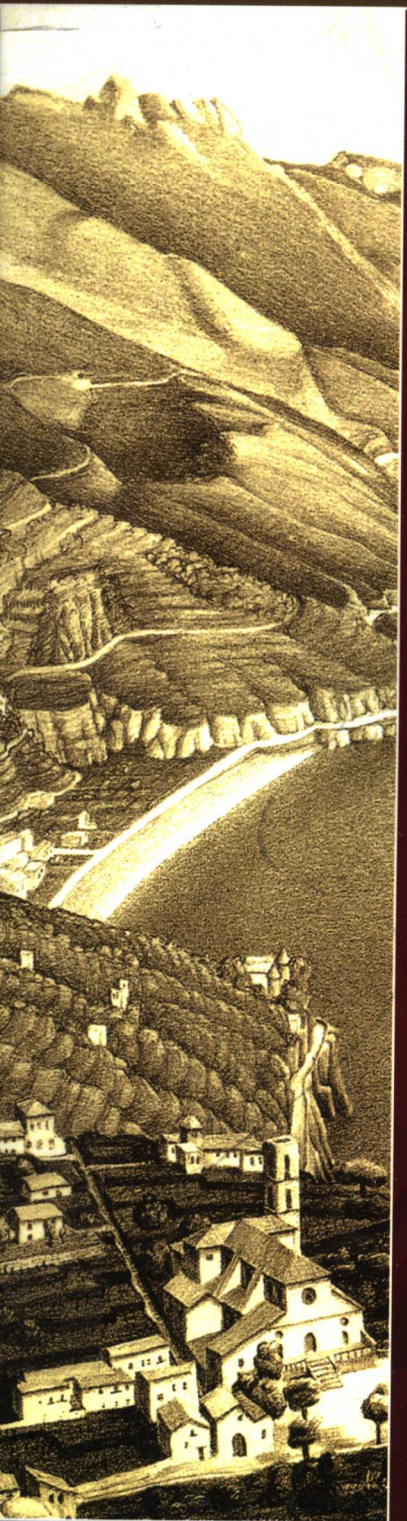
开发大师系列



PRENTICE HALL PTR

Linux Programming by Example

实战Linux编程精髓

[以] Arnold Robbins 著
杨明军 曹亚菲 夏 毅 译

- 最有效的学习方式——使用实例程序介绍 Linux 编程
- 展示富有经验的程序员如何使用 Linux 编程接口
- 循序渐进地介绍高水平编程准则和“内幕”技术
- 详细描述程序性能、可移植性和健壮性

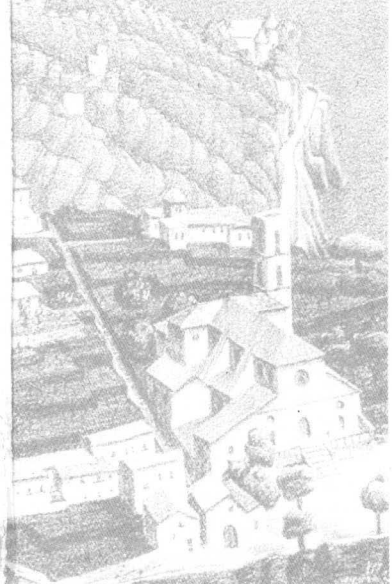
中国电力出版社
www.infopower.com.cn

开发大师系列

Linux Programming by Example

实战Linux编程精髓

[以] Arnold Robbins 著
杨明军 曹亚菲 夏 毅 译



中国电力出版社
www.infopower.com.cn

Linux Programming by Example (ISBN 0-13-142964-7)
Arnold Robbins

北京市版权局著作权合同登记号 图字：01-2005-0828 号

图书在版编目(CIP)数据
实战 Linux 编程精髓 /（以）罗宾斯（Robbins,A.）著；杨明军，曹亚菲，夏毅译.
—北京：中国电力出版社，2005
（开发大师系列）
书名原文：Linux Programming by Example
ISBN 7-5083-3007-2
I.实... II.①罗...②杨...③曹...④夏... III.Linux 操作系统－程序设计 IV.TP316.81
中国版本图书馆 CIP 数据核字（2005）第 012550 号

丛 书 名：开发大师系列
书　　名：实战 Linux 编程精髓
编　　著：（以）Arnold Robbins
翻　　译：杨明军，曹亚菲，夏毅
责任编辑：陈维宁
出版发行：中国电力出版社
　　地址：北京市三里河路 6 号　　邮政编码：100044
　　电话：（010）88515918　　传　　真：（010）88518169
印　　刷：汇鑫印务有限公司
开本尺寸：185×233　　印　　张：33　　字　　数：807 千字
书　　号：ISBN 7-5083-3007-2
版　　次：2005 年 7 月北京第 1 版　　2005 年 7 月第 1 次印刷
定　　价：59.80 元

