第二部分　Linux系统管理员指南

Lars Wirzenius　著

作者简介

几乎从Linux诞生之日起，Lars Wirzenius就和它结下了不解之缘，可以说他是第二个在自己计算机上运行Linux系统的人。在觉得内核编程索然无趣之后，他把精力集中在高层次的问题上，比如主持comp.os.linux.announce新闻组。1992年，他和Michael K.Johnson以及Matt Welsh一起，合作成立了“ Linux文档项目 ”（简称LDP）。LDP着眼于为Linux用户、程序员和系统管理员制定完整而免费的文档集。这个项目已成功接近尾声。 Lars为LDP编写了《Linux系统管理员指南》。

Lars仍然执迷于LDP，同时也是Debian Project（Linux版本之一）开发人员。他的大部分时间都花在实现自己喜爱的项目上。如想进一步了解他的工作情况，可访问他的主页，其地址是http://www.iki.fi/liw。也可通过电子邮件联系他，他的邮件地址是 http://liw@iki.fi（遗憾的是，为保证自己的休息时间， Lars通常不回答常见的Linux问题。但他热忱欢迎大家提出本书以及和他的工作项目相关的问题和建议）。

本书简介

从哪里获得源代码和预先格式化好的版本

本书的源代码和其他机器可读的格式可通过匿名 FTP在因特网上找到， Linux文档项目的主页（地址是http://sunsite.unc.edu/LDP）和本书的主页（http://www.iki.fi/liw.linux/sag/）上都可找到本书。可以用的格式有 PostScript和TeX.DVI格式。

前　言

“开始写本书的时候，脑海中一片空白。当作者的手指在键盘上游移时，口中念着：该写点内容了，于是就有了内容”。

《Linux系统管理员指南》主要讲述利用Linux进行系统管理方面的知识。它是为对系统管理一无所知的读者编写的，不过读者应该具备起码的基本操作知识。这本指南没有介绍如何安装、设置Linux；因为这些知识可在相关的“安装和启动”文档内找到。看了下面的内容，你会对各种Linux手册有更多的了解。

系统管理就是一个用户要将计算机系统维持在一个正常工作状态所必需做的事情。它包含了复制文件（如需要的话可以恢复），安装新的程序，为用户建立帐户（不再需要的时候再取消），确保文件系统正常等等。如果把一台计算机比作一间房子，系统管理就好似对这间房子的保养，比如打扫清洁，修理坏了的窗户等等类似的事情。不过，系统管理不同于保养，因为那样就太简单了（有人一定要称谓它为保养，但那不过是因为他们还没看过这本手册）。

在本书的结构安排上，很多章节相互之间都是独立的，所以如你需要某方面的相关信息时，比如说备份方面的，你就可以有针对性地只看那一章节。这就很容易地将这本书当成工具手册来翻阅，需要哪部分就查看哪部分。尽管这样，这本书主要还是一本指导性手册，偶尔也可兼作工作参考手册。

本手册不打算对Linux系统的方方面面逐一讨论。对系统管理员来说，其他的Linux文档也同样不可忽视。毕竟，系统管理员是一个拥有特殊权限和职责的用户。手册页也是一个非常重要的资源，大家在用到一个不熟悉的指令时，就可以在上面查到相关信息。

本手册以Linux为讨论主题时，还遵循了这样的普遍准则，即本书要能同时适用于其他与Unix相关的操作系统。但不幸的是，一般说来，Unix不同版本之间的差异太多，尤其是系统管理方面。所以要想全盘考虑所有的版本根本不可能。而且，由于Linux自身的发展，全面讲述Linux的方方面面也不是件容易的事。

现在，由于Linux极富个性，还没有一个“正式”的Linux版本，所以不同的用户有不同的设置，并且很多用户都只采用自己的设置。本书不是以某一个版本为讨论对象的，虽然我本人一直专用Debian GNU/Linux系统。只要可能，我仍然会尽力指出另外几个版本间的区别，并对它们进行详细说明。

我打算详细说明Linux系统的工作原理，而不是罗列相关的操作任务。也就是说，本书包含的信息可能不是每个人都需要了解的，但大部分如此。.如果你采用的是一个预先配置好的系统，大可跳过某些章节。通过本书的讲解，你将自然而然地增进对Linux系统的了解，更轻松愉快地使用和管理它。

