

Robert Love

Third Edition



Linux Kernel Development

A thorough guide to the design and implementation of the Linux kernel

Developer's Library



Linux Kernel Development

Third Edition

Robert Love

Translated by: Kevin Zhang

2010-12-10 – 2011-03-15(TODO)

本资料仅为学习所用，请于下载后 24 小时内删除，否则引起的任何后果均由您自己承担。本书版权归原作者所有，如果您喜欢本书，请购买正版支持作者。

目录

序.....	4
前言.....	6
如何使用本书.....	6
内核版本.....	7
本书读者.....	7
第三版鸣谢.....	8
关于作者.....	8
第 1 章 Linux 内核介绍	10
Unix 的历史	10
第 2 章 内核入门	11

序

随着 Linux 内核和使用内核的应用得到越来越广泛的使用，希望参与开发和维护 Linux 的系统软件开发者的数量也在不断增加。这些工程师中有的单纯是个人兴趣，有些任职于 Linux 公司，有些为硬件厂商工作，也有一些是个人开发项目相关。

但是所有人都面临同样的一个问题：内核学习曲线越来越长、越来越陡峭。内核系统非常庞大，而且复杂度与日俱增。随着这几年的发展，当前内核开发团队的成员对内核内部的知识更加深入和广泛，这也进一步扩大了他们和新手之间的差距。

我相信 Linux 内核源码的这种可达性衰退已经成为影响内核质量的一个问题，并且随着时间的发展会越来越严重。所有关心 Linux 发展的人都希望为内核贡献的开发者数量能够得到增长。

解决这个问题的一个办法是保持代码的清晰：合理的接口、统一的层次架构、“只做一件事并且做好”等等。这也正是 Linus Torvalds 所采取的办法。

我个人建议的办法是增加代码注释：读者在阅读代码时可以据此来理解代码编写者的真实意图。（找出意图和实现之间分歧的过程被称为调试，如果意图不明确就非常难以调试）。

但是代码注释也不能提供主要子系统的更高层次的意图，以及开发者是如何理解它的。要理解这种意图，就只有相关的书籍才能做到了。

Robert Love 的这本书给我们展示了富有经验的开发者对于内核的核心认识，内核子系统提供的各种服务、它们是如何提供这些服务。所有这些知识能够满足许多不同的人：好奇者、应用开发者、想要体会内核设计的人等等。

本书还能帮助有抱负的内核开发者步入下一个层次：修改内核来完成明确的既定目标。我非常鼓励有抱负的开发者亲自动手。理解内核的最佳途径就是修改它。修改内核要求开发者对内核的理解，要站在比阅读代码更高的层次。认真的内核开发者会加入邮件列表、并与其他开发者积极沟通。这种沟通是内核贡献者互相学习和保持一致的主要动力。Robert 很好的阐述了内核的结构和相关文化。

从 Robert 的书中你能学习并得到享受，你会决定进入下一个层次，成为内核开发社区的一份子。我们通过人们的贡献来评价和衡量一个人，当你为 Linux 做出贡献时，即使你完成的工作很小，但是能够立即让数百万人获益。这是特权和责任能够得到的最大享受。

Andrew Morton

前言

当我最早考虑将我对 Linux 内核的经验转换为一本书时，我始终都怀着一丝忧虑。怎样才能使我的书成为该主题最好的书籍？如果我不能做特别的、最好的工作，那我就不会感兴趣。

最终我意识到我可以对内核这一主题提供独到的方法。我的工作是 **hacking** 内核、我的嗜好是 **hacking** 内核、我也热爱 **hacking** 内核。这么多年以来，我积累了大量有趣的轶闻和内情。以我的经验可以写一本如何 **hack** 内核（同样重要的是如何防止 **hack** 内核）的书。首先也是最重要的，这是一本关于 Linux 内核设计和实现的书。这本书使用的方式与它的竞争者不同，我会提供足够的信息使你能更好地完成工作。我是一个实用主义的工程师，这也是一本注重实践的书。本书应该很有趣、阅读很轻松、并且非常有用。

我希望读者通过阅读本书，能够加深对 Linux 内核规则的理解。我的目标是让你（没有阅读过本书和内核源代码的人），能迅速入门并开始编写有用的、正确的、清晰的内核代码。当然，你也可以仅仅把本书当作娱乐使用。

本书第一版出版至今，已经好几年过去了。第三版比之前两版提供了相当多的内容：修订、更新、和许多全新的章节。第三版吸收并增加了第二版之后内核的所有变更。更重要的是，Linux 内核社区决定短期内不继续进行 2.7 版本内核的开发，而是计划继续开发和稳定 2.6 系列。这个决定有许多影响，但是与本书相关的则是继续使用 2.6 版本的内核。由于 Linux 内核非常成熟，即使是内核快照也能保持相当长时间的稳定。本书可以说是内核的权威文档，同时兼顾内核的历史和未来。

如何使用本书

开发内核代码不需要天才和魔法，甚至 Unix 黑客标志性的胡子也不需要。内核尽管也有一些自己的有趣的规则，但它与其它任何一个大型软件并无太大区别。当然你需要掌握许多细节（和任何大型项目一样），但区别只是数量，而不是性质。

要学习内核，使用源码是必须的。Linux 系统源码的开放可用性，是一份珍贵的礼物。但是仅仅阅读源码是远远不够的，你还需要深入进去并修改一些代码、找出 bug 并修复它、改进你使用的硬件驱动、增加新功能。即使是很小的事情，你都要找出并完成它。只有当你真正编写了内核代码，才能真正理解内核的所有。