译 者 序

编写应用软件总会至少涉及到两类接口。一类接口面向最终用户，衡量这类接口的标准主要是看用户使用这些接口是否方便，功能是否丰富，或者简单地说，就是看是否“好用”。另一类接口就是底层库和系统调用接口。这类接口将最终决定你所编写应用软件的功能和性能，因此如何选取这些接口是编码的重点之一。

我们知道，由于历史原因以及其他各种复杂因素，造就了今天 UNIX 的多个衍生版本。这些分支之间存在着各种各样的差异，单从各种 UNIX 系统复杂的衍生关系就可以看出来。Linux 作为 UNIX 界的后起之秀，就必须面对这个现实。它既要继续支持 UNIX 中业已存在的接口，又要对某些实现方法做出决策，以提供优良的性能。因此，它必定与已有的 UNIX 不同。要想在 UNIX/Linux 下编写出优秀的软件，就必须熟悉这些系统调用的方方面面。既要了解这些接口好的一面，也要知道它们的局限性。只有这样，才能在选取接口时做出正确的决策。当然，这是非常困难的一件事情。涉及的内容太多，而且很难找到这方面的资料，书店里基本上没有全面深入介绍这方面内容的书籍。

不过，这本书解决了上述问题。通过阅读这本书，你能够快速地掌握这些重要技术，以构建严谨的 Linux 软件。作者 Arnold Robbins 是一位来自以色列的专业程序员和教授，自 1980 年以来，他就开始使用 C、C++、UNIX 和 GNU/Linux，具有非常深厚的理论基础和丰富的编程实践。他还是一名优秀的作家，其作品有《UNIX in a Nutshell》、《Learning the Korn Shell》和《Effective awk Programming》等。Arnold 是一位令人尊敬的资深 GNU 项目志愿者，目前负责维护 gawk 项目。本书中的许多例子都是直接从这个重要项目的优秀代码中提取出来的，均代表了作者及其资深同行的真知灼见。

本书的特色是讲解深入，力求透彻，不赘述其他书籍中已有的内容。其内容覆盖面广，基本上涉及到了编写应用程序所用到的各个系统调用。通过阅读本书，读者可以全面深入地了解 Linux 编程过程中涉及的各个方面，为实际设计和编程打下良好基础。

翻译国外著名作家的书籍总是非常具有挑战性。他们的书籍往往比较有深度，在内容讲述上也是各有风格，常常是引经据典，这给我们的翻译工作带来了不小的压力。我们本着对读者认真负责的精神，力求做到技术内涵的准确无误和专业术语的规范统一，力求将翻译的准确性和灵活性有效结合。

本书由杨明军、曹亚菲、夏毅等翻译。全书最后由杨明军统稿。在翻译过程中，我们得到了张静、张煜、肖和平、张杰良、王景新、汪东、张英、张明军、许华、李慧霸、王凤芹等的支持，在此表示感谢。**Be Flying 工作室**负责人肖国尊负责在翻译过程中进行相关的协调工作，并控制全书翻译质量和进度。

敬请各位读者就本书提供反馈意见，我们希望通过读者的意见来了解自己的不足，以求在今后译作中更多地和更切实际地考虑读者的需要。读者可以将意见发送到 be-flying@sohu.com。此外，在本书翻译过程中，我们对原书中的一些错误进行了更正，所找出的原书错误及改正情况将在 www.china-pub.com 上贴出，请各位读者关注。

译 者

2004 年 10 月 26 日

前　言

学习编程的一种最好的方式是阅读优良的实例程序。这本书就是通过展示来自大家日常使用的产品级软件中的代码，来介绍基础的 Linux 系统调用 API，这些系统调用 API 构成了一切重要程序的核心。

通过查看具体的程序，你既能够学习怎样使用 Linux API，又能检验在编写软件时出现的一些实际性问题（性能、可移植性和健壮性）。

本书的书名是《实战 Linux 编程精髓》，书中涉及到的所有内容，除非进行了特别说明，都能适用于当前的 Unix 系统。总之，书中用“Linux”来表示 Linux 内核，用“GNU/Linux”表示整个系统（包括内核、库和工具）。同时，我们经常用“Linux”来指代 Linux、GNU/Linux 和 Unix 的全体，如果某一处是特别针对某个或其他的系统，文中会明确地指出。

