

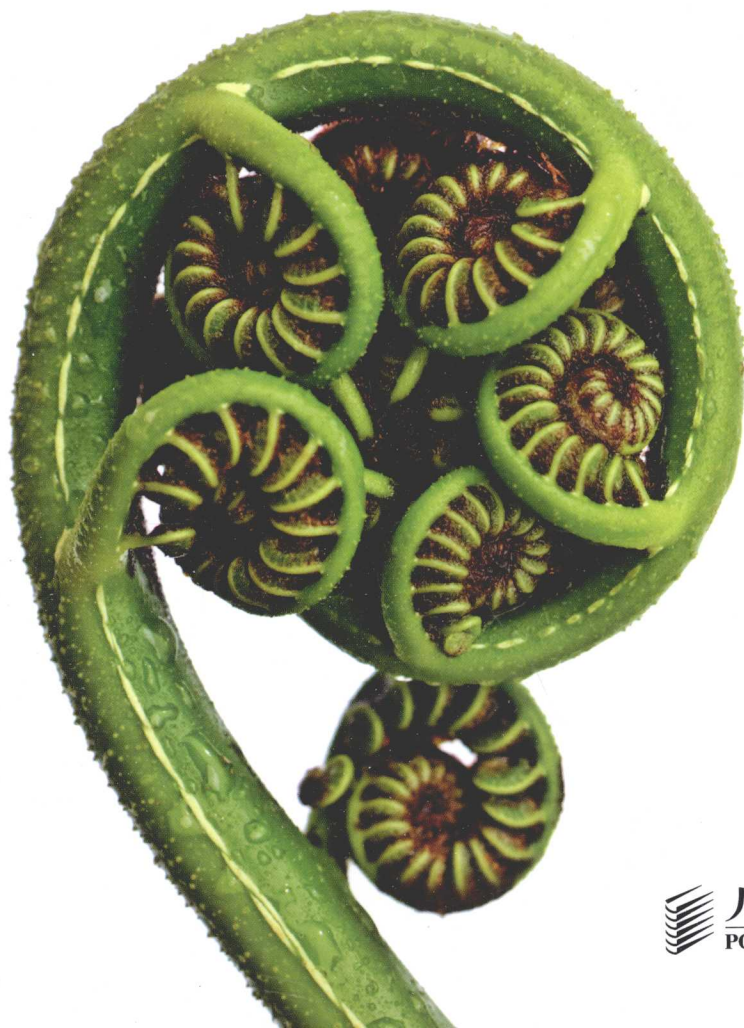
Linux/UNIX

系统编程手册 (上册)

THE **LINUX**
PROGRAMMING
INTERFACE

A Linux and UNIX[®] System Programming Handbook

[德] Michael Kerrisk 著
孙剑 许从年 董健 孙余强 译



 人民邮电出版社
POSTS & TELECOM PRESS

014017737

TP316.89-62

05

V1

Linux/UNIX 系统编程手册 (上册)

THE LINUX
PROGRAMMING
INTERFACE

[德] Michael Kerrisk 著

孙剑 许从年 董健 孙余强 译

A Linux and UNIX[®] System Programming Handbook



TP316.89-62

05

V1



北航

C1704660

人民邮电出版社

北京

图书在版编目(CIP)数据

Linux/UNIX系统编程手册：全2册 / (德) 凯利斯克
(Kerrisk, M.) 著；孙剑等译. — 北京：人民邮电出版
社，2014.1

ISBN 978-7-115-32867-0

I. ①L… II. ①凯… ②孙… III. ①Linux操作系统
—程序设计—手册②UNIX操作系统—程序设计—手册
IV. ①TP316.89-62②TP316.81-62

中国版本图书馆CIP数据核字(2013)第187440号

版权声明

Copyright © 2010 by Michael Kerrisk. Title of English-language original: The Linux Programming Interface, ISBN 978-1-59327-220-3, published by No Starch Press. Simplified Chinese-language edition copyright © 2014 by Posts and Telecom Press. All rights reserved.

本书中文简体字版由美国 No Starch 出版社授权人民邮电出版社出版。未经出版者书面许可，对本书任
何部分不得以任何方式复制或抄袭。

版权所有，侵权必究。

-
- ◆ 著 [德] Michael Kerrisk
译 孙 剑 许从年 董 健 孙余强 郭光伟
陈 舸
责任编辑 傅道坤
责任印制 程彦红 杨林杰
- ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
- ◆ 开本：787×1092 1/16
印张：76.25
字数：1 618 千字 2014 年 1 月第 1 版
印数：1-3 500 册 2014 年 1 月北京第 1 次印刷