像其他所有与Linux相关的开发一样，这本书的写作也是出于自愿。我写这本书只是因为我觉得这件事比较有趣，也非常必需。尽管如此，就像所有的义务工作一样，我所付出的努力，运用的知识和经验也是有限的。这就意味着，这本手册并没有当初心目中那么好，假设我能够得到一笔丰厚的报酬，并花一定的时间去完善它，我想这本书会出色得多。

在有一点上，我取了巧；对于记载在其他适用免费手册上的内容，我在这本手册里就没再多写。特别是对与程序有关的文档，例如选用mks的所有细节，我仅讲述了这个程序的目的，也尽可能地讲它的用法。如要了解更深的知识，我建议读者看一下其他的资料。通常，所有和这些文档有关的东西都是整个Linux文档集的一部分。

尽管我曾尽力使本书完善，但我仍然想听听你们有什么关于完善这本书的宝贵意见。语言不得体，内容出错，有重复的内容或部分，还是要补充关于不同的 Unix变体版本的信息，我对这些建议都非常关心。通过万维网的 http://www.iki.fi/liw/mail to lasu.html和我联系。

我写本书的过程中，很多人曾帮过我，不论直接地还是间接地。我特别要感谢 Matt Welsh，他给我的鼓励和领导着 LDP的完善工作；还有 Andy Oram，为我提供有价值的反馈信息，令我不断更新作品； Olaf Kirch为我指出哪些内容是必须包括的，以及 Adam Richter和其他人，他们为我指出大家都比较感兴趣的热点问题。

感谢Stephen.Tweedie、H.Peter Anvin、R ’emy Card和Theodore Ts ’o，允许我借用他们的作品（因此使这本书看起来比较厚，也更容易让人留下深刻印象），它们是：xva和ext2文件系统之间的对比、 ext2文件系统的描述和设备列表。我对这点非常感激，非常遗憾早期的版本中欠缺这些东西。

除此之外，我还要感谢 Mark Komarinski，他在1993年，将自己的资料以及许多系统管理栏目放入Linux Journal中。这些内容令人大开眼界，也很富有启发性。

另外，还有许多人为我提供了许多有用信息，已知的人员有（按字母表顺序）：Paul Caprioli、Ales Cepek、Marie-France Declerfayt、Dave Dobson、Olaf Flebbe、Helmut Geyer、Larry Greenfield及其父亲、 Stephen Harris、Jyrki Havia、Jim Haynes、York Lam、Timothy Andrew Lister、Jim Lynch、Michael J.Micek、Jacob Navia、Dan Poirier、Daniel Quinlan、Jouni K Seppanen、Philippe Steindl、G.B.Stotte。如有遗漏，我很感报歉。

Linux文档项目

Linux文档项目，即LDP，是一个由作者、校对人员和编辑组成的松散组织。我们共同为Linux操作系统提供完整的文档。该项目的领头人是 Greg Hankins。

本手册只是LDP出版的系列丛书之一，这一系列中包括《 Linux用户指南》、《系统管理员指南》、《网络管理员指南》以及《内核黑客指南》。这些手册以源码格式、 .dvi格式和PostScript输出的格式提供，可通过匿名 FTP，从sunsite.unc.edu的/pub/Linux/docs/LDP目录中下载。

我们鼓励任何有写作和编辑爱好的人加入我们的行列，改进 Linux文档。如果你能上网的话，你可通过gregh@sunsite.unc.edu，和Greg Hankis取得联系。

第1章　Linux系统综述

本章是对Linux系统的概述。首先为大家讲述该操作系统提供的主要服务。然后粗略谈谈实现这些服务的程序。本章的目的是使大家对该系统有一个全面了解，具体的各个部分将在以后章节里讲述。

1.1　操作系统的各个组件

Unix操作系统由一个内核和一些系统程序组成。其中也有执行特定工作的应用程序。内核是操作系统的核心（实际上，它通常被误认为是操作系统本身，但事实并非如此。操作系统提供的服务比内核提供的服务要多得多）。它能维护磁盘磁道中的文件、启动并同时运行多个程序、将存储空间和其他资源分配给不同程序，在网上收发数据包等。内核自身所做的工作少之又少，但它能提供建立所有服务程序的工具。它还能阻止任何用户直接访问硬盘，迫使每个用户都使用它提供的工具。通过这种方法，内核为用户相互间提供了一种保障。内核提供的工具是通过系统调用来使用的；关于这方面的详情，可参考手册的第二部分。