预期的读者

本书假设读者理解编程，并熟悉基础的 C 语言，至少达到 Kernighan 和 Ritchie 的《The C Programming Language》一书的水平。（希望阅读本书的 Java 程序员则应该理解 C 指针，因为 C 代码中大量地使用了指针。）书中的例子采用了 1990 版本的标准 C 和原始 C（Original C）。

你尤其应该熟悉以下内容：所有的 C 操作符、控制流结构、变量和指针的声明和使用、字符串处理函数、exit()函数的使用以及<stdio.h>中关于文件输入输出的一组函数等等。

你还应该理解标准输入设备、标准输出设备和标准错误设备这几个基本概念，以及了解所有的 C 程序都需要接收一个表示调用选项和参数的字符串数组。另外，应该熟悉基本的命令行工具，如 cd、cp、date、ln、ls、man（有的系统有可能是 info）、rmdir、rm，知道如何使用长命令行选项和短命令行选项、环境变量、输入/输出重定向，包括管道。

我们还假设读者希望编写的程序不仅能运行在 GNU/Linux 下，而且能够适应其他的 Unix 系统。所以，我们对每一个接口都标注了它的适用性（例如，只适用于 GLIBC 系统，或由 POSIX 标准定义等等）。同时，可移植性方面的建议则作为完整的部分在正文中给出。

书中所介绍的编程可能比你以前经历过的要更偏底层一些，事实上就是如此。因为系统调用是上层操作的基础构建部分，所以它们自然处于底层。这也是我们选择 C 语言的原因：因为 API 本来就是为方便 C 语言编程而设计的，而那些向更高层的编程语言（比如 C++和 Java）提供接口的代码也相应地处于底层，所以它们更多地用 C 语言来编写。然而，请注意：“底层”并不意味着“不好”，而仅仅意味着“更多的挑战”。

你能学到什么

本书着重介绍构成 Linux 编程核心的基础 API：

- 内存管理
- 文件输入/输出
- 文件元数据
- 进程和信号
- 用户和组
- 编程支持（排序、参数解析等）
- 国际化
- 调试

我们力图保持这个主题列表的精简。我们相信用单独的一本书就能介绍“所有需要学习的东西”是不大可能的。大部分的读者更青睐选题精辟、重点突出的书。同时，最好的介绍 Unix 的书也都是用这种方式来组织和编写的。

因此，我们不是将所有内容组织在一本大部头书里，而是划分成几册：一册介绍 IPC（Interprocess Communication，进程间通信）和网络编程；另一册介绍软件开发和代码移植。同时我们也关注另外的《实战 Linux 编程》系列，来介绍如线程编程和 GUI 程序设计等的主题。

书中介绍的 API 包括了系统调用和库函数。事实上，在 C 语言的层面上，它们都表现为简单的函数调用。一方面，系统调用是对系统服务的直接请求，如读或写文件、创建进程等。另一方面，库函数则运行在用户态，有可能从来不向操作系统申请任何服务。有关系统调用的文档见参考手册的第 2 节（也可以在线地用 man 命令查看），库函数相关文档在第 3 节。

我们的目标是通过实例来介绍如何使用 Linux API，尤其是尽可能地通过使用早期 Unix 源代码和 GNU 工具来掌握 API。然而，要找到那么多的独立而完备的实例非常困难。所以，我们也编写了相当多的小型的演示程序。我们十分强调编程的规范，特别是在 GNU 编程方面，如“无任何限制（no arbitrary limits）”，它们使得 GNU 工具成为十分优越的程序。

我们有意地选择了最常用的程序。只要曾经使用过 GNU/Linux，你就已经理解了如 ls 和 cp 这样的程序的作用。这样，你就无需花大量的时间去学习它们是做什么的，而可以直接深入地去考查这段程序是如何工作的。

有时，我们同时讲述了实现某项功能的两种方式，即通过上层标准接口和通过底层接口。通常情况下，上层标准接口是通过调用底层接口或底层构件来实现的。我们希望揭示这些“内幕”有助于你理解它们工作的细节。但是在你编写的所有代码中，应该只使用上层的标准接口。

同样，我们有时候介绍一些实现特定功能的函数，然后会劝告（会给出原因）你尽量避免使用它们！这样做的主要原因是，在遇到这些函数的时候，你仍能够辨认它们，进而能够理解调用它们的代码。要全方位地掌握一个主题，你不仅应该知道能做什么，而且更应该明白应该做什么和不应该做什么。

最后，每章都用练习题作为结束。一些题目要求修改或者编写代码，另外一些是更类似于“思考练习”或者“为什么你认为……”这一类的题目。强烈建议你完成所有的练习——它们能够有助于你巩固对正文内容的理解。

短小即精美：Unix 程序

Hoare 定律：

“在每个大型程序内部，都有一个核心小型程序体现其关键行为。”

