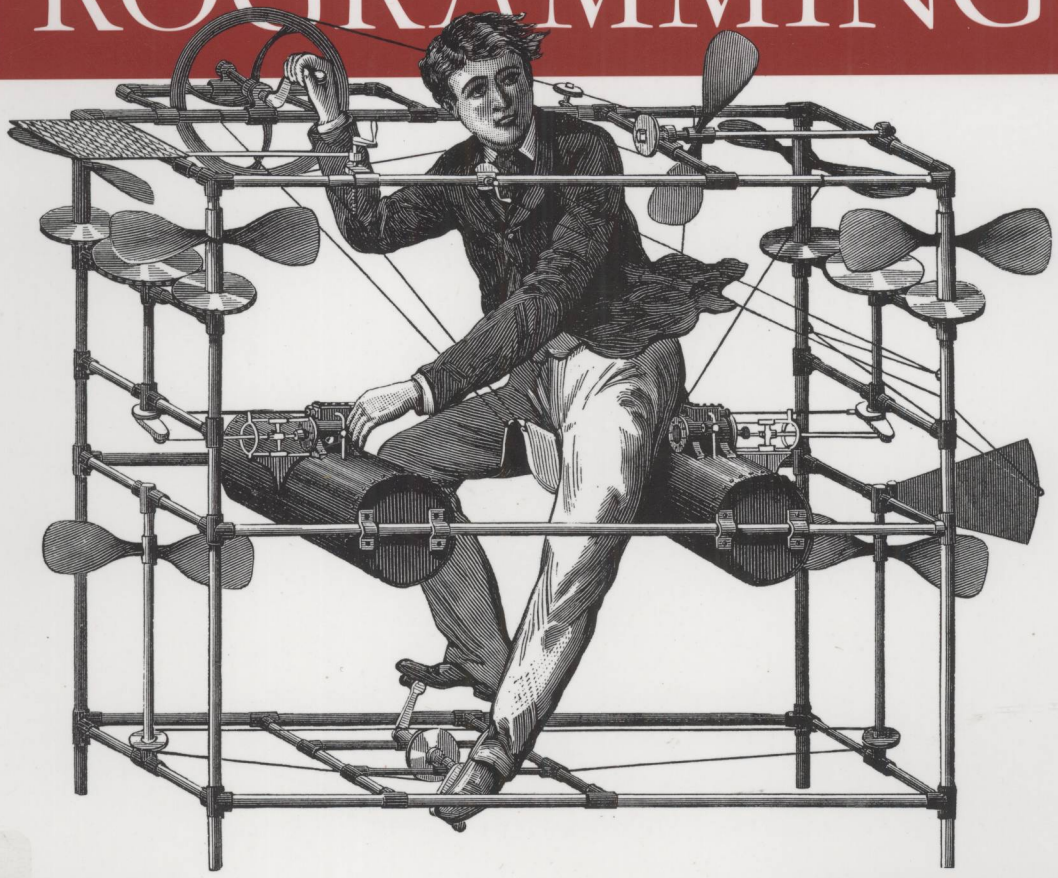
LINUX系统编程

LINUX SYSTEM PROGRAMMING



O'REILLY®
東南大學出版社

ROBERT LOVE 著
O'REILLY TAIWAN公司 编译

LINUX
系统编程

Robert Love 著
O'Reilly Taiwan 公司 编译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

南京 东南大学出版社

图书在版编目（CIP）数据

Linux 系统编程 /（美）洛夫（Love, R.）著；O'Reilly Taiwan 公司编译 . —南京：东南大学出版社，2009.7

书名原文：Linux System Pogramming

ISBN 978-7-5641-1519-7

Ⅰ . L… Ⅱ . ①洛… ②O… Ⅲ .Linux 操作系统－程序设计 Ⅳ .TP316.89

中国版本图书馆 CIP 数据核字（2009）第 084166 号

江苏省版权局著作权合同登记

图字：10-2008-357 号

©2007 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2009. Authorized translation of the English edition, 2007 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2007。

简体中文版由东南大学出版社出版 2009。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者 —— O'Reilly Media, Inc. 的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式重制。

