Linux 风暴系列

Linux 服务器管理员教程

王虹宇　张福利　编著

国防工业出版社

·北京·

图书在版编目(CIP)数据

Linux 服务器管理员教程/王虹宇,张福利编著 .
北京:国防工业出版社,2001.1
(Linux 风暴系列)
ISBN 7-118-02371-X

Ⅰ.L... Ⅱ.①王... ②张... Ⅲ.操作系统 .Linux
Ⅳ.TP316.89

中国版本图书馆 CIP 数据核字(2000)第 42137 号

国防工业出版社出版发行
(北京市海淀区紫竹院南路 23 号)
(邮政编码 100044)
北京奥隆印刷厂印刷
新华书店经售
*
开本 787×1092 1/16 印张 23¼ 532 千字
2001 年 1 月第 1 版 2001 年 1 月北京第 1 次印刷
印数:1-5000 册 定价:32.00 元

序　言

Linux 是计算机发展历史上的独特现象。虽然它滥觞于一位普通大学生的灵感与才思，却已成为当今最为流行的免费操作系统。对很多人来说，Linux 是一个谜，免费的东西怎么会变得如此有价值？事实上 Linux 的确稳定而富有竞争力，许多大学与研究机构都使用 Linux 完成他们的日常计算任务，同时，Linux 也逐渐成为各公司服务器的首选操作系统，许多公司将它用于邮件服务器或是 WWW 服务器，用于 DNS、路由和防火墙等。相信在不久的将来，人们在家用 PC 上也会广泛使用 Linux。

近年来，Linux 在中国也有了很大的发展，特别是随着 Internet 的普及，Linux 的发展更是如火如荼，国内的很多城市都成立了 Linux 俱乐部，Linux 发烧友队伍日益壮大，从而掀起了一场 Linux 风暴。

正是在 Linux 蓬勃发展之际，我们组织编写了本套丛书，旨在为一部分读者解开 Linux 成功之谜，更为 Linux 在中国的普及和发展贡献一份力量。

国防工业出版社计算机编辑室

前　言

相信每一位朋友都对大学的 Unix 课程有一定的印象。命令、文件操作、shell、X Window，最多再加上一些依赖于特定版本的 Unix 特定管理工具，就构成了一门可学可不学的课程。这倒不能怪学校，由于历史的原因，许多学校的 Unix 课程都以 Sun OS 或者 SCO Unix 作为教学的范本。笔者无意否认 Sun OS 和 SCO Unix 在 Unix 界的地位，但是由于软硬件的问题，在一台 Sun SPARC 上面让每个学生用超级用户权限折腾一气肯定是不现实的。还有许多问题，例如这些商品化的 Unix 系统通常不包含任何必要的系统工具，甚至通常不包含编译器，使得在这种系统上面用户唯一能做的就是编写 shell 脚本，或者其他任何将 Unix 变成个人工作站的操作。

事实上，Unix 恰好不是做这种工作的合适平台。实际上，除了高端的 3D 图形和 CAD，大部分 Unix 系统是作为网络服务器出现的。对于一个 Unix 人员来说，熟悉系统的管理远远比熟悉个人环境的定制重要得多。然而，在我们熟悉的传统教学方式中，却无法让学生学习 Unix 系统的管理，也无法接触包括安装、定制内核甚至更改系统源代码这样的任务，而这种任务恰好是每一个 Unix 管理者必须了解的。

我们需要一个可以免费使用的 Unix 系统，它必须具有 Unix 的基本特性，其复杂程度要涵盖在外观/管理这个层次上的所有 Unix 特性，而且不能偏离标准的 Unix 系统管理特性太远。像 SCO Unix 之类的产品在给了用户一个貌似简单的管理界面的同时，也使得用户同真正的 Unix 系统隔离开来，这样首先是用户在面对其他 Unix 产品的时候不知所措，其次是一旦出现真正的问题也无法处理。这样的情况我们在 Windows NT 中已经看到了很多。世界上有许多 NT 管理者，但是当 NT 出现问题的时候有几个人能成功地把问题解决呢?