—C.A.R. Hoare—

最初，我们准备仅仅用 GNU 工具程序中的代码来介绍 Linux API。但是，在当前版本中，即使是简单的命令行程序（如 mv 和 cp）也很大而且呈现多种特征。各种 GNU 标准工具中的程序更是如此：允许长短不等的选项，进行所有 POSIX 标准要求的处理，甚至还经常允许一些额外的看上去与实际处理无关的选项（如高亮输出）。

这样就存在一个合乎情理的问题：“在这么大而混乱的森林中，我们如何关注于一两棵关键的树呢？”换句话来说，如果展示当前的多特征的程序，我们如何可能关注潜在其中的核心操作呢？

Hoare 定律[1]启发我们从早期的 UNIX 程序中寻找示例代码。早期的 V7 Unix 工具程序都非常短小和简单，这样很容易观察其处理过程，以及理解如何使用系统调用。（V7 大约发布于 1979 年，它是所有当前 Unix 系统，包括 GNU/Linux 和 BSD 系统的公共发源）。

许多年以来，Unix 源代码一直受到版权和商业保密许可协议的保护，以至于不可能公开，很难用于学习。所有的商业 Unix 源代码也都是如此。但是，在 2002 年，Caldera（目前的 SCO）使早期的 Unix 代码（从 V7 到 32V Unix）在一个开放源码许可证（参见附录 B，“Caldera Ancient UNIX License”）的保护下可用。因而，我们才能够在这本书中使用早期 Unix 系统中的代码。

标准

全书中，我们涉及了几种不同的正式标准。标准是描述某一事物如何工作的文档。正式标准规定了事物的很多方面，比如对于墙上的电源插座的外观、布局和插孔都由一个正式的国家标准来规定，这样可以保证一个国家内的所有电源线插头都能和电源插座相匹配。

同样，计算机系统方面的正式标准规定了计算机的预期行为。这使得开发者和用户明白他们的软件应该能做些什么，并且，当软件出现故障的时候，可以投诉软件提供商。

与本书相关的标准有：

（1）ISO/IEC International Standard 9899：Programming Languages——C，1990。第一个针对 C 程序设计语言的正式标准。

（2）ISO/IEC International Standard 9899：Programming Languages——C，第二版，1999。第二个（也是目前的）针对 C 程序设计语言的正式标准。

（3）ISO/IEC International Standard 14882：Programming Languages——C++，1998。第一个针对 C++程序设计语言的正式标准。

（4）ISO/IEC International Standard 14882：Programming Languages——C++，2003。第二个（也是目前的）针对 C++程序设计语言的正式标准。

（5）IEEE Standard 1003.1-2001：Standard for Information Technology——POSIX®（Portable

1 这个著名的论述是 1970 年 8 月 10 日至 14 日在波兰 Jablonna 举行的“The International Workshop on Efficient Production of Large Programs”上得出的。

Operation System Interface，可移植操作系统接口）。当前版本的 POSIX 标准：描述了 Unix 和类 Unix 系统的预期行为。这个版本包括了 C 和 C++程序员可见的系统调用接口和库函数接口，用户可见的 shell 接口和工具接口。它包括几个部分：

- Base Definitions。术语、工具性程序和头文件等的定义。
- Base Definitions——Rationale。解释和说明工具性程序被选入或未被选入标准的原因。
- System Interface。包括系统调用和库函数。POSIX 统称它们为“函数”。
- Shell and Utilities。shell 编程语言和在进行 shell 编程时可交互使用的实用工具。

虽然阅读程序语言标准比较乏味，但也许你可能希望购买 C 语言标准的一份拷贝：它给出了该语言的最终定义。这些拷贝可以从 ANSI[2]和 ISO[3]购买。（PDF 版的 C 语言标准十分实惠）。

POSIX 标准可以向 The Open Group[4]订购。从发布目录到“CAE Specifications”下列出的条目，你能够找到针对标准的各个部分的单独页（命名为“C031”到“C034”）。每页都提供了相应部分标准的在线 HTML 版本的免费链接。

POSIX 标准是专门为 Unix 和类 Unix 系统的实现而提出的，它也能适用于非 Unix 系统。因此，它提供的基本功能是 Unix 系统提供的功能的子集。但是，POSIX 标准还定义了可选的扩展部分——辅助功能，比如提供线程和实时方面的支持。其中对我们最重要的扩展部分是 XSI（X/Open System Interface，X/Open 系统接口），它描述了早期 Unix 系统中用到的工具性程序。