著作权合同登记号 图字：01-2010-3829 号

定价：158.00 元（上、下册）

读者服务热线：(010)81055410 印装质量热线：(010)81055316

反盗版热线：(010)81055315

广告经营许可证：京崇工商广字第 0021 号

对本书的赞誉

编写 Linux 软件时如果只能选择一本参考书，则非本书莫属。

——MARTIN LANDERS, Google 公司软件工程师

本书描述精到，示例周详，涵盖了 Linux 底层 API 编程的详尽内容及个中细微之处——无论读者水平如何，都能从本书中受益。

——MEL GORMAN, *Understanding the Linux Virtual Memory Manager* 作者

Michael Kerrisk 的这本 Linux 编程巨著，不但论及 Linux 编程、其与各种标准之间的联系，而且还就作者所知，重点介绍了已获修正的 Linux 内核 bug 以及改进颇多的 Linux 手册页。凭此三点，足可让 Linux 编程更易上手。本书对各项主题的深入探讨使其成为必备的参考书籍——无论读者在 Linux 编程方面造诣如何。

——ANDREAS JAEGER, NOVELL 公司 OPENSUSE 项目经理

Michael 用他坚忍不拔的毅力为 Linux 程序员奉献了这本论述严谨、表述清晰、简洁的权威参考书。虽然本书针对 Linux 程序员而著，但对任何在 UNIX/POSIX 环境中编程的程序员来说都极具价值。

——DAVID BUTENHOF, *Programming with POSIX Threads* 作者、POSIX/UNIX 标准撰写者

本书在重点关注 Linux 系统的同时，对于 UNIX 系统和网络编程也阐述透彻，浅显易懂。无论是初涉 UNIX 编程的新丁，还是编程经验丰富的 UNIX 老手（想要了解大行其道的 GNU/Linux 系统有何新意），我都向他们力荐此书。

——FERNANDO GONT, 网络安全研究员、IETF 参与者、IETF RFC 作者

本书以百科全书般的叙述风格对 Linux 接口编程做了既深且广的覆盖，还提供了大量教科书风格的编程示例和练习。本书所包含的各项主题——从原理到可以实际运行的代码——都已描述清晰且易于理解。本书正是专业人士、学生以及教育工作者所期盼的 Linux/UNIX 参考书。

——ANTHONY ROBINS, 奥塔哥大学计算机科学副教授

无论从精确性、质量，还是详细程度来说，本书都令我印象深刻。身为 Linux 系统调用的行家，Michael Kerrisk 与我们分享了他对 Linux API 的认知和理解。

——CHRISTOPHE BLAESS, *Programmation système en C sous Linux* 作者

对于治学严谨的专业 Linux/UNIX 系统程序员而言，本书实为必备的参考书籍。本书涵盖了所有关键 API 的使用，同时兼顾 Linux 和 UNIX 系统接口，描述清晰，示例具体；除此之外，还强调了遵从诸如 SUS 和 POSIX 1003.1 等标准的重要性和益处。

——ANDREW JOSEY, The Open Group 标准部总监、POSIX 1003.1 工作组主席
由手册页的维护者亲自操刀，以系统程序员视角写出一本百科全书式的 Linux 系统编程巨著，还有比这更完美的吗？本书全面而又详实。我坚信本书将在我的书架上牢牢占据一席之地。

——BILL GALLMEISTER, *POSIX.4 Programmer's Guide: Programming for the Real World* 作者
本书是最新最全的 Linux/UNIX 系统编程参考书。无论读者是 Linux 系统编程新兵，还是关注 Linux 编程和程序移植性的 UNIX 系统编程老将，又或者只是在寻找一本 Linux 编程接口方面的优秀参考书的读者，Michael Kerrisk 的这本大作都笃定是其案头良伴。

——LOÏC DOMAIGNÉ, CORPULS.COM 首席软件架构师（嵌入式）

内容提要

本书是介绍 Linux 与 UNIX 编程接口的权威著作。Linux 编程资深专家 Michael Kerrisk 在书中详细描述了 Linux/UNIX 系统编程所涉及的系统调用和库函数，并辅之以全面而清晰的代码示例。本书涵盖了逾 500 个系统调用及库函数，并给出逾 200 个程序示例，另含 88 张表格和 115 幅示意图。

本书总共分为 64 章，主要讲解了高效读写文件，对信号、时钟和定时器的运用，创建进程、执行程序，编写安全的应用程序，运用 POSIX 线程技术编写多线程程序，创建和使用共享库，运用管道、消息队列、共享内存和信号量技术来进行进程间通信，以及运用套接字 API 编写网络应用等内容。

本书在汇聚大批 Linux 专有特性 (epoll、inotify、/proc) 的同时，还特意强化了对 UNIX 标准 (POSIX、SUS) 的论述，彻底达到了“鱼与熊掌，二者得兼”的效果，这也堪称本书的最大亮点。

本书布局合理，论述清晰，说理透彻，尤其是作者对示例代码的构思巧妙，独具匠心，仔细研读定会受益良多。本书适合从事 Linux/UNIX 系统开发、运维工作的技术人员阅读，同时也可作为高校计算机专业学生的参考研习资料。

前 言

Linux 和 UNIX

主题

本书将描述 Linux 编程接口：由 UNIX 操作系统的开源实现——Linux 所提供的系统调用、库函数以及其他底层接口。运行于 Linux 之上的每一个程序都会直接或间接地使用这些接口。这些接口允许应用程序去执行诸多任务：文件 I/O、创建/删除文件和目录、创建新进程、执行程序、设置定时器、在同一台计算机上发起进程或线程间通信，以及为联网计算机间的进程建立通信等等。有时，人们也将这一系列的底层接口称为系统编程接口。

