<<Linux内核设计与实现>>

图书基本信息

书名：<<Linux内核设计与实现>>

13位ISBN编号：9787111338291

10位ISBN编号：7111338294

出版时间：2011-4-30

出版时间：机械工业出版社华章公司

作者：Robert Love

页数：335

译者：陈莉君,康华

前言

随着Linux内核和Linux应用程序越来越成熟，越来越多的系统软件工程师涉足Linux开发和维护领域。
他们中有些人纯粹是出于个人爱好，有些人是为Linux公司工作，有些人是为硬件厂商做开发，还有一些是为内部项目工作的。
但是所有人都必须直面一个问题：内核的学习曲线变得越来越长，也越来越陡峭。
系统规模不断扩大，复杂程度不断提高。
虽然现在的内核开发者对内核的掌握越发炉火纯青，但新手却无法跟上内核发展的步伐，长此以往将出现青黄不接的断层。
我认为这种新老鸿沟已经成为内核质量的一个隐患，而且问题将继续恶化。
所以那些真正关心内核的人已经开始致力于扩大内核开发群体。
解决上述问题的一个方法是尽量保证代码简洁：接口定义合理，代码风格一致，“一次做一件事，做到完美”等。
这也就是LinusTorvalds倡导的解决办法。
我提倡的解决办法是对代码慷慨地加上注释，即能够让读者立刻了解代码开发者意图的文字（识别意图和实现之间差异的工作称为调试。
如果意图不明确显然调试就难以进行）。
可是，即使有注解，也没办法清楚地展现内核的各个主要子系统的全景，说明它们到底要做什么。
那么，这些开发者又该从何下手呢？
由文字材料来说明这些在起步阶段就该理解的材料，其实是最合适的。
RobertLove的贡献就在于此，有经验的开发者可以通过本书全面了解内核子系统提供的服务，同时还可以了解这些服务是怎么实现的。
对不少人来说，这些知识就已经足够了：那些好奇的人，那些应用程序开发者，那些想对内核的设计品头论足一番的人，都有足够的谈资了。
但是学习本书同样可以作为那些有抱负的内核开发者更上一层楼的契机，可以帮他们更改内核代码以达到预定的目标。
我建议有抱负的开发者能够亲身实践：理解内核某部分的捷径就是对它做些修改，这样能为开发者揭示仅仅通过看内核代码无法看到的深层机理。
严肃认真的内核开发者应该加入开发邮件列表，不断和其他开发者交流。
这是内核开发者相互切磋和并肩前进的最好方法。
而Robert在书中对内核生活中至关重要的文化和技巧都做了精彩介绍。
请学习和欣赏Robert的书吧。
想必你也希望能精益求精，继续探索，成为内核开发社区中的一员，那么首先你要清楚的是：社区欢迎你。
我们评价和衡量一个人是根据他所作的贡献，当你投身于Linux时，你要明白：虽然你仅仅贡献了一小份力，但马上就会有数千万或上亿人受益。
这是我们的欢乐之源，也是我们的责任之本。

<<Linux内核设计与实现>>

内容概要

《Linux内核设计与实现（原书第3版）》详细描述了Linux内核的设计与实现。
内核代码的编写者、开发者以及程序开发人员都可以通过阅读本书受益，他们可以更好理解操作系统原理，并将其应用在自己的编码中以提高效率和生产率。

本书详细描述了Linux内核的主要子系统和特点，包括Linux内核的设计、实现和接口。
从理论到实践涵盖了Linux内核的方方面面，可以满足读者的各种兴趣和需求。

作者Robert Love是一位Linux内核核心开发人员，他分享了在开发Linux
2.6内核过程中颇具价值的知识和经验。
本书的主题包括进程管理、进程调度、时间管理和定时器、系统调用接口、内存寻址、内存管理和页缓存、VFS、内核同步、移植性相关的问题以及调试技术。
同时本书也涵盖了Linux
2.6内核中颇具特色的内容，包括CFS调度程序、抢占式内核、块I/O层以及I/O调度程序。

《Linux内核设计与实现（原书第3版）》新增内容包括：
增加一章专门描述内核数据结构
详细描述中断处理程序和下半部机制
扩充虚拟内存和内存分配的内容
调试Linux内核的技巧
内核同步和锁机制的深度描述
提交内核补丁以及参与Linux内核社区的建设性建议

作者简介

Robert Love是一位资深的开源社区达人，很早就开始使用Linux。
目前他是Google公司高级软件工程师，是开发Android移动平台内核的团队成员；他曾在Novell公司任职Linux桌面系统的首席架构师；他之前也曾是MontaVista和Ximain公司的内核开发工程师。
他参与的内核项目包括抢占式内核、进程调度器、内核事件层、通知机制、VM改进，以及设备驱动程序。
他是《Linux journal》杂志的编辑。
另外他还著有《Linux System Programming》和《Linux in a Nutshell》。