我们给全书中的每个 API 都标注了它的针对标准的适用性：只适用于 ISO C、POSIX、XSI、GLIBC 其中一种标准；或者没有标注特定的标准，这时指普遍适用。

多特性且功能更强：GNU 程序

如果我们只是局限于早期 Unix 代码，那么这本书就成了一本有趣的历史书，它在 21 世纪将不是十分有用。当前的程序不再像早期 Unix 系统那样受许多的约束（内存、CPU 功耗、磁盘空间、速度等等）。此外，它们还得适应多种语言的环境——仅 ASCII 码和美国英语明显不够。

更为重要的是，自由软件联盟（Free Software Foundation）和 GNU Project[5]明确提倡的首要的自由之一就是“可自由地学习”。GNU 程序旨在提供一个巨大的优良的程序集合，从而可以提供给编程新手们作为学习资源。

通过使用 GNU 程序，我们力图达到的目标是：向你展示优良的和最新的程序，从中你可以学习到如何编写良好的代码以及如何很好地使用各种 API。

我们觉得 GNU 软件更好，是因为它的自由（free 取意为“自由”，而不是“免费啤酒”中的免费）。同时在技术上，GNU 软件也通常好于对应的 Unix 软件。这一点我们会在第 1.4 节“为什么 GNU 程序更好？”中专门阐述其原因。

大量的 GNU 示例代码取自 gawk（GNU awk）项目。这是因为我们非常熟悉那些代码，从而能更好地从中提取实例。后面我们就不特别声明程序出处了。

---

2 http://www.ansi.org
3 http://www.iso.ch
4 http://www.opengroup.org
5 http://www.gnu.org

各章内容概要

开车是一个全盘的过程，涉及到多个同时发生的动作。Linux 编程在很大程度上也和开车一样，需要理解 API 的多个方面，如文件输入/输出、文件元数据、目录、时间信息的存储等等。

本书的第 1 部分介绍足够多的独立的主题，使你能学习第一个重要的程序——V7 系统的 ls 程序。然后通过考查文件的层次结构、文件系统的工作方式和如何使用文件系统来结束对文件和用户的介绍。

第 1 章，“引言”

描述 Unix 和 Linux 的文件和进程模型；考查原始 C（Original C）和 1990 版本的标准 C 之间的差别；综述使得 GNU 程序通常优于标准 Unix 程序的因素。

第 2 章，“参数、选项和环境”

描述 C 程序是如何访问并处理命令行参数和选项的，并解释如何在环境下工作。

第 3 章，“用户级内存管理”

综述在执行进程中可用的不同种类的内存空间。用户级内存管理是每个重要应用的关键，所以有必要尽早理解它。

第 4 章，“文件和文件 I/O”

讨论基本的文件输入/输出，展示如何创建和使用文件。理解这一主题对学习后面的内容非常重要。

第 5 章，“目录和文件元数据”

叙述目录、硬链接和符号链接的工作原理。然后描述文件元数据，如文件所有者、文件许可等，还包括如何进行目录操作。

第 6 章，“通用库接口——第一部分”

考查第一组通用编程接口，通过它们能够有效地使用文件元数据。

第 7 章，“综合应用：ls”

将前面介绍的内容结合在一起，分析 V7 系统中的 ls 程序。

第 8 章，“文件系统和目录遍历”

阐述文件系统是如何安装和卸载的，以及程序如何辨识已安装的文件系统。同时，描述程序是如何顺利地遍历整个文件层次结构，并在此过程中，遇到每个对象时采取适当的动作。

本书的第 2 部分涉及进程的创建和管理、用管道和信号来进行进程间通信、用户和组标识以及其他的通用编程接口。随后描述 GNU 的 gettext 与国际化；最后讲述几个高级的 API。

第 9 章，“进程管理和管道”

考查进程的创建、程序的执行、采用管道的进程间通信，以及文件描述符管理，包括非阻塞输入/输出。

第 10 章，“信号”

讨论信号，它是进程间通信的一种简单有效的形式。同时，信号在父进程管理其子进程的过程中也扮演了重要的角色。

第 11 章，“权限、用户 ID 号及组 ID 号”

关注如何标识进程和文件，如何进行权限检查，以及 setuid 和 setgid 的工作机制。

第 12 章，“通用库接口——第二部分”

介绍剩下的通用 API，其中许多都要比前面介绍的那组通用 API 更加专用。

第 13 章，“国际化和本地化”

解释如何使你的程序适应于多种语言，同时尽可能减少工作量。

第 14 章，“扩展接口”

描述前面章节介绍过的一些接口的扩展版本，以及详细地讨论文件锁问题。

我们用一章来讨论调试，以完善本书的内容。因为（几乎）没有人能够在第一次就完美地解决问题。之后我们还提出了一个总结性的工程项目，以巩固你对本书中介绍的各种 API 的理解。