另一方面，我们用来教学的 Unix 系统必须具有足够的标准 Unix 应用，特别是各种网络服务程序。我并不认为像 vi 和 more、less 这样的程序对于学习 Unix 的人是必须了解的，相反，如果一个 Unix 系统没有提供 C 编译器和 WWW 服务程序，那么用它来干什么呢? 同样的道理，像 Apache 这样的程序，肯定不属于传统 Unix 的一部分，但是，如果你是一个实际工作中的 Unix 应用人员，不了解它是肯定不行的。

符合我们要求的产品有两种，分别是 Linux 和 FreeBSD。从技术上说，FreeBSD 更适合作为标准的服务器操作系统，但是我们认为，从教学上看，Linux 更容易掌握一些。何况，FreeBSD 的发展速度要比 Linux 慢得多，特别是 FreeBSD 对于高档硬件的支持还相当不能令人满意。

从教学的角度看来，Linux 是一种非常合适的教学平台，首先是它借鉴了 Unix 两大流派——System V 和 BSD 各自的优点，同时它非常简单，无论初学者还是其他 Unix 的使用者

都能很快地掌握它的主要特色，特别是能够迅速地掌握它的系统管理概念。而且，一旦掌握了 Linux 系统管理的基本技术，迁移到其他几种商业化的 Unix，如 Sun Solaris、IRIX 等都是很简单的事情。另外，Linux 下可以运行大部分免费的网络服务程序，如 sendmail、Apache、Samba 等等。在传统的 Unix 课程中，我们很难介绍这些东西，因为这种程序的设置大部分都需要对系统进行各种定制或者改变，至少需要超级用户权限；而在 Linux 下，我们可以让每个学生自己试验完成网络服务的配置和管理。

因此，我们决定用 Linux 作为我们 Unix 系统管理课程的基点，这是本书的缘起。除了教学以外，我们还希望这本书成为一本能够为广大 Linux 爱好者提供系统知识的参考书。近几年来，Linux 在我国的发展还算比较迅速，但是市面上的 Linux 书籍对 Linux 的介绍，多半停留在系统安装—配置工具—图形界面的层次上，而且往往是针对某个版本的 Linux 的具体介绍，很少有对 Linux 系统配置和管理的普遍性介绍。本书不奢望成为《Unix 系统管理技术》那样的名作，但希望能起一个抛砖引玉的作用。

笔者在自己所在单位使用 Linux 已经有一段时间了。事实上，在笔者的“权力所及”范围之内，除了核心应用（使用 IBM 的小型机）、桌面（使用 Windows + Office）和为某些使用 FrontPage 的学生提供的主页服务之外，其他的系统，尤其是网络服务器全部换成了 Linux。在这个过程中，笔者充分感受到了 Linux 的最大特性：简单。现在，笔者已经不想使用 NT 或者任何商业化的 Unix 系统了，因为它们总是把本来很简单的事情弄得十分复杂。并且我相信，如果我可以在不太长的时间内从一个仅有一点 Windows NT 系统管理知识的外行成为能够解决一般性的 Linux 系统问题的管理员的话，那么你也一定能够做到。

本书是按照教程的体例编写的，建议你在吃完晚饭之后读它，或者去听这样一门课。本书不是 Linux 系统管理员手册，所以不会像系统管理员手册那样包含详尽的命令开关、配置语句列表这样的内容，相反，我们满足于用实例告诉朋友们“如何去做”，以及一旦出现问题应该如何解决。

考虑到教学需要，本书将读者定位在初步使用过 Linux，准备研究系统管理的用户这个层次上。因为这个原因，本书首先介绍了一些对于普通读者可能不十分熟悉的概念，第 1 章介绍了 TCP/IP 网络服务的基本结构，特别是路由问题。第 2 章介绍几个比较容易出现问题的 Linux 概念，诸如 setuid、setgid 以及 inetd 守护进程等等。第 3 章和第 4 章介绍了系统的基本安装和配置过程，由于现在的 Linux 发行版本特别繁杂，所以本书将集中讨论具有共性的部分，以及手工调整系统服务的过程。