尽管本书着眼于 Linux，但对于标准和可移植性问题也倍加关注。对于 Linux 所特有的技术细节，以及已由 POSIX 和 SUS 标准化的 UNIX 普遍特性，本书会在论述中清晰地加以区分。因此，本书也提供了对 UNIX/POSIX 编程接口的全面描述。对于那些在其他 UNIX 系统环境中编程，或者编写跨平台可移植应用的程序员来说，本书同样具有实用价值。

本书的读者

本书主要针对以下读者：

- 为 Linux 系统、其他 UNIX 系统，或兼容于 POSIX 的操作系统编写应用程序的程序员和软件设计人员。
- 在 Linux 和其他 UNIX 实现之间，以及 Linux 和其他操作系统之间进行应用程序移植的程序员。
- 教/学 Linux 和 UNIX 系统编程的高校师生。
- 意欲深入理解 Linux/UNIX 编程接口以及系统软件各模块实现细节的系统管理人员和高级用户（power users）。

作者假定读者之前有些许编程经验，但不必是在系统编程领域。此外，作者还假定读者具备阅读 C 语言源码的能力，并了解如何使用 shell 和 UNIX 或 Linux 的常用命令。对于不熟悉 UNIX 和 Linux 的读者来说，阅读第 2 章中面向程序员对 UNIX 和 Linux 系统基本概念所做的回顾会有所帮助。

提示: [Kernighan & Ritchie, 1988]是最具权威性的 C 语言参考书籍。[Harbison& Steele, 2002]一书对 C 语言的介绍则更为详细,并涵盖了由 C99 标准所带来的变化。[van der Linden, 1994]也是一本不错的 C 语言书籍,寓教于乐。[Peek et al., 2001]则对 UNIX 的使用做了简洁而完整的介绍。

贯穿本书,会以这种缩进小字体的文字形式用于旁注,其内容包括基本原理、实现细节、背景信息、史上轶闻以及与正文相关的其他辅助主题。

Linux 和 UNIX

其他 UNIX 实现的大多数特性同样见诸于 Linux,反之亦然。有鉴于此,本书本可只关注于标准 UNIX(即 POSIX)的系统编程。编写可移植的应用程序固然是值得追求的目标,但描述 Linux 对标准 UNIX 编程接口的扩展也同样重要。Linux 的广受欢迎只是原因之一,而有时出于性能方面的考虑,或是需要访问标准 UNIX 编程接口所不支持的功能时,使用非标准扩展(正因如此,所有 UNIX 实现都提供有非标准扩展)就显得至为重要,此为原因之二。

综上所述,在构思本书时,作者不但力图使其对在各种 UNIX 实现中编程的程序员有所帮助,还全面介绍了 Linux 专有的编程特性,如下所示。

- `epoll`, 获取文件 I/O 事件通知的一种机制。
- `inotify`, 监控文件和目录变化的一种机制。
- `capabilities`, 为进程赋予超级用户的部分权限的一种机制。
- 扩展属性。
- `i-node` 标记。
- `clone()` 系统调用。
- `/proc` 文件系统。
- 在文件 I/O、信号、定时器、线程、共享库、进程间通信以及套接字方面, Linux 所专有的实现细节。

本书的用途和组织结构

本书主要有以下两方面的用途:

- 作为 Linux/UNIX 编程接口的入门教程,读者可循序阅读本书。后续各章内容均构建于之前诸章素材的基础之上,伴之以尽可能简短的前向引用。
- 作为 Linux/UNIX 编程接口的参考大全,读者可以根据书后的详细索引和频现于正文中的交叉引用,随机选择阅读主题。

本书各章可分为以下几个部分。

1. 背景知识及概念: UNIX、C 语言以及 Linux 的历史回顾,以及对 UNIX 标准的概述(第 1 章);以程序员为对象,对 Linux 和 UNIX 的概念进行介绍(第 2 章);Linux 和 UNIX 系统编程的基本概念(第 3 章)。
2. 系统编程接口的基本特性: 文件 I/O(第 4 章、第 5 章),进程(第 6 章),内存分配(第 7 章),用户和组(第 8 章),进程凭证(`process credential`)(第 9 章),时间(第 10 章),系统限制和选项(第 11 章),以及获取系统和进程信息(第 12 章)。
3. 系统编程接口的高级特性: 文件 I/O 缓冲(第 13 章),文件系统(第 14 章),文件

属性（第 15 章），扩展属性（第 16 章），访问控制列表（第 17 章），目录和链接（第 18 章），监控文件事件（第 19 章），信号（signals）（第 20~22 章），以及定时器（第 23 章）。