第 15 章，“调试”

描述使用 GDB 调试器的基础，尽可能多地把我们的程序设计中的调试经验表述出来，随后介绍几个适用于不同调试过程的有用的工具。

第 16 章，“综合前面所有主题的工程”

给出一个使用到本书中讨论过的所有主题的重要的程序设计项目。

最后，几个附录牵涉到相关的主题，包括本书中使用的源代码的许可证。

附录 A，“十年学会编程”

一句有名的谚语是“罗马非朝夕建成；伟业非一日之功。”同样，不经过长时间的学习和实践，也不能完全理解和精通 Linux/Unix 程序设计。出于这个目的，我们引用了 Peter Norvig 的这篇文章，强烈推荐你读一读。

附录 B，“Caldera 原始 UNIX 许可证”

本书使用的 Unix 源代码的许可证。

附录 C，“GNU 通用公共许可证”

本书使用的 GNU 源代码的许可证。

排版约定

如同所有的计算机相关书籍一样，我们采用特定的排版约定来表述不同信息。我们使用加粗字体来表示要点。用等宽字体表示存在于计算机中的名称，如文件名（foo.c）和命令名（ls，grep）。需要你键入的简短的代码片断使用单引号引起来：‘ls -l *.c’。

$和>是 Bourne shell 的主提示符和从提示符，它们用来展示交互实例。用户输入（User input）会采用和常规的计算机输出（computer output）不同的字体来表示。如下例所示：

```
$ ls -l                    查看文件，选项是数字 1，不是字母 l
foo
bar
baz
```

我们选择了 Bourne shell 和它的变体（ksh93、Bash），而不是 C shell。所以，我们只给出了在 Bourne shell 下呈现的实例。请注意 C shell 的引用和行连接规则是与 Bourne shell 有区别的。如果你

使用 C shell，自己一定要小心！[6]

对于程序中的函数，我们在函数名后面附加了一对空的圆括号：printf()，strcpy()。对于手册页（通过 man 命令访问），我们遵循标准 Unix 约定，标准形式是将命令或函数名用斜体字表示，节号放在圆括号中紧跟在其后：*awk*(1)，*printf*(3)。

如何获得 Unix 和 GNU 源代码

你也许希望得到本书中我们使用的程序的拷贝，以便自己进行试验和复习。所有的源代码都可以从因特网上得到，另外，你的 GNU/Linux 发布中也包含了 GNU 工具程序的源代码。

Unix 代码

TUHS（The UNIX Heritage Society，UNIX 继承协会）负责维护多种“古老”版本 Unix 的存档资料，其网址是：http://www.tuhs.org。

最令人兴奋的是，可以通过网页在线浏览早期 Unix 源代码资料，从这个链接开始访问：http://minnie.tuhs.org/UnixTree/。本书采用的所有的 Unix 实例代码来自于 Seventh Edition Research UNIX System，也就是大名鼎鼎的“V7”。

TUHS 站点本身位于澳大利亚，世界各地还有许多对这些资料的镜像站点——参见 http://tuhs.org/archive_sites.html。这个网页还指出可以使用 rsync 来镜像这些资料。（rsync 是 GNU/Linux 系统的标准部分，如果你没有 rsync，参见 http://rsync.samba.org/。）

你需要大约 2G～3G 字节的磁盘空间来存放所有资料的拷贝。拷贝时，创建一个空目录，然后在该目录下，运行如下命令：

```
mkdir Applications 4BSD PDP-11 PDP-11/Trees VAX Other

rsync -avz minnie.tuhs.org::UA_Root .
rsync -avz minnie.tuhs.org::UA_Applications Applications
rsync -avz minnie.tuhs.org::UA_4BSD 4BSD
rsync -avz minnie.tuhs.org::UA_PDP11 PDP-11
rsync -avz minnie.tuhs.org::UA_PDP11_Trees PDP-11/Trees
rsync -avz minnie.tuhs.org::UA_VAX VAX
rsync -avz minnie.tuhs.org::UA_Other Other
```

你可能希望忽略拷贝 Trees 目录，其中包含几个衍生版本的 Unix，它们占据大概 700MB 的磁盘空间。

你还可以通过查阅 TUHS 的邮件列表，查看在你附近是否有人可以提供 CD-ROM 形式的资料拷贝，从而避免在因特网上传输如此大的数据。

澳大利亚的 Southern Storm Software 公司已经成功将一部分 V7 用户层代码进行了修改和升级，使它们可以在目前的系统上编译并运行，特别是这些系统也包括了 GNU/Linux。这些代码可以从这个公司的网站[7]下载。