从第 5 章到第 8 章，按照局域网—内部网—外部网的次序介绍了如何将 Linux 定制成为一台能够满足实际工作需要的服务器，内容覆盖了目前 Linux 的主要应用，如 Samba、HTTP、邮件服务器、防火墙等等。不过由于种种原因，仍然有一些应用没有介绍，如数据库服务、LDAP 等等。对于这些未能列入教程中的内容，笔者感到抱歉。此外，由于个人水平问题，本书未能充分地介绍 Linux 的最新发展。

第 9 章以后是管理员的传统问题。其中，任务自动化这一部分是作者经过再三考虑选择的。尽管用 Perl 编写自动化任务处理程序显得不够 cool，但是 Perl 确实是笔者知道的最好的自动化系统管理工具。我不是一个 shell 狂，说实话我讨厌编写 shell 程序。我想，Linux 存在的秘诀之一就是它“有用”，而我确信，在第 11 章写的简短的 Perl 介绍，还是能够帮助

朋友们解决一些系统管理的实际问题的。至于 expect 应该没有什么可说的，毕竟这是一种很好用的把交互过程自动完成的工具。

如果作为一本教程，这本书大可不必讲完，像 qmail、fetchmail、wvdial 和 squid 这样的东西都仅仅具有实践价值，本书将它们包括进来的主要理由是：(1)它们有用；(2)世界上的大部分 Linux(或者 Unix)用户正在使用它们。同样，本书也没有介绍 OSI 模型，理由当然是世界上并没有这样的产品。抱着现实主义的态度总是有好处的，不管是对这本书还是对 Linux。

附录中介绍了一些必要的工具的使用，特别是 joe。差不多每一本 Unix/Linux 教科书都告诉你应该用 vi、awk 和 shell 进行操作，因为别的东西都要编译才能使用。我对此感到头晕。如果一个 Unix 系统不提供 C 编译器，也不让我自己安装任何程序，我不知道我该如何管理它。附录中还有一个简短的常用命令列表，如果你认为这个东西对你毫无用处，你可以将它撕掉，这不会伤害我的感情。附录的最后一部分是 GPL 规则，这可能对你有用，特别在你需要说服你的老板的时候，但是这个规则既不是我写的，也不是我翻译的，而且我不会从这几页中得到稿费，所以也不能对它的内容负责。

本书主要由王虹宇、张福利编著，同时参与编写工作的还有：薛小香、郑桂水、林章庆、杨旺平、张晓章、杨成刚、郑吉林、黄建森、林振宁、康拥红、梁清、刘小峰等。

在阅读或者教授本书的过程中，你无疑会发现本书作者的愚蠢之处，对此我只能请你包涵，并且欢迎你来信和我讨论，归根结底，如果你能指出我的错误，我是感激不尽的。

内 容 简 介

本书按照当前的 Linux 内核版本和发行版本的主要内容，介绍如何在一个普通的网络系统中使用 Linux 作为网络服务器，以及如何对这样的 Linux 服务器进行管理。全书首先介绍了 TCP/IP 和 Unix/Linux 网络服务系统的一些重要概念，然后按照安装→配置基本系统→配置网络服务器→网络服务器管理的次序介绍 Linux 的应用，内容包含目前 Linux 系统的主要应用，如文件/打印服务器，Internet/Intranet 服务器，路由器/防火墙等等，并且讨论了在系统管理中容易遇到的主要问题。

本书适合对 Linux 有初步了解，希望系统地了解 Linux 的服务器端应用或者准备在本单位网络上应用 Linux 的用户，对其他 Unix 的系统管理员也有一定的参考作用。也可以作为 Unix/Linux 网络管理员的基础教程使用。