4. 进程、程序及线程：进程的创建、终止，监控子进程，执行程序（第 24~28 章），以及 POSIX 线程（第 29~33 章）。
5. 进程及程序的高级主题：进程组、会话以及任务控制（第 34 章），进程优先级和进程调度（第 35 章），进程资源（第 36 章），守护进程（第 37 章），编写安全的特权程序（第 38 章），能力（capability）（第 39 章），登录记账（第 40 章），以及共享库（第 41 章和第 42 章）。
6. 进程间通信（IPC）：IPC 概览（第 43 章），管道和 FIFO（第 44 章），系统 V IPC 消息队列、信号量（semaphore）及共享内存（第 45~48 章），内存映射（第 49 章），虚拟内存操作（第 50 章），POSIX 消息队列、信号量及共享内存（第 51~54 章），以及文件锁定（第 55 章）。
7. 套接字和网络编程：使用套接字的 IPC 和网络编程（第 56~61 章）。
8. 高级 I/O 主题：终端（第 62 章），其他 I/O 模型（第 63 章），以及伪终端（第 64 章）。

程序示例

本书会以简短而完整的程序示例来描述大部分编程接口，以期读者通过命令行方便地体验这些程序，从而了解各种不同系统调用和库函数的运作方式。因此，本书包含了大量代码示例——约 15 000 行 C 语言代码和 shell 会话记录。

虽然阅读并执行上述示例代码是学习 Linux 编程接口的一个不错的起点，但巩固本书所述概念最为有效的方式还是动手编写代码——无论是修改示例程序以验证自己的编程思路，还是编写全新的程序。

书中所有源代码均可从本书网站上下载。网站所发布的源码中还包含了不少未见诸于本书的其他程序。这些程序的用途和详细信息在源码注释中均有描述。源码中还提供了 Makefile，用来构建相应的程序，附带的 README 文件则提供了相应程序的具体细节。

在 GNU Affero 通用公共许可证（Affero GPL）版本 3 条款的约束下，可自行重新发布和修改本书源码，源码中也提供了一份 GNU Affero GPL 版本 3 的文件副本。

习题

本书各章大都会在结尾处附有一组习题，其中一部分会利用书中的程序示例进行各种试验，另一些则与本章所讨论的概念相关，而其他习题则是引导读者亲自动手编程，意在巩固读者对所学内容的理解。附录 F 会有选择地给出部分习题的答案。

标准和可移植性

本书自始至终都对可移植性问题予以了特殊关注。对相关标准的引用在书中会反复出现，尤其是 POSIX.1-2001 和 SUSv3 的联合标准¹。此外，本书还包括了该标准最新版本（POSIX.1-2008 与 SUSv4 联合标准）的变更细节。（由于 SUSv3 较之于之前的版本做了大范围的修订，并且在写作本书之际，SUSv3 依然是影响最为广泛的 UNIX 标准，故而本书对标准的讨论一般都以

¹ 译者注：大致可视为一套标准的两种称谓。

SUSv3 为框架，并会注明其与 SUSv4 之间的差别。然而，对读者来说，除非另有说明，与 SUSv3 规范有关的论述同样适用于 SUSv4。）

对于那些尚未标准化的特性，本书会列出与其他 UNIX 实现间的差异范围。此外，本书也会强调那些独具 Linux 实现特色的主要特性，以及 Linux 和其他 UNIX 实现之间在系统调用和库函数方面的细微差别。任何特性，凡未注明为 Linux 所专有，读者通常可将其视为大部分或所有 UNIX 实现的标准特性。

书中的编程示例（除了注明为 Linux 所专有的特性）大多已在 Solaris、FreeBSD、Mac OS X、Tru64 UNIX 以及 HP-UX 中的所有或部分系统上进行了测试。为了改进针对其中某些系统的可移植性，本书的 Web 站点还为特定编程示例提供了其他版本，此类代码就不再于本书中列出。

Linux 内核和 C 语言函数库版本

本书主要着眼于 Linux 2.6.x，撰写本书之际，这一内核版本的使用也最为广泛。本书同样涵盖了 Linux 2.4 内核的详细信息，也会适时指明 Linux 2.4 和 2.6 内核间的特性差异。凡是见诸于 Linux 2.6.x 系列中的新特性，作者均会标出其（首度）出现的确切内核版本号（例如，2.6.34）。

至于 C 语言函数库，本书会重点关注 GNU C 语言库（glibc）版本 2。本书也会适时指出 glibc 2.x 版本之间所存在的差异。

本书付梓之际，Linux 内核版本 2.6.35 刚刚问世，不久又发布了 glibc 版本 2.12。本书目前的论述涵盖了以上两种软件的这两个版本。本书出版后，Linux 和 glibc 接口所发生的变化会在本书的 Web 站点上公布。

在其他语言中调用编程接口

虽然本书的程序示例都是以 C 语言编写而成的，但读者也能使用其他编程语言来调用本书所描述的编程接口。这些语言既包括编译语言，例如：C++、Pascal、Modula、Ada、FORTRAN 和 D 语言，也包括脚本语言，例如：Perl、Python 和 Ruby（如要使用 Java，则需另辟蹊径，可参阅[Rochkind, 2004]）。这需要运用不同的技术以获取必要的常量定义和函数申明（C++除外），而按 C 语言链接惯例所约定的方式来传递函数参数可能也需要额外的工作投入。虽然实现起来有差异，但基本原理却都相同，即便读者在使用其他语言进行编程，本书所含信息对他们也同样适用。