Linux 系统编程（中文版）

出版发行：东南大学出版社

地　　址：南京四牌楼 2 号　　邮编：210096

出 版 人：江　汉

网　　址：http://press.seu.edu.cn

电子邮件：press@seu.edu.cn

印　　刷：扬中市印刷有限公司

开　　本：787 毫米 × 980 毫米　16 开本

印　　张：24.75 印张

字　　数：412 千字

版　　次：2009 年 7 月第 1 版

印　　次：2009 年 7 月第 1 次印刷

书　　号：ISBN 978-7-5641-1519-7

印　　数：1~3000 册

定　　价：56.00 元（册）

本社图书若有印装质量问题，请直接与读者服务部联系。电话（传真）：025-83792328

O'Reilly Media, Inc. 介绍

为了满足读者对网络和软件技术知识的迫切需求，世界著名计算机图书出版机构O'Reilly Media, Inc.授权东南大学出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc.是世界上在 Unix、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时也是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为20世纪最重要的50本书之一）到GNN（最早的Internet门户和商业网站），再到WebSite（第一个桌面PC的Web服务器软件），O'Reilly Media, Inc.一直处于Internet发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc.是最稳定的计算机图书出版商 —— 每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc.具有深厚的计算机专业背景，这使得O'Reilly Media, Inc.形成了一个与其他出版商迥然不同的出版方针。O'Reilly Media, Inc.所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc.还有许多固定的作者群体 —— 他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc.依靠他们及时地推出图书。因为O'Reilly Media, Inc.紧密地与计算机业界联系着，所以O'Reilly Media, Inc.知道市场上真正需要什么图书。

作者简介

Robert Love 是一位资深的 Linux 使用者和计算机迷。他积极 —— 并且热心 —— 参与 Linux 内核以及 GNOME desktop 社区的各项计划。他近来对 Linux 内核的贡献包括 kernel event layer 与 inotify 子系统。GNOME 方面的贡献包括 Beagle、GNOME Volume Manager、NetworkManager 以及 Project Utopia。目前，Robert 在 Google 的 Open Source Program Office 工作。

Robert 还是《Linux Kernel Development Second Edition》（Novell Press）的作者以及《Linux in a Nutshell Fifth Edition》（O'Reilly）的共同作者。他也是《Linux Journal》的撰稿编辑，曾撰写过许多文章，而且受邀到世界各地发表演讲。

Robert 于佛罗里达大学取得他的数学文学士以及计算机科学理学士学位。他来自南佛罗里达州，现在住在波士顿。

封面设计

本书的封面是一个在飞行器里的人。在莱特兄弟于 1903 年做出第一个比空气还重的载人器之前，世界各地的人们都想要乘坐简单但精心制作的机器来飞行。

2 或 3 世纪，据说中国的诸葛亮发明了可以在空中飞行的孔明灯（天灯），这是热气球的始祖。5 或 6 世纪左右，据称中国有许多人将自己绑在大型风筝上飞行于空中。

还有一种说法，中国所创造的旋转玩具（竹蜻蜓）是直升机的雏型，它的外观可能启发了达芬奇设计出直升机的原型，以作为人类飞行的解决方案。达芬奇还研究鸟类，并且设计出了降落伞。1845 年，他设计出了扑翼机（ornithopter），这是一个可以载人的抖动双翼的飞行器。尽管他未曾制造过它，但是扑翼机与鸟相似的结构影响了整个世纪的飞行器的设计。

封面所描绘的飞行器相较于 James Means 于 1893 所设计的滑翔机模型（没有螺旋桨）更是精心杰作。Means 之后为他的滑翔机出版了说明书，书中提到“Mt. Willard 山顶，靠近 Crawford House, N.H.，找到了一个很好的地方，可以实验这些机器。”

此为试读，需要完整PDF请访问：www.ertongbook.com

但这样的实验往往很危险。19 世纪晚期，Otto Lilienthal 制造出了单翼机（monoplane）、双翼飞机（biplane）以及滑翔机（glider）。他首先证明，载人飞行器可以飞行至伸手可及的距离之内，他获得了“飞行测试之父”的昵称，因为他进行了2000 多次滑翔机的飞行测试，有时可以飞行超过 1000 英尺的距离。在一次紧急着陆折断他的脊椎骨后，他于 1896 年辞世。

飞行器也被称为机械鸟（mechanical bird）以及飞艇（airship），并偶尔会获得更加丰富多彩的名字，例如人造信天翁（Artificial Albatross）。飞行器的热潮仍持续着，因为航空发烧友们仍在制造早期的飞行器。

目录

此为试读,需要完整PDF请访问: www.ertongbook.com

序

每当 Linux 内核开发者不高兴的时候总是喜欢丢出一句话："用户空间只是内核的一个试验负载（test load）。"

内核开发者之所以会这么说，目的是于任何用户空间程序代码执行失败的时候撇清所有责任。当所发生的问题绝对不是内核的过错时，他们所关心的是用户空间开发者应该去修正他们自己的程序代码。

