Osborne 计算机专业技术丛书

NETWORK PROFESSIONAL'S LIBRARY

Linux File Systems

Linux 文件系统

Moshe Bar 著
天宏工作室 译

- 全面了解Linux 2.4内核可以使用的所有文件系统
- 有效地使用文件系统，包括ext2FS、UFS、UDF、HFS、HPFS、VFAT等
- 使用虚拟文件系统（VFS）、逻辑卷管理器（LVM）以及日记文件系统（JFS）
- 在标准Linux发布内核中实现未预编译的文件系统

清华大学出版社
http://www.tup.tsinghua.edu.cn

Osborne 计算机专业技术丛书

Linux 文件系统

[美] Moshe Bar 著

天宏工作室 译

北京·清华大学出版社

Linux 文件系统
Moshe Bar: **Linux File Systems**
EISBN: 0-07-212955-7

北京市版权局著作权合同登记号　图字 01-2002-0556 号

书　　名：Linux 文件系统
作　　者：Moshe Bar 著
出 版 者：清华大学出版社（北京清华大学学研大厦，邮编 100084）
http://www.tup.tsinghua.edu.cn
http://www.tup.com.cn
责任编辑：冯志强
印 刷 者：清华大学印刷厂
发 行 者：新华书店总店北京发行所
开　　本：787 × 960 1/16　**印张**：22.5　**字数**：492 千字
版　　次：2003 年 3 月第 1 版　2003 年 3 月第 1 次印刷
书　　号：ISBN 7-89494-028-3
印　　数：0001 ~ 3000
定　　价：45.00 元

作者简介

Moshe Bar是 *Linux Internals* 的作者和 Byte.com 及 Dr. Dobbs 的 Linux 高级编辑。他是 Linux 的 JFS 文件系统项目的一位投稿者，还是 Linux 群集的 Mosix 项目的主要投稿者。Moshe Bar 拥有计算机科学的博士学位。

致谢

编写本书不仅仅需要我的努力，还需要我的另一半的努力。有很多次，这本书都影响了我们在一起，她对此非常有耐心。非常感谢她对我的支持。

要特别感谢开放源代码项目 XFS、JFS 以及 ReiserFS 的贡献者。他们的数量太多了，无法在这里一一提及，但是他们当然包括 Stephen Lord 和 Jim Mostek，他们同意向我提供 XFS 的文档。IBM 公司的 Steve Best 非常耐心地回答了许多有关 JFS 的问题。Hans Reiser 提供了这个项目的基本文档，条件是我向他返回本书经过编辑的 ReiserFS 章节，作为 ReiserFS Web 站点的文档。

还要非常感谢我的朋友 Roberto、Tom 和 Barak 教授，他们提供了有洞察力的意见、建议和好的心情。

本书的内容很多都依赖于支持 Linux 及其文件系统的一些专家的帮助。Wietse Venema、Stepehn Tweedie、Andrea Arcangeli、Ingo Molnar、Remy Card 以及其他人通过他们自己的工作以及不知疲倦地积极解释问题提供了巨大的帮助。

其他杰出的开放源代码群体成员包括 Neil Brown、Alessandro Rubini、Doug Gilbert、Louis-Dominique Dubeau 以及其他许多人，他们对本书中讨论的子系统的研究和文档说明为本书提供了巨大的帮助。

最重要的是，我要感谢上帝为我提供了编写本书所需的一切条件。

目录

第 1 章

简　介

我使用 Unix 已经将近 20 年，而且成为 Linux 用户也有 8 年的历史了，因此我认为自己完全了解 Linux 如何存储和检索数据。虽然如此，但我还是无法相信 2000 年秋天在荷兰召开的一个 Unix 会议上所听到的。Wietse Venema（TCPWrappers 的作者以及 MTA[①] 后缀背后的策划者）是开幕式上的演讲者，他宣布自己从标准的 Linux 文件系统恢复了一度认为已经丢失的文件，这个消息使听众目瞪口呆[②]。