系统程序利用内核所提供的工具执行操作系统要求的各种服务程序。系统程序和其他所有的程序一起，以“用户模式”运行于内核顶部。系统程序和应用程序之间的区别在于其目的不同：应用程序用来做一些有用的、实际的事（或是娱乐，假如它正好是游戏的话），而另一方面，系统程序则是用来维护系统工作的。例如，字处理程序是一个应用程序； Telnet是一个系统程序。通常，系统程序和应用程序之间的界限有些模糊，虽然如此，这种区别对那些热衷于归类的人来说，仍然是非常重要的。

操作系统中，还包括编译程序和与它们对应的库（尤其是 Linux下的GCC和C语言库），虽然并非所有的编程语言都必需成为操作系统中的一部分。文档，有时甚至于游戏都可成为操作系统的一部分。过去，操作系统一直由安装盘或安装磁带来定义，但 Linux则不一样，它相当个性化，任何人只要有兴趣，都可在全球各 FTP站点下载并制定自己的操作系统。

1.2　内核的重要组件

Linux内核由几个重要部件组成：进程管理、内存管理、硬件设备驱动程序、文件系统驱动程序、网络管理和其他零碎的东西。图 1-1展示了部分组件。

内核部件中，最重要（没有它，什么也干不了的）的可能是内存管理和进程管理。内存管理照管已分配给进程、内核部件和缓冲区的内存区和交换空间。进程管理则创建进程，并通过在处理器上交换活动进程的方式，实施多任务操作。

在最低级上，针对每个自己支持的硬件设备，内核中都包含相应的驱动程序。由于各种硬件设备名目繁多，所以对应的驱动程序也多如牛毛。有些硬件设备的行为会因为驱动程序的不同而不同。不过，按其类似之处可以对支持类似操作的设备进行归类；同类的设备采用同样的方式与内核中的其他部件沟通，但实施方式不尽相同。例如，所有的磁盘驱动程序看起来和内核中的其他部件差不多，也就是说，它们都有类似于“初始化驱动器”、“读取扇区N”