目　录

第 1 章　TCP/IP 基础

本章介绍 TCP/IP 和 Unix 联网的基本知识，以便使读者能够顺利理解配置服务器中的一些困难问题。

本章具体包括以下内容。

- TCP/IP 栈的基本结构
- IP 地址解析和路由
- TCP/UDP 协议的概况

1.1　基 本 概 念

1.1.1　TCP/IP

TCP/IP 是 Unix/Linux 世界中网络的基础，甚至，在某种意义上，Unix 网络就是 TCP/IP，而且 TCP/IP 也就是网际互联的标准。你可能知道有一个叫 OSI(开放系统互联标准)的模型，它是教科书上描述网络互联的标准模型，不幸的是，这个标准在实际的网络世界中毫无意义，尽管许多教程会以它为模板传授网络理论。这是对标准化和美国政府的一种嘲弄。事实是，TCP/IP 在前进，而 OSI 已经不复存在。

本书试图直接按 TCP/IP 的构成来理解问题，尽管这也许不会在考试中给你带来高分，但它却是唯一合理的方式。如果你喜欢 OSI 那种似乎井井有条的方式，你总会找到合适的教材的。

作为网络互联的标准，TCP/IP 给出的是一系列协议，这些协议要完成各种各样的网络应用支持。你可以合理地把这协议看成一组编程接口，一旦应用程序了解这些接口便能使用网络。为了实现这种抽象，TCP/IP 被分成了四层，我们可以通过一个 WWW 浏览器来看看这四层的概况：

应用层：WWW，SMTP，DNS，FTP
传输层：解释数据
网路层：定位 IP 地址、确定连接路径
链路层：与硬件驱动程序对话

首先 WWW 浏览器必须调用 HTTP 协议，这个协议规定用什么样的命令来得到 WWW

文本,这种协议构成了 TCP/IP 的最高层——应用层。

为了使用 HTTP 协议,浏览器要把命令发送到服务器上去,并且从服务器得出回答,但是,必须记住,网络上传输的总是一些字节,哪些字节是命令,哪些是回送数据,又有哪些是用于表示"就绪"、"传输中"或者"停止"的验证码呢?这些解释工作需要一串复杂的协议进行控制,这构成了第三层——传输层。

假定所有校验和和控制字都已经完毕,现在要把这些数据真正发送到对方那里,那么,面对下一个问题是:这台机器到底位于哪里?如何保证数据被发送到了我希望的位置?这种地址定位构成了第二层——网路层。

最底层——链路层完成最终的工作:将地址、数据等等变换成真正的电气信号,并且在网络上送出,让网络设备彼此对话。

在以后的各节中,我们将会概述一下各层的情况以及与我们相关的主要的问题。这里讨论的只是主要的问题。许多问题由于仅有历史的意义而无需考虑,而另外一些问题不太像是一个 Linux 管理者会非常关心的,例如 ATM 交换。相反,如果要管理 Linux 服务器,迟早你必需理解 ARP 和路由选择。

从理论上说一个服务器管理员也许根本不需要了解下面陈述的问题,也许需要知道更深奥的知识。但是根据我的经验,一旦你成为服务器管理者,并且你是在真实世界而不是官僚机构中,你将很快发现一切困难的任务都必须由你来完成,而且毫无商量余地,最终你会看到你将成为整个系统的管理者。了解足够多的东西,你将会在系统管理员的恶梦中生存下来。

在后面的解释中,你将看到 TCP/IP 是如何工作的,但是我们必须先说明一件事情,TCP/IP 的设计是有弹性的,实际上,存在某种方法把其他网络协议当成 TCP/IP 的基础,例如,我们一般让 TCP/IP 协议直接与网卡驱动程序对话,但是,也可以想象这样的情况,把其他的网络协议,例如 IPX 当成一种虚拟的"网络设备",这样,TCP/IP 就可以运行在其他的网络协议之上,这种技术被称为 IP 隧道。不过,在本章中,将不涉及如此复杂的问题。

