Apress®

Linux System Administration Recipes

A Problem-Solution Approach

Linux 系统管理疑难解析

[美] Juliet Kemp 著

邢志安 孙连坤 张百慧 李广鹏 译

人民邮电出版社
POSTS & TELECOM PRESS

Linux
系统管理疑难解析

[美] Juliet Kemp 著
邢志安 孙连坤 张百慧 李广鹏 译

人民邮电出版社
北京

内 容 提 要

本书是为 Linux 系统管理员准备的一本 Linux 疑难处理技巧集合，它提供了快速修复 Linux 相关问题的工具和建议，旨在帮助 Linux 系统管理员在运维系统时，能够避免可能出现的各种问题，或者是在问题出现时，能够迅速定位并解决问题。

本书总共分为 12 章内容，分别介绍了集中管理网络资源、监控与升级 Linux 服务器、Linux 系统的备份与数据管理、文件系统管理、系统的安全保护、文本操作、信息输入和输出、系统故障追踪，以及时间管理和人员管理等内容。本书所讲解的疑难处理技巧均在 Debian 5.0 和 Ubuntu 9.04 版本中测试并通过，这些技巧背后的理论知识可用于任何 Linux 版本。

本书是一本不可多得的 Linux 疑难处理技巧手册，具有一定 Linux 相关知识的管理员新手可以从本书中找到一些有用的信息，具有 1 年以上 Linux 系统管理经验的管理员也可以从本书中获益。

关于作者

Juliet Kemp 是在研究生勤工助学期间，发现 Linux 系统管理的兼职工作要好于在酒吧里面做招待之后，才开始对 Linux 的态度发生极大变化——由开始淡淡的喜欢变为不可自拔。在研究了几年政治学之后，她觉得系统管理远远要比攻读硕士有趣得多。

尽管不是一名天体物理学家，但是在接下来的几年时间里，她在伦敦帝国学院对天体物理学中的 IT 危机进行了整理，并提升了团体中后端软件的性能。当她意识到自己为了解决遇到的问题，而花费了大量时间来查找相关信息时，她决定将这些信息整理成书，以飨后人。

当她在澳大利亚的悉尼旅居时，完成了本书大部分内容的编写，当然，她现在已经回伦敦定居。尽管伦敦的阳光没有悉尼那样迷人，而且也没有宽广的海洋，但是她还是义无反顾地喜欢伦敦。她喜欢骑自行车运动，并鼓励其他人也加入到骑自行车运动的大军中来，她最近用攀登运动来取代了冲浪运动（总不能在泰晤士河上冲浪吧）。她目前是一名自由职业者。

关于技术审稿人

Sean Purdy 从 7 岁时第一次接触计算机——他叔叔的苹果机。他学会了在 BBC 微型计算机上编程，就像许多老一辈的极客在 20 世纪 80 年代做过的那样。在攻读电子学学位期间，他开始接触 SunOS，从此勾起了他对类 UNIX 操作系统的终身兴趣。

Linux 源自于 20 世纪 90 年代中叶的 Slackware，就在同一时期，他终于实现了在一台 386sx 机器上运行 Linux。从那时起，他就一直致力于使 Linux 能够在所有设备上运行任何事情——游戏机、PDA 和微型笔记本电脑。

Sean 先后在几家 ISP 公司和技术公司供职，有近 15 年的 Linux、FreeBSD 和 Solaris 工作经验。他现在作为系统专家供职于 Consolidated Independent 公司，负责把音乐和视频上传到 iTunes、Amazon 和其他的数字音乐服务网站。

致　谢

向 Apress 出版社所有参与本书编著工作的人员表示诚挚的谢意：Frank Pohlmann 负责了技术编辑工作并帮我完成了本书的最初构想；Kim Wimpsett 耐心地将我带有英国口音的口述复写出来；Kylie Johnston 则接替 Sofia Marchant 负责项目的管理工作，并在本书的出版过程中负责高效、诙谐地协调所有事项。由 Sean Purdy 担任技术校订使我非常高兴，因为他是我多年的好友之一。不管怎样，都谢谢他们辛勤的校订和修正工作，是他们的付出使得本书成为优秀之作。

在我的系统管理员职业生涯中，很荣幸地遇到了几位经理，他们允许我随意运行系统，我为此深受鼓舞。尤其是我在 St Hilda's 大学和牛津大学工作的时候，IT 部门经理 Anne Wilson 非常支持并帮助我。当时我刚刚开始工作并在工作中学到很多事情（当然犯了很多错误）。当我在皇家学院工作的时候，天体物理学工作组的同事们带给我各种各样的与 IT 相关的问题，在解决过程中我又学到了许多。我非常享受在那里工作的日子。