我一直认为一旦删除了文件，文件就将永远丢失——无法再次取消删除。Wietse Venema 指出，事实上恢复文件是相当容易的，这本身就是一个严重的安全问题。我和数百个听众都意识到实际上自己只掌握了文件系统的很少一部分知识，而我们正是将宝贵的数据存储在了这种文件系统上。

1.1 Gnu/Linux 和文件系统

在 Linux 的早期，它完全是在 Minix[③] 操作系统下交叉开发的。Linus Trovald 选择这种策略是因为这允许它在两个系统之间共享磁盘，而不需要从头创建一个新的文件系统。Minix 文件系统是一种有效的、错误相对较少的软件。但是，Minix 文件系统设计中的限制条件太苛刻了，因此人们开始考虑并着手在 Linux 中实现新的文件系统。

为了使新的文件系统可以更容易地添加到 Linux 内核中，人们开发了一个虚拟文件系统（Virtual File System，VFS）层。在集成到 Linux 内核中之前，VFS 层最初是由 Chris Provenzano 编写的，后来又由 Linus Torvalds 改写。本书的第 4 章说明了这一点。

将 VFS 层集成到内核中之后，1992 年 4 月实现了一个名为扩展文件系统（Extended File System，ext）的新文件系统，并添加到 Linux 0.96 中。这个新的文件系统消除了 Minix 两个大的限制：它的最大值是 2GB，最长的文件名是 255 个字符。虽然 Ext 文件系统已经在 Minix 文件系统的基础上有了很大的改进，但是它仍然存在一些固有的缺点。它不支持单独访问、信息节点（inode）修改以及数据修改时间戳[④]。文件系统使用内核中的链接列表来跟踪文件系统中的可用块和信息节点，但

① MTA 是一个邮件传输代理（Mail Transportation Agent），发送和接收电子邮件需要使用它。

② 在删除文件时，Linux ext2 文件系统不会删除目录项目，这样就保留了名称-信息节点映射，至少在覆盖该目录项目之前是这样。因此，如果你能够访问原始的目录块，则可以查找已删除文件的名称以及它们的信息节点编号，这样你就可以访问大约前 10 个直接块。

③ Minix 及其文件系统是由 Andrew Tanenbaum 教授为教学目的而开发的。

④ 我们将在后面的章节中介绍这些特性。

是它也引入了新的缺点：随着文件系统使用的增多，这些列表变得很混乱，文件系统将会碎片化，这导致总体文件系统性能的降低。

为了解决这些问题，1993 年初发布了两个新的文件系统：Xia 文件系统（Xiafs）和第二扩展文件系统（或 ext2fs）。Xia 文件系统主要以 Minix 文件系统代码为基础，只是添加了一些基本的改进。Xia 文件系统提供了长文件名，支持更大的分区，并且支持三种时间戳：创建时间、修改时间和访问时间。另一方面，ext2fs 以 ext 文件系统代码为基础，它包含了许多改进。一开始设计它就是为了允许将来进行改进。

最初发布这两个新的文件系统时，虽然它们具有不同的实现方式和源代码，但是它们实际上提供了相同的特性集。由于 Xiafs 的设计是最小的，因此事实上它比 ext2fs 更稳定。最终，ext2fs 中的错误得到了修复，并且集成了许多改进和新特性。目前，ext2fs 非常稳定，事实上已经成为了标准的 Linux 文件系统。

除了深入地研究 ext2fs，本书还讨论了目前可用于 Linux 的重要文件系统，研究它们的优点和缺点，并展示如何有效地使用它们。

1.2　本书的目的

编写此书是为了增加对目前可用于 Linux 的最重要文件系统的一般了解。有时，为了能够完全理解文件系统，我们必须查看它们是如何编写的。但本书不是这些文件系统的源代码注释。相反，本书说明了在什么时候有效地使用哪一个文件系统。

下面是主要术语的概述：

▼ 内核是运行在保护模式中并有权访问硬件的特权寄存器的操作系统软件。

- 文件系统是控制块的逻辑集合，这些控制块代表用户或系统数据的独立容器。在提到术语文件系统时，它的意义可能会有点含糊。使用这个术语的一种情况是在指一种特定类型的文件系统（如 ext2fs 或 NFS）时。或者，它也可以用来表示文件系统的一个特定实例，如/usr 或/boot。
- 名称空间是惟一指定的标识符（如文件名）的集合。在一个名称空间中，只能存在文件名的一个实例。通常，名称空间包含在一个目录范围中。