1.1.2 IP 地址和子网

你肯定知道,互联网用 IP 地址来标识主机地址,而数据传输则是通过把数据分成一系列的小"包"来完成。TCP/IP 世界里的每一台主机都要有唯一的 IP 地址,这个 IP 地址是一个 32 位无符号整数,不过通常使用点分十进制表示。例如,00010000110000001111111000000001 是 281079553 的二进制形式,但是实际应用中把它按照 8 位一段的方法分成四段,第一段是 00010000,也就是 16,第二段是 11000000,十进制是 192,同样,剩下来的两段是 254 和 1,因此这个地址是 16.192.254.1。

这听起来很简单,但是在实际实现中有些复杂,主要问题是 TCP/IP 必须兼顾许多困难的问题,其中之一是网络连接。全世界有太多的网络主机,随便给你一个主机号,你怎么知道它在哪里?最大问题在于,或许 13.2.0.5 在美国纽约,而 64.0.7.6 却在北京,如果搜索一台主机地址就要查阅全世界的主机列表的话,那么互联网将立即崩溃。

解决这一个问题的第一个要点是网络分类。分类把世界上的 IP 地址分段,各段之间独立管理,通常每一段用同样的方式连接到一起,本段的机器之间可以直接互相访问,这样的

一个段称为一个系列网络地址。

为了管理的方便,分段用一个有点古怪的方式。例如,对大公司(比如IBM),一个上万台机器的网络很正常,而小公司也许只有十几台机器,TCP/IP实现中用A、B、C类地址来处理这个问题。

A类地址用于超过65534个主机的网络,例如,一个前缀为18的网络使用18.0.0.0~18.255.255.254的网络地址,这些地址之间是直通的,主机之间可以彼此访问。为了说明这一点,TCP/IP使用网络掩码和网络地址的概念。

在上面的例子中,网络地址是18.0.0.0,这表示网络的包含地址段是18.0.0.0~18.255.255.254,即地址部分除去前缀后,余下的部分由全零变成全1,不过全1的地址将保留为广播使用。

与网络地址相应,对在此网络中的主机,TCP/IP使用网络掩码。在上面的例子中,掩码是255.0.0.0,其含义是这样:假设18.0.0.3想和18.11.0.75对话,那么,将这两个地址相互异或,再与掩码取"与"(&),结果是零,说明这两个地址可以直接对话;相反,如果要与19.1.1.1对话,那么运算之后不为零,说明必须使用间接方式才能到达。

可以用直接方式理解掩码,掩码中的"1"用来描述网络地址(前缀),"0"用来描述子网的主机,如:255.0.0.0意味着将主机地址的前8位解释为网络号,而后24位解释成网内的主机地址。也就是说,与某个A类主机地址前8位相同的主机地址被认为是在同一子网之内。

A、B、C类用下面的方式规定:

A类地址使用255.0.0.0的掩码,为了管理方便,规定A类地址总是使用头一位为0的地址值,这意味着A类地址是从1.0.0.0到126.0.0.0,另外还有一个特殊的A类地址127.0.0.0它用来表示"本机",即自身。

B类地址使用255.255.0.0的掩码,B类网从128.0.0.1到191.255.0.0。

C类地址的掩码是255.255.255.0,网络地址从192.0.0.0到233.255.255.0。

首字节在224以上的地址用于实验和开发,部分用于组播,通常无需关心。

地址255为特殊的含义,它用于广播,即"向本网上所有人通话"。这个概念对任何人都应该是很容易理解的。使用广播有两种方法,并且是几乎等价的,即:(1)使用全1的地址,即255.255.255.255;(2)使用本址广播,即用网络号加上全1,例如,A类网18.0.0.0可以使用18.255.255.255广播,B类网190.4.0.0的广播地址则是190.4.255.255。无论哪一种,对于广播地址的访问都将引发将数据发送给本网络上的所有机器。