关于作者

本人于 1987 年开始使用 UNIX 和 C 语言。当时，作者连续几个礼拜都泡在一台 HP Bobcat 工作站旁，陪伴我的只有 Marc Rochkind 所著 *Advanced UNIX Programming*（第 1 版）一书，以及一本最终被翻得卷了边的 C shell 手册的印刷本。投入时间阅读文档（如果有的话），并编写一些小型的（规模可逐渐变大）测试程序进行试验，直至自己对软件的理解感到信心满满——这是作者当时所采用的编程学习方法，并一直沿用至今——作者也向任何试水新型软件技术的人们推荐这一做法。依作者拙见，从长远来看，这种自学方法能够大大节约时间。本书所载的许多编程示例正是在这一学习方法的激励之下设计而成。

作者的主要身份是软件工程师和设计师。然而，作者同样好为人师，并在学术或商业领域有过数年的教学经验。作者还开设过多门为期一周的 UNIX 系统编程课程，这一经验对本

书的写作也颇有影响。

作者使用 Linux 的时间大约只有与 UNIX 打交道的时时间一半长,在这段时间里,作者的兴趣也逐渐集中在了内核和用户空间的“分水岭”——Linux 编程接口上。这一兴趣也使作者投身于一系列紧密相联的活动中。作者会时不时地对 POSIX/SUS 标准提出自己的意见并提供 BUG 报告;对新加入 Linux 内核中的用户空间接口进行测试和设计评审(还能帮助发现并修复那些接口中的诸多代码和设计缺陷);作者还经常在关于编程接口及其文档的主题会议上发言,并受邀多次出席 Linux 内核开发者年度峰会。将上述所有活动串接在一起的主线是作者对 Linux 领域最突出的贡献:作者为 Linux man-pages 项目 (<http://www.kernel.org/doc/man-pages/>) 所做的工作。

Linux 手册页中的第 2、3、4、5 以及 7 部分都属于 man-pages 项目。这几部分也是手册页中描述编程接口的内容,这些编程接口由 Linux 内核及 GNU C 语言库提供,本书所要介绍的正是这方面的内容。作者参与 man-pages 项目已逾十载。自 2004 年起,作者成为了该项目的主要维护人,所承担的任务大致包括:撰写文档、阅读内核和 C 语言库源码,以及通过编程来验证文档细节(通过为接口撰写文档来发现相关接口中的 BUG,效果颇为不俗)。此外,作者对 man-pages 项目的贡献也最多——在约 900 页的手册页中,作者独自编写了其中的 140 页,并与他人合著了另外的 125 页。因此,在购买本书之前,读者想必已经阅读过本人的工作成果了。作者希望这些成果能对读者有所帮助,希望本书更是如此。

致谢

没有一千人等的支持,本书的质量绝不会如此之高。我要向他们致以最诚挚的谢意。

来自世界各地的多位技术审稿人都参与了本书初稿的阅读,找出错误,指出含糊不清的解释,对措辞、插图以及习题提出建议,测试程序,发现不为作者所知的 Linux 和其他 UNIX 实现间的行为差异,并不时为作者打气助威。在本书中,作者将许多审稿人无私奉献的真知灼见一并收纳,实则作者的知识并非如此渊博。当然,书中的任何错误都是作者一人之过。

无论以下技术审稿人(按姓氏字母排序)对本书手稿的审校巨细与否,篇幅多少,作者都要向他们致以由衷的谢意。

- **Christophe Blaesss** 是一名软件咨询工程师和培训专家,专长是 Linux 在工业(实时和嵌入)方面的应用。Christophe 是 *Programmation système en C sous Linux* 一书的作者,这本法文杰作涵盖了与本书相同的多项主题。他不吝审阅了本书的众多章节。
- **David Butenhof** (HP 公司) 是原 POSIX 线程工作组和 SUS 线程扩展工作组的成员之一,也是 *Programming with POSIX Threads* 一书的作者。他曾为开放软件基金会编写了最初的 DCE 线程参考实现,并曾担任 OpenVMS 和 Digital UNIX 线程实现的首席架构师。David 审校了本书与线程相关的章节,提出了诸多改进意见,还耐心地纠正了几处作者对 POSIX 线程 API 的理解错误。
- **Geoff Clare** 目前在为 The Open Group 开发 UNIX 一致性测试包,从事 UNIX 标准化工作已逾 20 年,是 Austin Group 的 6 位关键参与者之一,该组的宗旨是开发出构成 POSIX.1 和 Single UNIX Specification 基础卷的共同标准。Geoff 仔细审校了手稿中与 UNIX 编程接口相关的内容,耐心细致地提出了众多修正和改进意见,发现了诸多潜藏于手稿中的错误,在突出标准对可移植编程的重要性方面助益良多。
- **Loïc Domaigné** (当时供职于德国空中交通管制中心[German Air Traffic Control]) 是一名系统软件工程师,主要从事分布式、多并发、容错型嵌入式系统的设计和开发,此