---

6 查看 *csh*(1)和 *tcsh*(1)手册页，或参考《Using csh & tcsh》，Paul DuBois 著，O'Reilly&Associates，Sebastopol，CA，USA，1995。ISBN：1-56592-132-1。

7 http://www.southern-storm.com.au/v7upgrade.html

有趣的是 V7 代码本身不含任何版权和许可声明。其设计者主要是出于自己研究的目的而编写这些代码的，而将许可证问题移交给了美国电话电报公司的版权维护部门。

GNU 代码

如果你正在使用 GNU/Linux，该系统的发行版中已经包括了源代码，它们可能被打包成不同格式的文件（Red Hat 的 RPM 文件、Debian 的 DEB 文件、Slackware 的 `.tar.gz` 文件，等等）。本书中的许多例子都取自 5.0 版本的 GNU Coreutils。要获得这些代码，只要找到相应的 GNU/Linux 发行版光盘，然后用合适的工具软件进行解压提取即可。你也可以按照下面几段给出的指令来获得代码。

如果你想自己从 GNU 的 `ftp` 网站获取那些文件，可以链接这个地址：ftp://ftp.gnu.org/gnu/coreutils/coreutils-5.0.tar.gz。

你可以通过 `wget` 工具程序来取得文件：

```
$ wget ftp://ftp.gnu.org/gnu/coreutils/coreutils-5.0.tar.gz      取得发行版
……这里输出大量的已获取到的文件……
```

另外，你也可以用经典的 `ftp` 命令来获取文件：

```
$ ftp ftp.gnu.org                                          链接 GNU ftp 站点
Connected to ftp.gnu.org (199.232.41.7).
220 GNU FTP server ready.
Name (ftp.gnu.org:arnold): anonymous                       采用匿名方式登录 ftp
331 Please specify the password.
Password:                                                  密码不会回显在屏幕上
230-If you have any problems with the GNU software or its downloading,
230-please refer your questions to <gnu@gnu.org>.
...                                                        省略了许多冗长的登录信息
230 Login successful. Have fun.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /gnu/coreutils                                     进入 Coreutils 目录
250 Directory successfully changed.
ftp> bin
200 Switching to Binary mode.
ftp> hash                                                  输出正在处理指示符 #
Hash mark printing on (1024 bytes/hash mark).
ftp> get coreutils-5.0.tar.gz                              获取文件
local: coreutils-5.0.tar.gz remote: coreutils-5.0.tar.gz
227 Entering Passive Mode (199,232,41,7,86,107)
150 Opening BINARY mode data connection for coreutils-5.0.tar.gz (6020616 bytes)
##########################################################################################
##########################################################################################
...
226 File send OK.
6020616 bytes received in 2.03e+03 secs (2.9 Kbytes/sec)
ftp> quit                                                  注销
221 Goodbye.
```

获得文件后，按如下方式进行解压提取：

```
$ gzip -dc < coreutils-5.0.tar.gz | tar -xvpf -            解压缩文件
……这里输出大量的解压得到的文件……
```

采用 GNU tar 的系统可能要使用下面的命令解压：

```
$ tar -xvpzf coreutils-5.0.tar.gz                    解压缩文件
……这里输出大量的解压得到的文件……
```

为了遵守 GNU 通用公共许可证（GNU General Public License），这里列出本书引用的所有 GNU 程序的版权信息。所有程序都是“自由软件；在自由软件联盟（Free Software Foundations）发布的 GNU 通用公共许可证第二版或（你愿意的话）任何更新版本的保护下，你可以重新发布和/或修改这些程序。”GNU 通用公共许可证的正文参见附录 C，“GNU 通用公共许可证”。

Coreutils 5.0 文件	版 权 时 间
lib/safe-read.c	Copyright © 1993－1994，1998，2002
lib/safe-write.c	Copyright © 2002
lib/utime.c	Copyright © 1998，2001－2002
lib/xreadlink.c	Copyright © 2001
src/du.c	Copyright © 1988－1991，1995－2003
src/env.c	Copyright © 1986，1991－2003
src/install.c	Copyright © 1989－1991，1995－2002
src/link.c	Copyright © 2001－2002
src/ls.c	Copyright © 1985，1988，1990，1991，1995－2003
src/pathchk.c	Copyright © 1991－2003
src/sort.c	Copyright © 1988，1991－2002
src/sys2.h	Copyright © 1997－2003
src/wc.c	Copyright©1985，1991，1995－2002
Gawk 3.0.6 文件	**版 权 时 间**
eval.c	Copyright © 1986，1988，1989，1991－2000
Gawk 3.1.3 文件	**版 权 时 间**
awk.h	Copyright © 1986，1988，1989，1991－2003
builtin.c	Copyright © 1986，1988，1989，1991－2003
eval.c	Copyright © 1986，1988，1989，1991－2003
io.c	Copyright © 1986，1988，1989，1991－2003
main.c	Copyright © 1986，1988，1989，1991－2003
posix/gawkmisc.c	Copyright © 1986，1988，1989，1991－1998，2001－2003
Gawk 3.1.4 文件	**版 权 时 间**
builtin.c	Copyright © 1986，1988，1989，1991－2004

