域名管理机构、域名体系结构以及对申请注册域名单位的要求,对三级域名及使用的规定,域名注册的审批程序等。

1997 年 10 月 7 日,中国互联网络信息 CNNIC 正式获得受理新增 7 个国际通用顶级域名注册申请的权利,国内的机构欲注册顶级域名可在 CNNIC 就近办理,出现域名争端在国内就可解决。这对我国互联网络的发展,具有十分重要的意义。CNNIC 负责除 EDU.CN 域之外所有域名的申请,而 CERNIC (中国教育网络信息中心)则负责二级域 EDU.CN 下所有域名的申请。如果你想了解有关详细情况,可以访问 CNNIC 的主页:http://www.cnnic.net.edu 和 CERNIC 的主页:http://www.cernic.edu.cn。

1.2.2 建立名字服务器

BIND 可被配置成几种不同的运行方式。DNS 采用的是客户机-服务器机制。在服务器方,BIND 可被配置为主服务器 (Primary Server)、辅服务器 (Secondary Server)和纯缓存服务器(Caching-only Server) 和转发服务器(Forward Server)等;而客户方则被配置成解析器

(Resolver)。

1. 域名服务器的配置

一般来说,建立域名系统至少需要建立两台名字服务器:一台主名字服务器 PM 和一台辅名字服务器 SM。下面我们将结合山西大学的实例,就主、辅名字服务器的建立作详细介绍(山西大学已向教育科研网络信息中心(CERNIC)申请了域名 sxu.edu.cn)。

一个区由主域名服务器管理,它维护该区的所有域名空间信息。一个区至少要有两个主域名服务器,PM 服务器和 SM 服务器,它们位于不同的主机上,当 PM 服务器关闭、损坏或过载时,SM 服务器能做为备份服务器工作。它们之间的区别在于:PM 服务器从其硬盘上装入其管理区内域名数据,并对本区内其他域名服务器授权;SM 服务器从主域名服务器获得授权,从主域名服务器取得其管理区内域名数据,并周期性地与主域名服务器上的数据相比较,更新其管理区内域名数据。

配置服务器需要有如下配置文件:启动文件 named.boot 和数据库文件 db file。其中 db file 是如下文件的集合:(1) 用于映射主机名到地址的正规域(regular domain)的区文件(db.DOMAIN);(2) 将地址映射成主机名的反向域(reverse domain)的区文件(db.ADDR);(3) 指向根域服务器的缓存文件(db.cache);(4)本地解析自反地址的自反(Loopback) 文件(db.127.0.0)。

下面我们将逐一仔细讨论用于配置 named 的这些文件。

(1) named.boot 文件

named.boot 文件说明域名服务器的类型、授权管理的区,以及从什么地方读取初始化数据。此文件位于/etc 目录中。当/etc/named.boot 存在时,系统在启动时,将启动域名服务器进程,它首先读取 named.boot 文件对其自身进行初始化。named.boot 文件主要有如下配置命令:

①directory directory-name:说明域名服务器的工作目录为 directory-name。

②primary zone-name data.zone:说明域名服务器是第一主域名服务器,授权管理的区为 zone-name,本管理区初始化地址映射文件是 data.zone

③secondary zone-name IP-address1...IP-address data.bak:说明域名服务 器是 SM 服务器,授权管理的权限为 zone-name, 本管理区地址映射数据来自于地址是 IP-address1,...IP-addressn 的主机上的域名服务器; data.bak 是用来储存从主服务器接收信息的本地文件的名字。

④cache.root.cache:说明根域名服务器的信息来自于 root.cache 文件。

⑤forwarders IP-address1,...,IP-addressn:说明它是转发服务器,它把所有的映射请求转发给地址是 IP-address1,...,IP-addressn 的域名服务器处理。

配置 named.boot 文件的方式决定了名字服务器是否作为主服务器还是一台辅服务器。下面的 named.boot 文件定义了 deer 为 sxu.edu.cn 域的 PM 名字服务器。

```
;BIND data file to boot a primary name server for sxu.edu.cn Domain.
directory     /var/named
;type        domain                    source host/file
primary       sxu.edu.cn              db.sxu
primary       208.207.202.in-addr.arpa    db.202.207.208
```

```
primary      0.0.127.in-addr.arpa      db.127.0.0
cache        .                         db.cache
```

Directory 语句为后续语句节省了录入工作量，它指定了域 sxu.edu.cn 的工作目录为/var/named。

第一条 primary 语句声明这是用于 sxu.edu.cn 域的主名字服务器，用于该域的数据从 db.sxu 文件中装载。

第二条 primary 语句指向映射 IP 地址 202.207.208 到主机名的文件，该语句说明本地服务器是反向域 208.207.202.in-addr.arpa 的主服务器，并且用于该域的数据要从 db.202.207.208 中装载。