类系统对实时性有着严苛的要求。他针对 SUSv3 中与线程规范有关的内容发表过评论和建议，在多个网上技术论坛中古道热肠，诲人不倦，无私地分享自己的编程心得。Loïc 细致地审校了本书与线程相关各章节，以及多处其他内容。除了编写若干精巧的程序来验证 Linux 线程实现的细节以外，他还倾注了巨大的热情并鼓励作者，提出了许多建议以改进本书整体的表现形式。

- **Gert Döring** mgetty 和 sendfax 程序的开发者，这一“双子座”也是 Linux/UNIX 系统上使用最为广泛的开源传真软件包。最近，他主要忙于搭建并维护基于 IPv4 和 IPv6 的大型网络，其肩负的主要任务是：与全欧洲的同事一起定义有效的网络策略，以确保 Internet 基础设施顺畅运行。Gert 审校了本书与终端、登录记账、进程组、会话以及任务控制相关各章节，并反馈了大量的有用信息。
- **Wolfram Gloger** 是一名 IT 顾问，过去 15 年，他参与过许多自由和开源软件项目（Free and Open Source Software, FOSS）。除此之外，Wolfram 还是 GNU C 语言库中 malloc 软件包的实现者。目前，他主要从事于 Web 服务的开发，尤其专注于网上教学，当然，在内核和系统库方面，他仍会偶露峥嵘。Wolfram 审校了本书诸多章节，尤其是侧重于内存方面的内容。
- **Fernando Gont** 是阿根廷国家科技大学（Universidad Tecnológica Nacional, Argentina）电子信息中心（Centro de Estudios de Informática, CEDI）成员。Internet 工程技术（Internet engineering）是其兴趣所在，在 Internet 工程任务组（IETF）也能见到其活跃的身影，他还是多个 RFC 文档的作者。此外，Fernando 还为英国 CPNI（国家基础设施保护机构）中心效力，以提供对通信协议安全方面的评估，而首个完整的 TCP 和 IP 协议安全评估报告也正是由他提出的。Fernando 仔细审校了本书涉及网络编程的相关章节，不厌其烦地向作者解释了 TCP/IP 协议的诸多细节，并对相关内容提出了不少改进意见。
- **Andreas Grünbacher**（SUSE 实验室）是位内核高手，还是 Linux 扩展属性和 POSIX 访问控制列表的实现者。Andreas 除了仔细审校了本书多章内容以外，还对作者勉励有加，有时，他的只言片语便极有可能改变这部书的整体结构。
- **Christoph Hellwig** 是 Linux 存储和文件系统咨询师，也是内核方面公认的行家，参与过 Linux 内核中多个部分的开发工作。在忙于编写及审查 Linux 内核补丁代码之余，Christoph 抽空审校了本书若干章节，并提出了诸多有益的改进和修正意见。
- **Andreas Jaeger** 曾领导过 Linux 向 x86-64 架构的移植开发工作。身为 GNU C 语言库的开发者，他不但将该库移植到了 x86-64 平台，而且还促成该库符合多个领域的标准，尤其是在数学库方面。他当前效力于 Novell 公司，是 openSUSE 的程序经理。Andreas 审校的章节之多，超乎作者预期。在本书的写作过程中，他除了提出诸多改进意见之外，也给予作者热情的鼓励。
- **Rick Jones** 绰号“Mr. Netperf”（HP 公司联网系统的性能偏执狂），对本书的网络编程相关章节提出了宝贵意见。
- **Andi Kleen**（当时效力于 SUSE 实验室）长期以来，一直是内核方面公认的行家里手，对 Linux 内核诸多不同领域贡献颇多，包括：网络、错误处理以及底层架构代码等方面。Andi 对网络编程相关内容做了全面审校，使作者在 Linux TCP/IP 实现方面大开眼界，此外，他还提供了许多建议，用以改善本书主题的展现形式。
- **Martin Landers**（Google）在我有幸与他共事时，他还是一名学生。其后，他在短期内便集诸多技能于一身，且形象百变——软件架构师、IT 培训师以及职业黑客。劳