为了证明这通常不是内核的过错，3 年多前，一位具领导地位的 Linux 内核开发者曾在挤满人的会议室里，以"Why User Space Sucks"（为何用户空间程序很糟糕）为题发表演讲，他举出实例说明我们每个人每天使用了哪些可怕的用户空间程序代码。其他内核开发者则以自己创建的工具来展示差劲的用户空间程序如何滥用硬件，并耗尽无预警的笔记本电脑电池的电量。

尽管用户空间程序对嘲笑它的内核开发者而言可能只是一个"试验负载"，不过这些内核开发者也是每天都得依靠这些用户空间程序。如果没有用户空间程序可用，所有的内核充其量就只能在屏幕上交替输出 ABABAB 样式的信息。

而今，Linux 已经成为有史以来最灵活、最强大的操作系统，随处都可以看到它的踪迹，不仅最小型的手机和嵌入式装置运行它，全世界前 500 台速度最快的超级计算机中也有70%以上的在运行它。其他操作系统从未曾有过这么好的规模，也不会遭受各种硬件和环境的挑战。

如同内核一样，在 Linux 的用户空间上运行的程序也得运作在各种平台上，以人们所依赖的应用程序和公用程序提供给全世界使用。

Robert Love 自找麻烦地想要教读者关于Linux 系统上所有的系统调用，本书于焉诞生。本书将以用户空间的观点让读者全面了解 Linux 内核的运作原理，以及如何利用此系统的强大功能。

本书将告诉你如何创建可以在各种 Linux 发行版本以及硬件平台上运行的程序代码。本书也会让你了解 Linux 如何运作以及如何善用它的灵活性。

最后，也是最好的一件事，本书会教你如何让自己所编写的程序不会太差劲。

—— Greg Kroah-Hartman

前言

本书所要探讨的是系统编程——特别是Linux系统编程。系统编程就是编写系统软件，也就是位于底层的程序代码，它可以直接跟内核与核心系统链接库交互。换句话说，本书的主题就是 Linux 系统调用以及其他的低端函数，例如 C 链接库所定义的。

尽管已经有许多书在探讨 Unix 的系统编程，但很少有只将焦点放在 Linux 上的，就算有，也很少有探讨最新 Linux 版本以及 Linux 独有的高级接口的。此外，本书的读者将得益于我的特殊经验：我曾为 Linux 编写过许多程序代码（包括内核以及系统软件的）。事实上，我曾经实现过本书所探讨的某些系统调用以及其他功能。因此，本书揭露了许多内幕知识，不仅会描述系统接口应该如何运作，也会说明它们实际的运作方式，以及你（编程者）如何以最有效的方式来使用它们。所以本书是 Linux 系统编程的教材，也是 Linux 系统调用的参考手册，亦是编写更聪明、更快的程序代码的权威指南。本书内容有趣且容易理解，不管你是否需要每日编写系统层的程序代码，本书都会教你让你可以编写出较理想程序代码的诀窍。

本书的读者

本书假设读者已经熟悉 C 编程以及 Linux 编程环境，不需要精通这些主题，但至少要了解它们。如果你尚未读过任何有关 C 编程语言的书籍，例如 Brian W. Kernighan 和 Dennis M. Ritchie 的经典之作《*The C Programming Language*》（Prentice Hall 出版，本书就是大家所熟悉的 K&R），我强烈建议你阅读这本书。也许你不习惯使用 Unix 文本编辑器，Emacs 和 *vim* 是最常见且备受推崇的编辑器，你可以择一使用。你还应该熟悉 *gcc*、*gdb*、*make* 等工具的基本用法。市面上可以找到许多与 Linux 编程工具及实践有关的书籍，本书结尾的参考文献列举了几本有用的参考书。

关于 Unix 或 Linux 系统编程，我已经对读者所应具备的知识作了以上假设。本书将从基础开始谈起，并蜿蜒前进到高端接口及优化诀窍。希望所有层次的读者都会觉得这是一本值得阅读以及能够学到新东西的书。在本书的写作过程中，我确实是这么想的。

我也不是没有假设本书读者的类型以及阅读本书的动机。本书适用于那些希望能够在底层写出好程序的工程师，但是那些想在自己安身之处寻找更稳固立足点的较高层的编程人员也可以在本书中找到许多对他们有用的资料。

不管你的动机是什么，祝你阅读愉快。

本书的内容

本书分成 10 章、1 个附录以及 1 个参考文献。

第一章 *介绍与基本概念*

本章可作为导读，它概述了 Linux、系统编程、内核、C 链接库以及 C 编译器。即使是高级用户也应该阅读这一章 —— 相信我。

第二章 *文件 I/O*

本章将介绍文件（Unix 环境中最重要的抽象概念）以及文件 I/O（Linux 编程的基本模式），并且说明如何进行文件的读写以及其他基本的文件 I/O 操作，最后还会探讨 Linux 内核的实现和管理文件的方式。