陈莉君，西安邮电学院教授，十多年来一直致力于推动Linux在中国的发展，多年从事Linux内核的教学和研究，并积极跟踪Linux内核的发展动向，对Linux内核版本的不断演化有着深刻的理解。
著译作品有《Linux操作系统原理与应用》、《Linux操作系统内核分析》、《深入分析Linux内核源代码》、《深入理解Linux内核》和《Linux内核编程》等。

<<Linux内核设计与实现>>

书籍目录

章节摘录

不知不觉涉足Linux内核已经十多个年头了，与其他有志（兴趣）于此的朋友一样，我们也经历了学习—实用—追踪—再学习的过程。
也就是说，我们也是从漫无边际到茫然无措，再到初窥门径，转而觉得心有戚戚焉这一路走下来的。
其中甘苦，犹然在心。
Linux最为人称道的莫过于它的自由精神，所有源代码唾手可得。
侯捷先生云：“源码在前，了无秘密。
”是的，但是我们在面对它的时候，为什么却总是因为这种规模和层面所造就的陡峭学习曲线陷入困顿呢？
很多朋友就此倒下，纵然Linux世界繁花似锦，纵然内核天空无边广阔，但是，眼前的迷雾重重，心中的阴霾又怎能被阳光驱散呢？
纵有雄心壮志，拔剑四顾心茫然，脚下路在何方？
Linux内核入门是不容易，它之所以难学，在于庞大的规模和复杂的层面。
规模一大，就不易现出本来面目，浑然一体，自然不容易找到着手之处；层面一多，就会让人眼花缭乱，盘根错节，怎能让人提纲挈领？
“如果有这样一本书，既能提纲挈领，为我理顺思绪，指引方向，同时又能照顾小节，阐述细微，帮助我们更好更快地理解STL源码，那该有多好。
”孟岩先生如此说，虽然针对的是C++，但道出的是研习源码的人们共同的心声。
然而，Linux源码研究的方法却不大相同。
这还是由于规模和层面决定的。
比如说，在语言学习中，我们可以采取小步快跑的方法，通过一个个小程序和小尝试，就可以取得渐进的成果，就能从新技术中有所收获。
而掌握Linux呢？
如果没有对整体的把握，即使你对某个局部的算法、技术或是代码再熟悉，也无法将其融入实用。
其实，像内核这样的大规模的软件，正是编程技术施展身手的舞台（当然，目前的内核虽然包含了一些面向对象思想，但还不能让C++一展身手）。
那么，我们能不能做点什么，让Linux的内核学习过程更符合程序员的习惯呢？
RobertLove回答了这个问题。
RobertLove是一个狂热的内核爱好者，所以他的想法自然贴近程序员。
是的，我们注定要在对所有核心的子系统有了全面认识之后，才能开始自己的实践，但却完全可以舍弃细枝末节，将行李压缩到最小，自然可以轻装快走，迅速进入动手阶段。
因此，相对于DanielP.Bovet和MarcoCesati的内核巨著《UnderstandingtheLinuxKernel》，它少了五分细节；相对于实践经典《LinuxDeviceDrivers》，多了五分说理。
可以说，本书填补了Linux内核理论和实践之间的鸿沟，真可谓“一桥飞架南北，天堑变通途”。
就我们的经验，内核初学者（不是编程初学者）可以从本书着手，对内核各个核心子系统有个整体把握，包括它们提供什么样的服务，为什么要提供这样的服务，又是怎样实现的。
而且，本书还包含了Linux内核开发者在开发时需要用到的很多信息，包括调试技术、编程风格、注意事项等。
在消化本书的基础上，如果你侧重于了解内核，可以进一步研究《UnderstandingtheLinuxKernel》和源代码本身；如果你侧重于实际编程，可以研读《LinuxDeviceDrivers》，直接开始动手工作；如果你想有一个轻松的内核学习和实践环节，请访问我们的网站www.kerneltravel.net。
Linux内核是一艘永不停息的轮船，它将驶向何方我们并不知晓，但在这些变化的背后，总有一些原理是恒定不变的，总有一些变化是我们想知晓的，比如调度程序的大幅度改进，内核性能的不断提升，本书第3版虽然针对的是较新的2.6.34Linux内核版本，但在旧版本上积累的知识和经验依然有效，而新增内容将使读者在应对变化了的内核代码时更加从容。
感谢牛涛和武特，他们在第2版和第3版差异的校对中花费了大量精力。