从个人角度来讲，我想要感谢我的父母。他们非常支持我，虽然他们对 Linux 一窍不通，但是他们肯定会阅读本书。

最后，向我的伙伴 Pete 和 Doop 以及我最好的朋友 Marna 致谢。Marna 耐心地忍受我的抱怨，在关键的时候鼓励我，给我沏茶、煮咖啡并准备巧克力。我快速养成的喝咖啡的习惯就是他们的功劳。总之，纵是相隔万里，他们仍是令我敬佩的人。特别感谢 Pete 允许我以科学实验的名义对他的机器“动手动脚”。

前　言

本书是为 Linux 系统管理员准备的一本 Linux 疑难处理技巧集合，它可以作为 Linux 系统管理员的案头参考手册，也可以作为一本具有相当深度的核心技术手册，当然，前提是读者需要有时间能够静下心来仔细阅读。本书旨在为读者提供快速修复问题的工具，并帮助读者在安装 Linux 系统时，能够避免可能出现的各种问题（至少可以在问题出现时，能够帮助你迅速定位到问题所在，并尽快解决它）。之所以写作本书，是因为我在作为系统管理员时，曾经花费了大量的时间来查找这类信息。因此，本书的内容非常实用，希望能和大家一起分享。

读 者 对 象

本书适合具有 1 年 Linux 系统管理经验的管理员阅读，也就是说，本书读者需要具有一定的 Linux 相关知识。管理员新手也会从本书中找到一些有用的信息，但是还是建议他们具有一定的 Linux 知识以及故障修复经验之后，再阅读本书。本书是为具有一定系统管理经验的管理员准备的，他们可以从中获益匪浅。

本书尤其适合在小公司工作的独立系统管理员（比如，负责解决用户级问题和后端问题，而不是负责一个特定领域的管理员）阅读。

本 书 结 构

本书所有的技巧在一定程度上都是相互独立的，尽管本书某些章节包含了用于某一软件的多个疑难解析处理技巧，而且在这些章节中，基础知识都是出现在最初的一个或几个技巧中，随后的技巧则要求读者具有一定的软件知识。特别是在第 2 章中，与 Kerberos 和 LDAP 相关的技巧都具有很强的相关性。

本书结构如下。

第 1 章，“节省您的精力”，包含的提示可以让您的工作更好地文档化，可以提升您的 shell 和 Perl 脚本编程能力（因为将来某一天您会重新读到它……），还可以通过使用版本控制工具来记录配置文件的历史。

第 2 章，“集中管理网络资源：Kerboros、LDAP 和 NFS”，包含了可以使 LDAP 和 Kerboros 流畅运行的技巧，以及一些添加 NFS 的提示。

第 3 章，“监控与升级”，讲解了如何设置 Nagios 来监控服务器，以及如何设置 Puppet 来简化配置。

第 4 章，“系统备份与数据管理”，如何适当地设置备份系统，以及如何在数据库崩溃时快速恢复。最后，读者还将学会当备份不可用时，如何从坏掉的硬盘中拯救数据。

第 5 章，“文件系统管理”，介绍了在运行过程中编辑和重新设置文件系统大小的方法，以及如何发挥 RAID 的功效。

第 6 章，“保护您的系统”，讲解了 SSH、密钥策略、sudo、Kerberos 和 Apache 等与系统安全相关的部件。在入侵者入侵之前，要自行测试密码强度。

第 7 章，“用 Apache 进行工作”，讲解了使用主流 Web 服务器时的一些技巧。

第 8 章，“更好地使用命令行”，讲解了万能的 bash 知识，以及 find 和 xargs 的快速参考知识。find 和 xargs 都是相当不错的工具，但是如果不熟悉它们的使用，则 Linux 的帮助页面也帮不了您。

第 9 章，“文件中的文本操作”，讲解了 less、sed、awk 和 Perl 相关的知识，它们都可以用于处理文本文件。本章讲解的最后一个技巧是从其他文件中获得文本。

第 10 章，“信息输入和输出”，讲解了键盘映射、打印机和远程桌面的相关知识。

第 11 章，“系统故障追踪”，讲解了用于追踪故障的最佳做法和最佳工具。系统管理员通常需要花费大量的时间来追踪故障，因此要尽可能地让这个过程顺利进行，而不要增加诸多烦恼。

第 12 章，“时间管理和人员管理”，讲解了时间管理和人员管理的知识。不知道到底是幸运还是不幸，我们并不是将所有的时间都用在与计算机打交道上。当您是一线的用户支持人员时，考虑如何与同事相处将会让工作随着时间的日积月累发生显著变化。同样，在多个需求之间管理您的时间将会是一个巨大的挑战，