Martin 大驾审校本书，实为作者之幸。他对本书的评论和更正往往一针见血，对多章内容质量的改进功莫大焉。

- **Jamie Lokier** 是公认的内核高手，投身于 Linux 开发已达 15 年之久。如今，他自封为“专家，长于解决潜伏于 Linux 系统中的疑难杂症”。Jamie 极其全面地审校了本书涉及内存映射、POSIX 共享内存以及虚拟内存操作等方面的章节。他的审校工作不但纠正了作者对相关主题细节方面的许多误解，相应各章的结构也得以大为改观。
- **Barry Margolin** 在其 25 年职业生涯中，从事过系统程序员、系统管理员以及技术支持工程师。当前，他作为一名系统性能工程师，供职于 Akamai 技术公司。在各种讨论 UNIX 和 Internet 技术主题的网络论坛上，他频频现身，威名素著。他还是多本相关技术主题书籍的技术审稿人。Barry 审校了本书若干章节，并提出了诸多改进意见。
- **Paul Pluzhnikov** (Google) 之前曾是 Insure++ 内存调试工具的技术带头人和主要开发者。有时，他也会以 GDB 黑客的身份现身，在网上论坛里积极地回复有关调试、内存分配、共享库以及运行时环境方面的问题。Paul 审校了本书多章内容，提出了许多宝贵意见。
- **John Reiser (与 Tom London)** 实现了 UNIX 向 32 位架构移植的早期版本之一 VAX-11/780。他还是 mmap() 系统调用的编写者。John 审校了本书多章内容，自然也包括了 mmap() 所在的章节。他所提供的大量历史洞见，及其对技术透彻地阐释为本书增色不少。
- **Anthony Robins** (新西兰 Otago 大学计算机科学副教授) 笔者 30 年的密友，本书某些章节的第一个读者，也是最早提出宝贵意见的技术审校者，在本书的写作过程中，一直给予作者以激励。
- **Michael Schröder** (Novell) GNU screen 程序的主要开发者之一，这项编程工作已令 Michael 在终端驱动程序的实现方面达到了“巨细靡遗，了如指掌”的境界。Michael 除了审校本书与终端和伪终端相关的章节以外，还针对进程组、会话以及任务控制等章节反馈了极为有益的意见。
- **Manfred Spraul** 曾从事过 Linux 内核中 (包括但不限于) IPC 的开发工作，不吝审校了本书与 IPC 相关的若干章节，并提出了许多改进意见。
- **Tom Swigg**, 作者在 DEC 从事培训工作时曾与他共事，作为本书最早的技术审校者之一，他对许多章节都反馈了极其重要的意见。Tom 从事软件工程师和 IT 培训师的工作已逾 25 年，目前就职于伦敦南岸大学 (London South Bank University)，在 VMware 环境下从事 Linux 编程和技术支持工作。
- **Jens Thoms Törning** 继承了物理学家改学编程的优良传统，大批开源的设备驱动程序和其他软件都出自他手。Jens 审校的章节之多，在技术方面跨度之大，着实令人瞠目，他对各章内容的改进都提出了独特而又弥足珍贵的见解。

还有许多其他技术审稿人也审校了本书的不同内容，并提出了诸多宝贵意见。在此，作者向以下一干技术审稿人表示感谢 (以姓氏字母顺序排列): George Anzinger (MontaVista Software)、Stefan Becher、Krzysztof Benedyczak、Daniel Brahneborg、Andries Brouwer、Annabel Church、Dragan Cvetkovic、Floyd L. Davidson、Stuart Davidson (Hewlett-Packard Consulting)、Kasper Dupont、Peter Fellingner (jambit GmbH)、Mel Gorman (IBM)、Niels Göllesch、Claus Gratzl、Serge Hallyn (IBM)、Markus Hartinger (jambit GmbH)、Richard Henderson (Red Hat)、Andrew Josey (The Open Group)、Dan Kegel (Google)、Davide Libenzi、Robert Love (Google)、

H.J. Lu (Intel Corporation)、Paul Marshall、Chris Mason、Michael Matz (SUSE)、Trond Myklebust、James Peach、Mark Phillips (Automated Test Systems)、Nick Piggin (Novell SUSE 实验室)、Kay Johannes Potthoff、Florian Rampp、Stephen Rothwell (IBM Linux 技术中心)、Markus Schwaiger、Stephen Tweedie (Red Hat)、Britta Vargas、Chris Wright、Michal Wronski 以及 Umberto Zamuner。

除了技术审稿人之外，作者还得到了各界人士及组织在其他方面的帮助。

我要感谢以下人等为我解答技术难题，他们是：Jan Kara、Dave Kleikamp 和 Jon Snader。我要感谢 Claus Gatzl 和 Paul Marshall 在系统管理方面对我的帮助。

我要感谢 Linux 基金会 (LF)。2008 年间，LF 资助我作为一名全职研究人员参与 man-pages 项目，并从事 Linux 编程接口的测试和设计评审工作。虽然 LF 全职研究员的身份不能为本书的写作提供直接的资金支持，但却使作者得以养家糊口，这一助力本意在于令我全身心投入对 Linux 编程接口的测试以及对文档的编纂工作，却也惠及作者的“私活”。抛开公事不谈，我要感谢 Jim Zemlin——我在 LF 的“接口”人，还要感谢 LF 技术咨询委员会的一干专家，感谢他们对我的聘任。

感谢 Alejandro Forero Cuervo 对本书书名的建议！

25 年前，在我为第一个学位拼搏之时，Robert Biddle 激起了我对 UNIX、C 以及 Ratfor 的兴趣，谢谢你，老兄。虽然以下诸君与本书并无直接干系，但当我在新西兰坎特伯雷大学攻读第二学位时，他们就鼓励我在写作道路上坚持下去，在此，我要向他们表示感谢，他们是 Michael Howard、Jonathan Mane-Wheoki、Ken Strongman、Garth Fletcher、Jim Pollard，以及 Brian Haig。