和“写入扇区N”之类的操作。

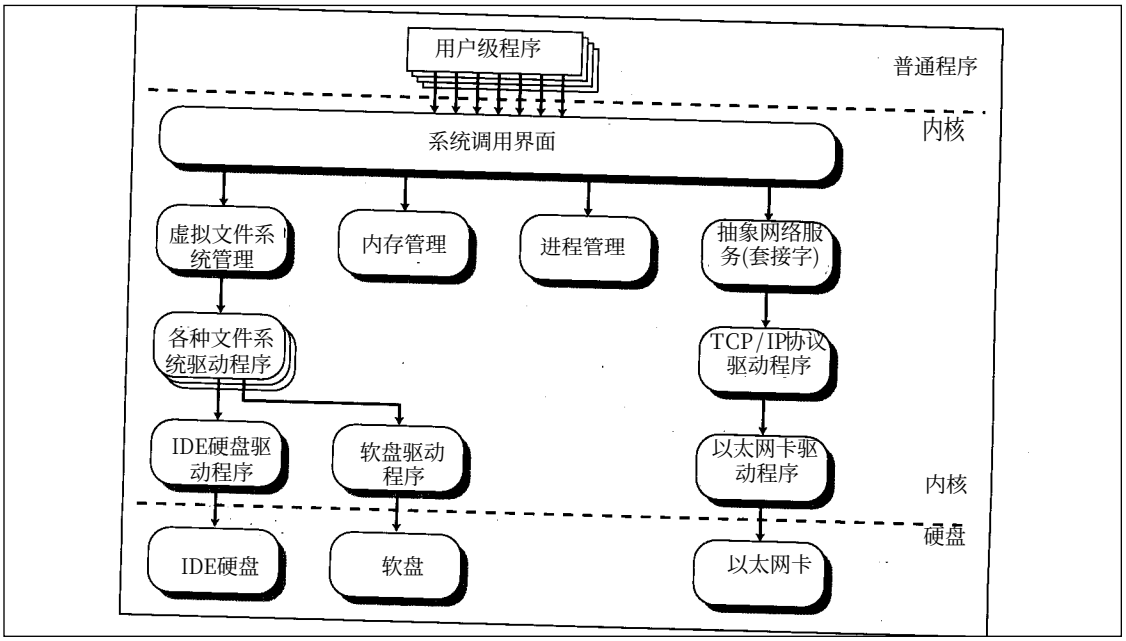

图1-1 Linux内核的几个重要部件

内核本身提供的某些软件服务也有类似的属性，因此，也可将具有类似属性的服务归入一类。例如，各种网络协议已被归入一个编程接口——BSD套接字库。另一个例子是虚拟文件系统（VFS）层，它把文件系统操作从其实施过程中提了出来。每个文件系统类型都提供各个文件系统操作的实施。在有些实体打算使用文件系统时，请求就会到达 VFS，然后VFS再把请求路由到恰当的文件系统驱动程序。

1.3 Unix系统提供的主要服务

本小节将概括解释一些重要的Unix服务，其详情参见后续章节。

1.3.1 init

Unix系统中，最重要的服务是由 init提供的。init是每个Unix系统中第一个启动的进程，也是内核在启动时所进行的最后一件事情。 init启动时，它会处理各种启动“杂活儿”（检查和装入文件系统、启动后台程序等），继续执行启动进程。

init具体做的事和个人目的有关。init通常提供单用户模式，这种模式下，没有人能够登录，只有root才能在控制台使用外壳；普通模式称为多用户模式。有的人把它统称为运行级别；单用户和多用户模式都被认为运行级别是 2，还有别的运行级别，比如X运行级别。

普通操作中，init确定getty正在运行（允许用户登录）并接收“孤儿”进程（其父进程已经死了的进程；在 Unix系统中，所有进程都必须在一个单独的树中，所以 init必须接收孤儿进程）。

系统关闭时，init就负责杀死其他所有进程、卸载所有的文件系统、中止处理期以及处理

已配置好的各种杂事。

1.3.2 从终端登录

从终端（通过串行线路）和控制台（在没有运行 X时）登录是由getty程序提供的。init为允许登录的每个终端提供一个独立的 getty实例。getty读取登录用户名，并运行 login（登录）程序，该程序便开始读取密码。如果用户名和密码都正确，login程序就开始运行shell（外壳）。由于用户名和密码不匹配而引起 shell中断时（比如，用户注销或登录中断等），init会注意到这一点，并发起一个新的 getty实例。内核不会注意到这一点，这一切都是由系统程序来控制的。

1.3.3 syslog

内核和许多系统程序都会发出错误、警告和其他消息。这些消息可稍后进行查看，所以，把它们写入一个文件是非常重要的。这就是系统日志。可以根据消息的重要性或写入者，对消息进行分类。比如，内核消息通常被直接写入一个不同于其他消息文件的文件内，因为内核消息通常更为重要，需要定期读取以探测问题的起因。

1.3.4 周期性执行的命令：cron和at

对用户和系统管理员来说，通常都需要周期性地运行某些命令。例如，系统管理员可能想运行一个命令，从老文件中清空带有临时文件（ /tmp和/var/tmp）的目录，以防止磁盘填满数据，因为并非所有的程序都能正确善后。

cron服务就是为此而设计的。每个用户都有一个 crontab文件，他们在该文件内列出自己想执行的命令及其命令执行次数。 cron后台程序负责在指定时间运行指定的命令。 at服务和cron服务类似，但它只执行一次：这个命令是在指定时间执行的，而不是重复执行的。

1.3.5 图形化用户接口

Unix和Linux都没有将用户接口结合在内核内；而是令其由用户级程序来实施。这种方式适用于文本模式和图形化环境。

这样的安排令系统更为灵活，但有个缺点：为每个程序实施不一样的用户接口令系统更难以掌握。

Linux系统主要使用的图形化环境叫作“ X Window System ”（简称X）。X也没有实施用户接口；它只实施一个窗口系统，也就是能够实施图形化用户接口的若干个工具。X窗口系统上，最常见的三类用户接口是 Athena、Motif和Open Look（最近还有KDE和Gnome，两者都是建于X之上的用户环境）。