感谢素不相识的网友ChengRenquan，他主动承担了其中一章的修订。
还要感谢苏锦绣、黄伟、王泽宇、赵格娟、刘周平、周永飞、曹江峰、陈白虎和孟阿龙，他们参与了后期的校对和查错补漏。
最后，特别感谢我的合作者康华，从十年前一块分析Linux内核代码到今天，他对技术孜孜不倦的追求不但在业界赢得声誉，也使我们在翻译过程中所遇到的技术难点和晦涩句子被一一迎刃而解。
感谢合作者张波，他流畅有趣的文笔让本书少了份枯燥，多了份趣味。
陈莉君2010年10月前言在我刚开始有把自己的内核开发经验集结成册，撰写一本书的念头时，我其实也觉得有点头绪繁多，不知道该从何下手。
我实在不想落入传统内核书籍的窠臼，照猫画虎地再写这么一本。
不错，前人著述备矣，但我终归是要写出点儿与众不同的东西来，我的书该如何定位，说实话，这确实让人颇费思量。
后来，灵感终于浮现出来，我意识到自己可以从一个全新的视角看待这个主题。
开发内核是我的工作，同时也是我的嗜好，内核就是我的挚爱。
这些年来，我不断搜集与内核有关的奇闻轶事，不断积攒关键的开发诀窍，依靠这些日积月累的材料，我可以写一本关于开发内核该做什么—更重要的是—不该做什么的书籍。
从本质上说，这本书仍旧是描述Linux内核是如何设计和实现的，但是写法却另辟蹊径，所提供的信息更倾向于实用。
通过《Linux内核设计与实现（原书第3版）》，你就可以做一些内核开发的工作了—并且是使用正确的方法去做。
我是一个注重实效的人，因此，这是一本实践的书，它应当有趣、易读且有用。
我希望读者可以从这本书中领略到更多Linux内核的精妙之处（写出来的和没写出来的），也希望读者敢于从阅读本书和读内核代码开始跨越到开始尝试开发可用、可靠且清晰的内核代码。
当然如果你仅仅是兴致所至，读书自娱，那也希望你能从中找到乐趣。
从第1版到现在，又过了一段时间，我们再次回到本书，修补遗憾。
本版比第1版和第2版内容更丰富：修订、补充并增加了新的内容和章节，使其更加完善。
本版融合了第2版以来内核的各种变化。
更值得一提的是，Linux内核联盟做出决定，近期内不进行2.7版内核的开发，于是，内核开发者打算继续开发并稳定2.6版。
这个决定意味深长，而本书从中的最大受益就是在2.6版上可以稳定相当长时间。
随着内核的成熟，内核“快照”才有机会能维持得更久远一些。
本书可作为内核开发的规范文档，既认识内核的过去，也着眼于内核的未来。
使用这本书开发Linux内核不需要天赋异秉，不需要有什么魔法，连Unix开发者普遍长着的络腮胡子都不一定要有。
内核虽然有一些有趣并且独特的规则和要求，但是它和其他大型软件项目相比，并没有太大差别。
像所有的大型软件开发一样，要学的东西确实不少，但是不同之处在于数量上的积累，而非本质上的区别。
认真阅读源码非常有必要，Linux系统代码的开放性其实是弥足珍贵的，不要无动于衷地将它搁置一边，浪费了大好资源。
实际上就是读了代码还远远不够呢，你应该钻研并尝试着动手改动一些代码。
寻找一个bug然后去，改进你的硬件设备的驱动程序。
增加新功能，即使看起来微不足道，寻找痛痒之处并解决。
只有动手写代码才能真正融会贯通。
内核版本《Linux内核设计与实现（原书第3版）》基于Linux2.6内核系列。
它并不涵盖早期的版本，当然也有一些例外。
比如，我们会讨论2.4系列内核中的一些子系统是如何实现的，这是因为简单的实现有助于传授知识。
特别说明的是，本书介绍的是最新的Linux2.6.34内核版本。
尽管内核总在不断更新，任何努力也难以捕获这样一只永不停息的猛兽，但是本书力图适合于新旧内