由 Richard Stevens 所著的几部关于 UNIX 编程和 TCP/IP 方面的杰作，数年来一直被我辈程序员奉为圭臬，只可惜先贤已逝。凡是读过上述书籍的读者势必会注意到，本书与 Richard Stevens 的那几本巨著看起来有些相似。这并非偶然。在构思本书时，作者曾从较为宏观的角度就书籍设计反复斟酌，可最终发现 Richard Stevens 所采用的方法才是正解，正因如此，本书采用了与其相同的展示方式。

感谢下列人士和组织为我提供 UNIX 系统，使我得以运行测试程序，并验证其他 UNIX 实现的细节，感谢 Anthony Robins 和 Cathy Chandra 在新西兰 Otago 大学所提供的多种 UNIX 测试系统，感谢 Martin Landers、Ralf Ebner 和 Klaus Tilk 在德国慕尼黑技术大学 (Technische Universität) 所提供的多种 UNIX 测试系统，感谢 HP 公司在 Internet 上免费开放他们的 testdrive 系统，感谢 Paul de Weerd 使我得以访问 OpenBSD 系统。

要衷心感谢两家慕尼黑公司及其老板，这两家公司除了为我提供了工作机会（还是弹性工作制）和热情的同事，还格外开恩，允许我在写作本书时使用他们的办公室。感谢 exolution 有限公司的 Thomas Kahabka 和 Thomas Gmelch，特别要感谢 jambit 有限公司的 Peter Fellingner 和 Markus Hartinger。

感谢下列人士对我提供的各种帮助，他们是 Dan Randow、Karen Korrel、Claudio Scalmazzi、Michael Schüpbach 和 Liz Wright。感谢 Rob Suisted 和 Lynley Cook 为封面和封底所提供的照片。

感谢下列人士以不同方式给作者以鼓励和支持，他们是 Deborah Church、Doris Church 和 Annie Currie。

感谢 No Starch 出版社大队人马为这一庞大创作项目所提供的各种帮助。Bill Pollock 从项目之初就一直秉持直言不讳的风格，始终对本书的完成充满信心，并耐心地关注着项目的进展，我要对他表示感谢。感谢本书最初的责任编辑 Megan Dunchak。感谢本书的文字编辑

Marilyn Smith, 无论我如何殚精竭虑以求文字的清晰与一致, 此君总能从鸡蛋里挑出骨头。本书的版面和设计由 Riley Hoffman 全面负责, 在“上了同一条船”后又挑起了制作编辑的重担。Riley 总是不厌其烦地满足我的请求, 以求本书的排版无误——最终结果堪称完美。谢谢你。

现在, 我才体味出下面这句老话的真正含义: 一人写作, 全家受累。感谢 Britta 和 Cecilia 对我的支持, 感谢你们能容忍我因写作本书而长时间地不着家。

许可

承蒙 IEEE (美国电气电子工程师学会) 和 The Open Group 惠允, 本书得以引用 IEEE Std 1003.1, 2004 版以及 The Open Group 基础规范第 6 号 (Issue 6) 中 POSIX (可移植性操作系统接口)——信息技术标准的部分文字。可通过 <http://www.unix.org/version3/online.html> 在线查阅规范的完整版本。

Web 站点和程序示例的源码

读者可在 <http://man7.org/tlpi> 上找到更多有关本书的信息, 包括本书的勘误表和程序示例源码。

反馈

欢迎读者提供 BUG 报告、对代码的改进建议, 以及为进一步提高代码可移植性而提出的修订意见。同样欢迎读者提供针对本书内容的缺陷报告和改进叙述方式的一般性建议。当前的勘误列表可参见 <http://man7.org/tlpi/errata/>。由于 Linux 编程接口变化无常、且变更有时极为频繁, 仅凭作者一己之力很难“与时俱进”, 因此读者就全新或已变更的 Linux 编程接口特性所提供的反馈信息, 作者也将乐于收到, 并会纳入本书的下一版中。

Michael Timothy Kerrisk

于德国慕尼黑和新西兰克赖斯特彻奇

2010 年 8 月

mtk@man7.org

目 录

第 1 章 历史和标准	1
1.1 UNIX 和 C 语言简史.....	1
1.2 Linux 简史.....	4
1.2.1 GNU 项目.....	4
1.2.2 Linux 内核.....	5
1.3 标准化.....	8
1.3.1 C 编程语言.....	8
1.3.2 首个 POSIX 标准.....	9
1.3.3 X/Open 公司和 The Open Group.....	10
1.3.4 SUSv3 和 POSIX.1-2001.....	10
1.3.5 SUSv4 和 POSIX.1-2008.....	12
1.3.6 UNIX 标准时间表.....	12
1.3.7 实现标准.....	14
1.3.8 Linux、标准、Linux 标准规范 (Linux Standard Base).....	14
1.4 总结.....	15
第 2 章 基本概念	17
2.1 操作系统的核心——内核.....	17
2.2 shell.....	19
2.3 用户和组.....	20
2.4 单根目录层级、目录、链接及文件.....	21
2.5 文件 I/O 模型.....	23
2.6 程序.....	24
2.7 进程.....	25
2.8 内存映射.....	27
2.9 静态库和共享库.....	28
2.10 进程间通信及同步.....	28