▲ 目录是文件系统维护的一种特殊文件，它包含了一个项目列表。对于用户来说，一个项目显示为一个文件，并根据其符号化项目名称（即用户的文件名）进行访问。

1.2.1　本书的读者

本书是针对系统管理员和网络管理员、开发人员以及容量设计管理员的。本书对拥有全面硬件和软件知识的 Linux 爱好者也有很强的吸引力。

在下面的章节中，系统管理员将学习如何准备内核，以便使用特定的文件系统，哪些文件系统是可用的以及如何正确使用它们。他们还将能够适当地调整文件系统，从而大幅度提高系统的通过量。

开发人员将学习文件系统是怎样影响他们的应用程序的。虽然许多人可能会争论说文件系统对开发人员来说实际上是透明的，但这是错误的。比如，知道文件系统有效地实现了锁定机制可以使程序员或开发人员不必在应用程序中编写这项功能的代码。

1.2.2 阅读本书之前应该了解的知识

很好地理解计算机科学理论的一般概念（特别是在名称空间和 I/O 领域）是有帮助的。另外，如果读者具有使用 Linux 界面的工作经验以及基本系统管理所涉及内容的初步了解，那么这也是非常有帮助的。阅读本书不需要事先了解关于文件系统的知识。

虽然我对本书提供的大多数代码都进行了解释，但是能够阅读 C 程序还是有帮助的。要想更好地了解 C 语言，本书作者总是强烈推荐 C 语言的设计者 Kernighan/Ritchie 编写的 *The C Programming Language* 一书。

1.2.3 本书的内容

本书简要地介绍了 Linux 操作系统以及它的结构。本书还解释了如何重新编译 Linux 内核，对于使用没有预编译到标准 Linux 发布内核中的文件系统来说，这是非常重要的知识。

在介绍了文件系统的一般 Unix 方法之后，我们将通过虚拟文件系统（或 VFS）学习文件系统的 Linux 抽象，然后介绍和解释所有重要的文件系统。

本书没有试图毫无遗漏地说明所有可用的 Linux 文件系统，而是将重点放在最重要的文件系统以及如何有效地使用它们上面。

本书末尾包括了附录，其中介绍了系统调用、GPL 许可证以及其他有用的信息。

1.2.4 阅读本书的方法

最好的方法就是从头到尾地通读本书。阅读了第一遍之后，读者就可以将本书作为日常使用文件系统时的快速参考。

1.3 查找更多信息的位置

可以在 Internet 上获得有关 Linux 文件系统的最新信息。但是这些信息实际上是一些原始资料。收集这些信息、组合它们并将它们用一种适当的格式表达出来并不

是一项很容易的任务。本书最大的价值就在于它提供了相关的、经过研究的、格式化的信息源，没有本书，公众就只能通过 Web 站点、源代码以及文章获得这些信息。

当然，关于内核的一个很好的信息来源就是实现该文件系统的内核源代码。预订本书中提到的所有文件系统的开发人员邮件列表是获得文件系统的所有相关方面的极佳方式。

1.3.1 建议和意见

我们非常欢迎读者提出任何建议和意见，可以将它们发到 moshe _ bar@hotmail.com 或 Moshe Bar c/o McGraw-Hill, Professional Book Group, Two Penn Plaza, New York, NY10121。

1.3.2 开发源代码——一个现代操作系统的本质

毫无疑问，Linux 获得不同寻常的成功的主要原因是 General Public License (GPL)。但是，开放源代码（有时也称为免费软件）的概念实际上已经非常过时了。免费软件的第一个拥护者就是 Free Software Foundation（FSF）的 Richard Stallman。他为自己编写的一些优秀并且现在广为传播的软件设计了 GPL 许可证，最著名的可能就是 Emacs 编辑环境以及 C 和 C++ 的 GCC 编译器。要想查看 GNU 工具的完整列表，请访问站点 www.gnu.org。