1.3.6 连网

“ 连网 ”是指把两台或两台以上的计算机连接在一起，以便它们可以进行沟通。实际应用中的连网和沟通稍微有些复杂，但非常有用。

Unix操作系统提供了许多连网功能。大多数基本服务（文件系统、打印、备份等）都可

在网络中进行。这样一来，系统管理就变得更为简单，因为它允许集中管理，而且还可享受到微型计算处理和分布式计算的好处，比如降低成本和容错性更佳等。

但是，此处只是蜻蜓点水般地提一下连网；有关详情，还请参考本书第一部分。

1.3.7 网络登录

网络登录和普通登录稍有不同。每个终端都有一个独立的物理串行线路，通过这条线路，用户就可以登录了。对每一个通过网络登录的用户来说，有一条独立的虚拟网络连接，而且连接的多少是任意的（至少可以说有许多。网络带宽仍然是很宝贵的，所以通过一条网络连接的并发登录数实际上是有上限的）。所以，不能为每一条可能的虚拟连接运行一个独立的getty程序。通过网络登录还有几种方式，比如，正在成为 TCP/IP网络主流的 telnet和rlogin。

实现网络登录时，取代运行一大堆 getty程序的是一个独立的后台程序（ telnet和rlogin都有独立的后台程序），该后台程序负责监听所有的进入登录请求。一旦它注意到有请求进入，就会启动自己的一个新实例，处理这个登录请求；原来的实例仍然继续监听其他的登录请求。新实例的作用和getty类似。

1.3.8 网络文件系统

采用连网服务的一大优点是能够通过网络文件系统共享文件。最常用的网络文件系统称为“ 网络文件系统 ”（简称NFS），是Sun公司开发的。

有了网络文件系统，一台计算机上某个程序执行的任何文件操作通过网络，被发送到另一台机器上。这样可欺骗程序，使其认为另一台机器上的所有文件实际上在自己正在运行的机器上。这样用不着修改程序，就能方便地实现信息的共享。

1.3.9 邮件

对计算机之间的交流来说，电子邮件通常是最重要的方式。电子邮件保存在一个采用特殊格式的文件内，而特殊的邮件程序用于发送和阅读电子邮件。

每个用户都有一个接收邮件的信箱（一个采用特殊格式的文件），所有的新邮件都保存在这个信箱内。有人在发送邮件时，邮件程序就找出收件人的信箱，并把这封邮件添加到信箱文件内。如果收件人的信箱在另一台机器上，这封邮件就会被发送到另一台机器上，再由它采用最佳路由，把邮件转发到收件人的信箱。

邮件系统由许多程序组成。把邮件投递到本地或远程邮箱是通过一个程序来实现的，这个程序就是邮件传输代理或 MTA，例如sendmail和smail。而用户使用的程序（邮件用户代理或MUA，例如pine和elm）也是举不胜举的。信箱通常保存在 /var/spool/mail内。

1.3.10 打印

虽然一次只允许一个用户使用打印机，但用户间如果不能共享打印机的话，就会多花许多钱。因此，打印机就交给软件来管理，这类软件用于实施打印队列：所有的打印作业都进入队列等候打印，打印机一次只能处理一个作业，完成之后，下一个作业自动跟上。这样便减轻了用户重新组织及管理打印作业的负担。

打印队列软件还将打印输出假脱机在磁盘上，也就是说，打印作业在排队等候期间，打印文本是保存在一个文件内的。这样便允许一个应用程序迅速向打印队列软件供给打印作业。这的确非常方便，因为它允许用户在得到新的完全修订本之前，打印一份样本，而不是被动地等待打印。

1.3.11 文件系统布局

文件系统分为许多部分； root文件系统通常有 /bin、/lib、/etc、/dev以及其他几个部分；/usr文件系统内包含的是程序和一些没有发生变化的数据； /var文件系统内包含的是正在发生变化的数据（比如日志文件）；而 /home文件系统则保存每个用户的私人文件。根据硬件配置和系统管理员的决定，文件系统的布局可以不一样；甚至还可把所有的数据统统装入一个文件系统内。

第3章将详细讨论文件系统的布局；这方面的详情，还可参考 Linux文件系统标准。