第三条 primary 语句将本地服务器定义为其自反域的主名字服务器，并且说明用于该域的信息存放在 db.127.0.0 文件中，自反域为 in-addr.arpa，它映射地址 127.0.0.1 到本地主机。

cache 语句告诉 named 来维护一个名字服务器响应的缓存，并用文件 db.cache 中的内容来初始化缓存。cache 语句几乎用于每一种名字服务器配置当中。

为了便于阅读，库文件中常常含有注释和空行。其中“;”标志着该行为注释行。

SM 服务器的配置同主名字服务器的区别仅在于用 secondary 代替 primary 语句。Secondary 语句指定远程服务器而不是本地磁盘文件做为域信息源。

```
;BIND data file to boot a secondary name server for sxu.edu.cn Domain.
directory   /var/named
;type       domain                          address          source host/file
secondary   sxu.edu.cn                      202.207.208.8    db.sxu.bak
secondary   208.207.202.in-addr.arpa 202.207.208.8           db.202.207.208.bak
primary     0.0.127.in-addr.arpa                             db.127.0.0
cache       .                                                db.cache
```

第一条 secondary 语句使其成为 sxu.edu.cn 域的辅名字服务器。该语句告诉 named 从 IP 地址为 202.207.208.8 的服务器上获取 sxu.edu.cn 域的数据，并将其存放到/etc/named/db.sxu.bak 文件中去，如果该文件不存在，named 创建它，并从远程服务器上获取数据，然后将数据写入新创建的文件中去。如果该文件存在，named 检查远程服务器中的数据同文件中数据是否有区别。如果有区别，named 获取最新数据并用新数据重写该文件内容。如果数据未改变，named 装入磁盘文件中的内容就不再进行区传输。在本地磁盘上保留一份数据库文件的拷贝使得辅名字服务器不必在每次重启时都要传输一次区数据，只有数据改变时才有必要传输区文件。

第二条 secondary 语句说明本地服务器也是反向域 208.207.202.in-addr. arpa 的辅名字服务器，并且该域的数据也从 202.207.208.8 中存取。

(2)标准资源记录

域名系统的数据库文件 db file 上(db.DOMAIN，db.ADDR，db.127.0.0 和 db.cache) 实际储存了域数据库信息。各种信息以同一类型的记录的形式组织，它们所用的记录是标准资源记录，称为 RR。主要有以下几种类型的记录：

·SOA(Start of Authority)：说明一个授权域名服务器开始工作，它包含有服务器版本号、cache 中数据的有效期、更新的时间和间隔等。

·NS(Name Server):指明了本域内的域名服务器的地址。

·A(Address):转换主机名到地址。

·PTR(Pointer):转换地址到主机名。

·MX(Mail eXchanger):标明发往给定域名的邮件应传送到的位置。

·CNAME(Canonical NAME):定义主机名别名。

·HINFO(Host INFOrmation):描述主机硬件和操作系统信息。

·WKS(Well Known Service):通告网络服务。

资源记录在 RFC 1034(域管理员操作指南)中定义,你可以查阅有关记录描述。

DNS 资源记录的结构是:

[name] [ttl] IN type data

name 为资源记录所引用的域对象的名字。它可以是单独的主机或整个域。名字中所输入的字符串除以点结尾的之外都是相对于当前域来写的。如果名字字段(field)为空白,则该记录应用于其上最近命名的域对象。

ttl 生存期以秒为单位定义了时间长度,它用于确定该资源记录中的信息在缓存中保存的时间长度。通常该字段是一个空白,若使用在 SOA 记录中,则为整个区设置的默认 ttl.

IN 标明该记录为 Internet DNS 资源记录。有一些其他类的记录,但并不用在 DNS 中。

type 标明这是一种什么类型的资源记录。

data 规范这一类型资源记录的信息。例如,在 A 类记录中,这一字段包含了实际 IP 地址。

在下面的部分中,我们将讨论各种库文件的配置。

(3)缓存初始化文件(db.cache)

db.DOMAIN 中的 cache 语句指向缓存初始化文件。每一台维护缓存的服务器都有这样一个文件。它包含了当名字服务器启动并开始创建域数据缓存时必须的信息。根域由带点的 cache 语句指出,db.cache 文件至少包含根名字服务器的名字和地址。db.cache 文件内容如下:

```
;;Initial cache data for root domain servers
;;list of servers
edu.cn.   99999999      IN      NS          ns2.net.edu.cn.
.         99999999      IN      NS      B.ROOT-SERVERS.NET.
.         99999999      IN      NS      C.ROOT-SERVERS.NET.
.         99999999      IN      NS      D.ROOT-SERVERS.NET.
.         99999999      IN      NS      E.ROOT-SERVERS.NET.
.         99999999      IN      NS      I.ROOT-SERVERS.NET.
.         99999999      IN      NS      F.ROOT-SERVERS.NET.
.         99999999      IN      NS      G.ROOT-SERVERS.NET.
.         99999999      IN      NS      NS.INTERNIC.NET.
.         99999999      IN      NS      H.ROOT-SERVERS.NET.
;;...        and their address;
ns2.net.edu.cn.                 99999999        IN      A       202.112.0.33
```