Richard Stallman 还在一个完整的 GNU 操作系统（这是一个名为 GNU Hurd 的项目）上工作了很长一段时间。经过了近十年的开发工作，这个操作系统仍然没有实现，但是从这个项目已经开发出许多强大的技术，它们已经在其他操作系统（如 Linux、BSD 等）上得到了应用。

使用开放源代码方法进行开发对操作系统有两个好处：可靠性和性能。在很大程度上，Linux 的可靠性来源于数百（或数千）开发人员仔细地审查代码、改进它、更改它并尝试它。正如 Eric Raymond 在著名的 *The Cathedral and the Bazaar* 一文中指出的："只要有足够多的测试版测试人员以及共同开发人员，将很快发现几乎每一个问题，并由某个人来解决它。"

因此，由于有这么多人注视着 Linux 代码基数，这就组成了一个比任何封闭式软件开发组织可能提供的 QA（Quality Assurance，质量保证）部门都更好的 QA 部门。这反过来又导致了质量更高的软件。

仅仅一个像开放源代码这样的开发模型本身并不能替代正确的设计和编码方法。但是，开放源代码模式远远超过了专有的开发模型。你可以考虑一下 Linux 中的内核改进的例子。

1.4 Linux 的历史

和许多成功的例子一样，Linux 一开始也是一个由需求产生的项目。1991 年，Linus B. Torvalds 还是芬兰赫尔辛基大学的学生，他自己买了一台基于 i386 的计算机，i386 是第一个具有对虚拟内存管理的芯片支持的 Intel CPU。由于 Linus 对 MS/DOS 操作系统不是非常满意，所以他决定在自己的计算机上实现 Minix 操作系统。

很快他就增强了 Minix，以提供学习所需的功能和特性。后来，他认为 Minix 主要是一个用于教学的操作系统，应该从头创建一个操作系统。Torvalds 还认为最重要的一点是使人们可以从 Internet 上免费获得他的新操作系统的源代码，也就是作为开放源代码并使用名称 Linux（Linus 和 Unix 的缩写）。

第一个版本 0.01 于 1991 年 8 月被放在 Internet 上供下载。同年 10 月，Linus 正式宣布 0.02 版本面世了。这个版本已经可以执行一些 Unix 用户常用的程序，如 bash 命令解释程序、GNU gcc 编译器以及其他基本的实用程序，但是没有更多的功能。

很快，由于这个项目开放源代码的性质，并且全世界的人立刻都可以使用源代码，因此许多黑客、计算机迷以及 PC 爱好者都开始查看这些代码并增强它。许多人开始向 Linus 提出建议，Linus 开创了“正式”参考 Linux 源代码开发树。审阅了这些代码建议之后，Linus 拒绝了大多数建议，但是也采纳了一些建议。在第一个产品版本的问世之前，就这样经过了三年多的开发工作。

1994 年 4 月，1.0 版本问世了。本书作者还记得，它仍然会偶尔显示一些很奇怪的行为，但它是相当适用的。Linux 1.0 的特性包括 TCP/IP、SLIP 和打印机支持，并且具有足够的驱动程序来支持当时可用的各种 PC 设备。

在此之后，Linux 的繁荣时代就真正开始了，世界上各个角落的数百万爱好者都开始使用这个系统。在 1992—1993 年，第一个 Linux 发布版本出现了。发布版本是获得完全正常工作的操作系统的主要方式，包括 Linux 内核、X 窗口系统以及应用程序和实用程序（数以百计）的全面软件包。发布版本还包括一个安装程序，它可以准备操作系统的二进制镜像以及启动/关闭脚本，并且保证所有组件都是兼容的，并针对这些组件进行调整。最后但非最不重要的一点是，发布版本还提供了说明文档。目前存在着许多发布版本，市场上最成功的版本是 RedHat、SuSE 和 Caldera。

下面列出了 Linux 稳定上升到它目前作为 IT 市场中的重要力量这一过程中的重要时期：

1991 年 8 月	版本 0.01
1991 年 10 月	正式宣布版本 0.02