GLIBC 2.3.2 文件	版 权 时 间
locale/locale.h	Copyright © 1991，1992，1995－2002
posix/unistd.h	Copyright © 1991－2003
time/sys/time.h	Copyright © 1991－1994，1996－2003
Make 3.80 文件	**版 权 时 间**
read.c	Copyright © 1988－1997，2002

获取本书中采用的示例程序

本书中采用的实例程序可以在 http://authors.phptr.com/robbins 上找到。

关于封面

“这是杰迪武士的武器……，是来自更文明世纪的精良武器。
在暗黑时代来临前，在帝国创立前的一千多代人的时间里，
杰迪武士一直是古代共和社会的和平和正义的守护神。”

－Obi-Wan Kenobi－

你也许觉得奇怪，我们为什么选择一把光剑作为封面，并在整本书中都用到它？它代表什么意思？它和 Linux 编程有什么关系吗？

在杰迪武士手中，这把光剑不仅强有力，而且非常高雅。它诠释着力量、知识、对权力的控制以及 Jedi 承受过的残酷的训练。

这把精良的佩剑也反映了早期 Unix API 程序设计的高雅。同样，对 API、软件工具、GNU 设计准则的准确地和充满智慧地使用，促成了今天的功能强大、灵活、高效的 GNU/Linux 系统。这个系统也充分证明了编写这些系统组件的程序员的知识和技能。

当然，光剑也非常酷！

致谢

编写一本书要花费大量的精力，如果没有许多对本书做出贡献的人，本书是不可能完成的。Brian 博士、W.Kernighan、Doug McIlroy 博士、Peter Memishian 和 Peter van der Linden 审阅了本书的初稿。David J.Agans、Fred Fish、Don Marti、Jim Meyering、Peter Norvig 和 Julian Seward 对全书中众多引用的主题提供了转载许可。感谢 Geoff Collyer、Ulrich Drepper、Yosef Gold、C.A.R（Tony） Hoare 博士、Manny Lehman 博士、Jim Meyering、Dennis M.Ritchie 博士、Julian Seward、Henry Spencer 和 Wladyslaw M.Turski 博士，他们为本书提供了许多有用的信息。也要感谢 GNITS 组的其他成员：Karl Berry、Akim DeMaille、Ulrich Drepper、Greg McGary、Jim Meyering、François Pinard 和 Tom Tromey，他们提供了许多实用的编程实践方面的反馈。还有 Karl Berry、Alper Ersoy 和 Nelson H.F.Beebe 博士提供了很多关于 Texinfo 和 DocBook/XML 系列工具的有用的技术信息。

优秀的技术审阅者不仅确保了作者所述内容的正确性，而且还确保了作者仔细考虑其表达方式。整本书的技术审阅者有：Nelson H.F.Beebe 博士、Geoff Collyer、Russ Cox、Ulrich Drepper、Randy Lechlitner、Brian W.Kernighan 博士、Peter Memishian、Jim Meyering、Chet Ramey 和 Louis Taber。此外，Michael Brennan 博士为第 15 章做了有益的评注。本书的行文和许多实例程序都获益于他们的审校。所以，我非常感激他们。如同大部分作者所说的一样："书中任何遗留错误的责任都在我。"

我还要特别感谢培生教育出版集团（Pearson Education）的 Mark Taub，是他提出了该书的计划，并对整个系列表示了热情的关注，而且，在编写本书的不同阶段他给出了不少帮助和建议。Anthony Gemmellaro 为实现我对封面的想法做了出众的工作，Gail Cocker 对书的内部设计非常美观。Faye Gemmellaro 的工作使得整个出版过程一点也不杂乱，而是十分愉快。Dmitry Kirsanov 和 Alina Kirsanova 进行了图表和页面的版面设计，还完成了本书的索引。和他们一起合作是非常愉快的事。

最后，我要把深深的感激和爱献给我的妻子，Miriam，在这本书的编写过程中，是她给了我最大的支持和鼓励。

Arnold Robbins
Nof Ayalon
于以色列

目　录