计算网络掩码和标志子网地址是网络管理者经常需要进行的工作,第一种计算是根据某个确定的主机地址和它的网络掩码求出所有可以和它直接通信(在同一子网之内)的主机地址。例如,主机地址是202.112.50.3,掩码是255.255.255.0,与它可以直接通话的主机的地址可以这样计算:掩码是255.255.255.0,因此32位网络地址的前24位被解释为网络地址,所以凡是和202.112.50.3的前24位内容相同的主机地址就和它处在同一子网之内,后8位是子网里面的机器地址,它可以从全0变到全1,所以所有和这个主机在同一子网之内的主机的地址是202.112.50.0~202.112.50.255。

网络地址/掩码一般用斜线分开,如网络地址是202.199.248.0,掩码是

255.255.255.0,在 Unix 中一般写成 202.199.248.0/255.255.255.0。但是也有另外一种写法,就是用掩码中 1 的个数来代替掩码的实际形式。例如,255.255.255.0 的前 24 位是 1,其他位是 0,因此可以写成 202.199.248.0/24。同样,下面的两栏地址形式是彼此等效的:

202.199.248.0/255.255.255.0	202.199.248.0/24
122.24.0.0/255.255.0.0	122.24.0.0/16
13.4.5.7/255 0.0.0	13.4.5.7/8

我们在前面一直设置掩码的 0/1 分界在字节边界处,但是这其实并不是必须的。完全可以有其他形式的掩码。例如,240 等于二进制的 11110000,因此一个 255.255.255.240 的掩码意味着前面 28 位是网络地址,而后面 4 位是子网内的 IP。同样,网络地址也可以不由 0 结尾。举个例子来说,202.112.50.16/255.255.255.240 是什么意思? 202.112.50.16 是二进制的 11001010011100000011001000010000,套上一个有 28 位为 1 的网络掩码,意味着最后四位由全 0 变成全 1,就是最小是 11001010011100000011001000010000,最大是 11001010011100000011001000011111,这两个数是 202.112.50.16 和 202.112.50.31,所以这个网络地址代表 202.112.50.16 到 202.112.50.31 的所有主机。这个网络地址也可以写成 202.112.50.16/28。

1.1.3 网络硬件

TCP/IP 并不关心具体的网络实现,但是不幸的是系统管理员必须了解。你的网卡可能会失效,以太网可能会拥塞,调制解调器会被击穿,还有诸如此类的种种问题。另一方面,特别对于以太网,如何将 IP 地址译码成实际的电气位置是非常重要的问题。

对于一般的系统,最重要的两种网络连接是以太网和电话线。以太网使用同轴电缆,双绞线或光纤作为连接介质,而电话线则为广域网提供主要的连接手段。

以太网无疑是最主要的网络技术,对一个 DIY 人员来说,10MB 或 100MB 以太网的硬件连接如此简单以至于无需在这里作任何描述。但是有几个特殊点可能仍然需要指出。

首先是以太网的通信协议,以太网使用一种载波监听/碰撞检测的方式工作,因为线路上跑的是电气信号,所以以太网卡通过监视电缆电位来判断是否有某个设备在传输数据。如果网卡接到一个发送任务,它首先判断线路上是否有数据传输,如果有,它等待一会,如果没有,它就以一个特定的数据包大小(帧)将数据发送到线路上。

以太网的帧格式有些怪异,它允许在帧中包含帧的类型,当一个帧到达某台计算机时,计算机根据帧的类型作不同的处理。如果你管理过 Netware,那么肯定会面对 802.2 或 802.3 以及 Ethernet - II 的种种令人晕头转向的配置问题。幸运的是,在 Linux 或者 Windows 的世界里,你极少需要真正处理它们。

当一个帧被发送到网络上时,它实际上是在网络的整个电气连接上漫游。“电气连接”指电信号可以传递到的任何地方,比如,集线器上连接的几片网卡,穿过几根电缆或双绞线,等等。然而一台机器上的两片网卡不在一个电气连接上,因为电信号无法直接在两者之间传递(你会问如果这两片卡正好插在同一个集线器上会如何,我只能说:! @ # $%)。每个以太网卡必须有一个唯一的“以太网地址”,也叫 MAC 地址。它是一个 48 位整数,理论上,