内核版本

本书基于 Linux 内核 2.6 系列版本，不涉及更老的内核，除非与历史相关。例如只要对于我们的讨论是有帮助的，我们也会讨论某个内核子系统在 2.4 版本中的实现方式。具体来说，本书第三版使用了 Linux 内核 2.6.34。尽管内核一直在不断的前进，要保持更新确实很难，但我的目标是使本书能够同时适用于老的和新的内核开发者和用户。

尽管本书讨论 2.6.34 内核，但我花了很大力气确保书中的材料也能适用于 2.6.32 内核。后者是许多 Linux 发布版为“企业”内核所做的剪裁，在许多年内都将在产品系统中使用，也会有持续动态的开发支持（2.6.9，2.6.18 和 2.6.27 都是类似的长期发布版）。

本书读者

本书的目标读者是对理解 Linux 内核感兴趣的开发者和用户。它不是对内核源码一行一行的注释，也不是驱动开发的指南，或内核 API 的参考手册。相反本书的目标是提供 Linux 内核设计和实现的足够的信息，使程序员能开始开发内核代码。内核开发非常有趣，报酬也不错，而我希望本书能尽快把读者引入这个世界。本书既讨论理论又兼顾实用，所以应该同时适用于学院派和实践派的读者。我一直认为只有掌握了理论，才能更好地进行实践，所以我试图在本书中平衡二者。我希望无论你理解 Linux 内核的动机是什么，本书都能满足你的需求，为你解释清楚内核的设计和实现。

因此本书同时讲解了核心内核子系统的使用，和它们的设计与实现。我认为这一点很重要，值得我们稍加讨论。以第 8 章“Bottom Halves 和 Deferring Work”

为例，该章讲解了一个被称为 **bottom halves** 的设备驱动组件。在那一章中，我既讨论了内核 **bottom-half** 机制的设计和实现（核心内核开发者和学院派应该会感兴趣），也讨论了如何使用导出的接口来实现你自己的 **bottom half**（设备驱动开发者和黑客会感兴趣）。核心内核开发者当然需要理解内核的内部工作原理，而他们一般都对接口的使用非常熟悉。同时设备驱动开发者也能从理解接口背后的实现得到很多好处。

这和学习某个库的 **API** 与学习库的实现是一个道理。表面看来，应用程序员只需要理解 **API**（将接口视作黑箱通常是足够的）；库开发者只需要关心库的设计和实现。但我认为双方都应该花费一些时间来学习另一方面。那些更好地理解底层操作系统的应用程序员，能够更好地使用操作系统的接口。同样，库开发者也不应该与应用如何使用库完全脱离。因此，我同时讨论了内核子系统的设计和使用，希望本书能够对双方都有用。

我假设读者了解 **C** 编程语言，熟悉 **Linux** 系统。有一些操作系统设计和计算机科学相关的理论对学习本书会有帮助，我也会尽可能多地解释相关的概念。如果我没有，附录的参考书目列出了一些操作系统设计方面的卓越书籍。

本书也适用于操作系统设计课程的大学教材，可以作为理论介绍教材的补充材料。本书可以在大学高年级课程或研究生课程中使用。

第三版鸣谢

（略）

关于作者

Robert Love 是一个开源程序员，演讲者，和作者，他使用和贡献 **Linux** 已经超过 15 年。**Robert** 目前是 **Google** 的高级软件工程师，是 **Android** 移动平台内核团队的成员之一。在去 **Google** 之前，他是 **Novell Linux** 桌面的首席架构师。在 **Novell** 之前，他是 **MontaVista** 软件和 **Ximian** 的内核工程师。

Robert 的内核项目包括抢先式内核、进程调度器、内核事件层、**inotify**、**VM**

增加、和几个设备驱动。

Robert 进行了无数演讲，撰写了许多关于 Linux 内核的文章。他也是 Linux Journal 的编辑之一。他编写的其它几本书包括 Linux System Programming 和 Linux in a Nutshell。

Robert 获得了福罗里达大学的数学和计算机科学学位。现居住于波士顿。

第 1 章 Linux 内核介绍

本章介绍 Linux 内核和 Linux 操作系统，将它们跟 Unix 的历史结合在一起。今天，Unix 指的是一组操作系统，它们实现了类似的应用编程接口（API），并且遵循相同的设计原则。但是 Unix 也是一个特定的操作系统，距今已有 40 多年。要理解 Linux，我们必须首先讨论最初的 Unix 系统。

Unix 的历史

经过四十多年的使用，计算机科学家仍然赞誉 Unix 操作系统是当今最强大最优美的系统。从 1969 年 Unix 诞生以来，Dennis Ritchie 和 Ken Thompson 的作品已经成为一个传奇，Unix 的设计经受住了时间的考验。

Unix 起源于 Multics，后者是 Bell 实验室参与的一个失败的多用户操作系统项目。Multics 项目终止以后，Bell 实验室计算机科学研究中心的成员发现自己没有可用的交互式操作系统。于是在 1969 年夏，Bell 实验室的程序员们设计出了一个文件系统，最终演化成了 Unix。为了测试该系统的设计，Thompson 在一台 PDP-7 机器上实现了这个新系统。在 1971 年，Unix 成功移植到 PDP-11；然后在 1973 年，又用 C 语言重写了 Unix 操作系统。在当时这是史无前例的一步，但也正是如此，才为将来的可移植性铺平了道路。第一个在 Bell 实验室之外得到广泛使用的 Unix 操作系统是 Unix System 第六版，经常被称为 V6。

许多公司将 Unix 移植到新机器上。这些移植在增强 Unix 功能的同时，也导致了 Unix 操作系统的几个不同变种。1977 年 Bell 实验室组合这些变种到一起，发布了 Unix System III；1982 年 AT&T 发布了 System V。¹

第 2 章 内核入门