第三章 *缓冲式 I/O*

本章将探讨基本文件 I/O 接口的问题 —— 缓冲区大小的管理，以及介绍一般的缓冲式 I/O（尤其是标准 I/O）作为解决方案。

第四章 *高级文件 I/O*

本章将以高级的 I/O 接口、内存映射以及优化技术来结束本书对 I/O 的探讨。本章还会附带探讨如何避免搜索磁盘以及 Linux 内核的 I/O 调度程序扮演何种角色。

第五章 *进程管理*

本章将介绍进程（Unix 中第二重要的抽象概念）以及基本进程管理的一系列系统调用，包括令人尊敬的 *fork*。

第六章 *高级进程管理*

本章将继续探讨高级进程管理，包括实时进程。

第七章 *文件和目录管理*

本章将探讨创建、移动、复制、删除以及其他用来管理文件和目录的功能。

第八章 *内存管理*

本章将说明内存管理。首先会介绍 Unix 的内存概念，例如进程地址空间以及页面调度（paging），接着会探讨从内核取得内存空间以及将内存空间释回内核的接口。最后会探讨进阶的内存相关接口。

第九章 *信号*

本章将说明信号。首先会探讨信号以及它们在 Unix 系统中所扮演的角色。然后会说明信号接口，先探讨基本部分，再探讨高级部分。

第十章 *时间*

本章将探讨时间、修眠以及时钟管理。本章会从基本的接口开始谈起，一直到 5 POSIX 时钟以及高精度定时器。

附录 *GCC 对 C 语言的扩展*

本附录会回顾 *gcc* 和 GNU C 所提供的许多优化功能，例如constant、pure 和 inline 等用来标记函数的属性。

本书最后提供了一份参考书目，列出了对系统编程有帮助以及论述本书未提到的必备知识的相关书籍。

本书所涵盖的版本

Linux 系统接口可被定义成Linux 内核（操作系统的中心组件）、GNU C 链接库（*glibc*）以及 GNU C 编译器（*gcc* —— 现在被正式称为 GNU Compiler Collection，但是我们只关心 C 编译器）三者所提供的应用程序二进制接口（application binary interface）以及应用程序编程接口（application programming interface）。本书内容涵盖 Linux 内核 2.6.22 版、*glibc* 2.5 版以及 *gcc* 4.2 版三者所定义的系统接口。本书对系统接口所做的说明应该可以向下兼容于较旧的版本（不含新的接口）以及向上兼容于较新的版本。

如果任何进化中的操作系统是一个移动的目标，Linux 就是一只饥渴的猎豹。Linux 的进展是以日数（而非年数）来度量的，它频繁地发布内核与其他组件并且在不断变化中。没有哪一本书可以永久捕获这只充满活力的怪兽。

尽管如此，系统编程所定义的编程环境却不再改变。内核开发者尽可能不去改变系统调用，*glibc* 开发者高度重视向上（forward）和向下兼容（backward compatibility），而且 Linux 工具链（toolchain）可以产生跨版本（特别是 C 语言的）的兼容程序代码。因此，尽管 Linux 不断在变化，Linux 系统编程仍能保持稳定。我想说的事情很简单：不要担心系统接口有任何变化，购买本书就对了！

本书的排版惯例

下面是本书的排版惯例：

斜体字（*Italic*）
: 用于强调、新术语、URL、外来词汇、Unix 命令与实用程序、文件名、目录名以及路径名称。

等宽字（`Constant Width`）
: 用于指出头文件、变量、属性、函数、类型、参数、对象、宏以及其他编程结构。

等宽斜体字（*`Constant Width Italic`*）
: 用于指出以用户提供的值替换的文本（例如路径名称组件）。

此图标代表诀窍、建议或一般注意事项。

本书大部分的程序将以简洁、实用的程序代码片段来呈现，例如：

```
while (1) {
        int ret;

        ret = fork ();
        if (ret == -1)
                perror ("fork");
}
```

本书煞费苦心地提供简洁而实用的程序代码片段。你不会看到特殊的头文件、没有节制的宏和难以辨认的快捷方式，你不会看到庞大的程序，你所看到的是许多简单的范例程序。本书的范例不仅描述充分、完全可用，而且小而明确，我希望这些范例在读者首次

此为试读，需要完整PDF请访问：www.ertongbook.com