但是对您的工作非常重要。

版式说明

以#开头的行表示根用户的命令提示符。代码衔接符表示在 shell 或文本文件中本来是一行，但是为了适应本书的版式，而分到了多行之中。

先决条件

本书中的疑难处理技巧都在 Debian 5.0（lenny）和 Ubuntu 9.04（Jaunty Jackalope）中测试并通过。在本书编写之时，这两个版本都是稳定的发行版本。其他 Linux 版本可能在文件名称上和系统问题的处理方式上有微小差别，我已经标记出了某些区别，但是其他区别则需要读者仔细辨别并标记。这些技巧背后的基本理论可用于当今任何 Linux 发行版本。本书给出了 Linux 的软件版本（在编写本书时，所使用的软件版本一般是 Debian 5.0 和 Ubuntu 9.04 版本）。

代码下载

读者可以从 http://www.apress.com 网站下载与本书相关的代码（配置文件和脚本），并查看本书的勘误表，其地址为 http://www.apress.com/9781430224495。

联系作者

读者可以通过 juliet@earth.li 或者 http://the.earth.li/~juliet 联系到作者。作者平时为 http://www.serverwatach.com 和 http;//www.linuxplanet.com 两家网站以及 Linux Format 和 Linux Pro Magazine 杂志写作。

目　录

第 1 章　节省您的精力

这一章涵盖了一个系统管理员最本质的特性：懒惰。总地来说，节省的时间和精力越多，就会有更多的时间去做其他事情：在等待要做下一件事情时，不管是修复问题，还是安装更多的系统，或者是网上冲浪，您都可以做。从长远来看，说明文档、好的脚本实践以及版本控制都可以用来节约时间（有时会节省很多），所以事先花些时间来学习它们是值得的。

1.1　说明文档：要知道它是一件好东西

有一个关于说明文档的问题（尤其是激励自己真正去写）是，许多人很容易认为自己可以准确地记住那些为修复某个特定问题而做的事情。一旦为某个问题花费几小时或者几天，那么所做的一切就会很清晰地放在头脑里，似乎非常难忘记。所以，大家不会停下来用文档把它记录下来，而是匆匆地奔向了下一个问题。

不幸的是，您无论如何乐观地认为，都不会拥有那么好的记忆力（对不起！）。在 6 个月内，当其他类似的事情，甚至同样的事情发生在不同的机器上时，您会意外地发现只剩下似曾相识的感觉，可能只记得一两个细节。

甚至当正在研究问题时，很容易就不知道当前做到哪里了。短时间内做了大量的修改，更改了很多因素，希望可以得到更多的信息或者找到一个解决的办法，但是接着您就可能忘记了刚刚做了哪些更改，在这之前更改了什么，为什么做这些更改，以及是否在每次更改后合理地测试过。

因此，为了正在进行的工作或者为将来留一份记录，合理编写说明文档是值得的，即使只是为了方便您自己。

1.2　说明文档：记录正在进行的工作

关于说明文档首先要做的就是在工作的时候记录所做的一切。这个技巧假定用户正在使用 bash 以及非常标准的 Linux 安装。

用户在积极地解决问题的时候，尽量编写至少一个简短的记录。它用来记录用户所尝试的一切。纸质的记事本在这里会非常有用。这不但能帮助用户记录下所做的尝试，而且有时写东西也会产生一些新的认识。如果用户不知道做到哪里了或者已经做了什么，那么在 bash 的历史记录中回找（或者查看～/.bash_history）将会提醒您。用户可以通过更改几个历史记录设置使得回找更容易，提供更多的信息。将下面这些行添加到～/.bashrc 中：

```
01 shopt -s histappend
02 PROMPT_COMMAND='history -n;history -a'
03 HISTSIZE=100000
04 HISTFILESIZE=100000
```

第 1 行修复了用户打开多个终端窗口，信息可能丢失的问题。发生这种情况的原因是 bash 的默认行为是根据会话来覆盖历史记录的，而不是在后面追加。

第 2 行将 bash 历史记录行为扩展成多个终端可以实时追加。PROMPT_COMMAND 设置在每一个提示符处执行给定的命令。这里意味着在用户每一次按回车键的时候，shell 向历史记录里写，然后再读取历史记录文件（默认的行为是只有当用户关闭终端的时候，才会将这个终端的历史记录写入文件中）。记住，这意味着当您往回查看历史记录时（例如，用向上箭头键或者快捷键），所访问的是在那一台机器上的最后一个命令，而不是在那一个终端窗口的最后一个命令。

第 3 行和第 4 行增加了历史记录保存的条数，默认是 500 条命令。这 500 条命令很快就会被用完，由于这个限制，用户很可能会丢失他们想保留的信息。如果愿意的话，用户可以增加这个数字。

如果事情已经变得混乱，并且用户也不想查看～/.bash_history，那么可以试