核的开发者和用户。
虽然本书讨论的是2.6.34内核，但我也确保了它同样适用于2.6.32内核。
后一个版本往往被各个Linux发行版本奉为“企业版”内核，所以我们可以在各种产品线上见到其身影。
该版本确实已经开发了数年（类似的“长线”版本还有2.6.9、2.6.18和2.6.27等）。
读者范围《Linux内核设计与实现（原书第3版）》是写给那些有志于理解Linux内核的软件开发者的。
本书并不逐行逐字地注解内核源代码，也不是指导开发驱动程序或是内核API的参考手册（如果存在标准的内核API的话）。
本书的初衷是提供足够多的关于Linux内核设计和实现的信息，希望读过本书的程序员能够拥有较为完备的知识，可以真正开始开发内核代码。
无论开发内核是为了兴趣还是为了赚钱，我都希望能够带领读者快速走进Linux内核世界。
本书不但介绍了理论而且也讨论了具体应用，可以满足不同读者的需要。
全书紧紧围绕着理论联系实践，并非一味强调理论或是实践。
无论你研究Linux内核的动机是什么，我都希望这本书都能将内核的设计和实现分析清楚，起到抛砖引玉的作用。
因此，本书覆盖了从核心内核系统的应用到内核设计与实现等各方面的内容。
我认为这点很重要,值得花工夫讨论。
例如,第8章讨论的是所谓的下半部机制。
本章分别讨论了内核下半部机制的设计和实现（核心内核开发者或者学者会感兴趣），随即便介绍了如何使用内核提供的接口实现你自己的下半部（这对设备驱动开发者可能很有用处）。
其实，我认为上述两部分内容是相得益彰的，虽然核心内核开发者主要关注的问题是内核内部如何工作，但是也应该清楚如何使用接口；同样，如果设备驱动开发者了解了接口背后的实现机制，自然也会受益匪浅。
这好比学习某些库的API函数与研究该库的具体实现。
初看，好像应用程序开发者仅仅需要理解API—我们被灌输的思想是，应该像看待黑盒子一样看待接口。
另外，库的开发者也只关心库的设计与实现，但是我认为双方都应该花时间相互学习。
能深刻了解操作系统本质的应用程序开发者无疑可以更好地利用它。
同样，库开发者也决不应该脱离基于此库的应用程序，埋头开发。
因此，我既讨论了内核子系统的设计，也讨论了它的用法，希望本书能对核心开发者和应用开发者都有用。
我假设读者已经掌握了C语言，而且对Linux比较熟悉。
如果读者还具有与操作系统设计相关的经验和其他计算机科学的概念就更好了。
当然，我也会尽可能多地解释这些概念，但如果你仍然不能理解这些知识的话，请看本书最后参考资料中给出的一些关于操作系统设计方面的经典书籍。
本书很适合在大学中作为介绍操作系统的辅助教材，与介绍操作系统理论的书相搭配。
对于大学高年级课程或者研究生课程来说，可直接使用本书作为教材。
第3版致谢与其他作者一样，我决非是一个人躲在山洞里孤苦地写出这本书来的（那也是一件美差，因为有熊相伴）。
我能最终完成本书原稿是与无数建议和关怀分不开的。
仅仅一页纸无法容纳我的感激，但我还是要衷心地感谢所有给予我鼓励、给予我知识和给予我灵感的朋友和同事。
首先我要对为《Linux内核设计与实现（原书第3版）》付出辛勤劳作的所有编辑表示感谢，尤其要感谢我的组稿编辑MarkTaber，他为这一版的出版从头到尾倾注了许多心血。
还要特别感谢业务开发编辑MichaelThurston和项目组织编辑TonyaSimpso。
本版的技术编辑RobertP.J.Day也是我需要倍加感谢的人，他独到的洞察力和准确的校对使书稿质量大大提高。

尽管他的工作称得上完美，但如果书中仍留有错误，责任由我承担。
需要十分感谢的还有AdamBelay、ZackBrown、MartinPool以及ChrisRivera，他们对第1版和第2版所做的一切努力我依然记忆犹新。
许多内核开发者为我提供了大力支持，回答了许多问题，还有那些撰写代码的人——本书正是由于有了这些代码，才有了存在的意义。
他们是：AndreaArcangeli、AlanCox、GregKroah-Hartman、DaveMiller、PatrickMochel、AndrewMorton、NickPiggin和LinusTorvalds。
我要大力感谢我的同事们。
他们的创造力和智慧无与伦比，能与他们一起工作其乐无穷。
因为篇幅原因，请谅解我不能列出所有人的名字。
但不得不提的是：AlanBlount、JayCrim、ChrisDanis、ChrisDiBona、EricFlatt、MikeLockwood、SanMehat、BrianRogan、BrianSwetland、JonTrowbridge和SteveVinter，感谢他们给予我的支持、知识和友谊！
还有许多值得尊敬和热爱的人，他们是：PaulAmici、MikeyBabbitt、KeithBarbag、JacobBerkman、NatFriedman、DustinHall、JoyceHawkins、MigueldeIcaza、JimmyKrehl、DorisLove、LindaLove、BretteLuck、RandyO扯owd、SalRibaudo和他了不起的妈妈ChrisRivera、CarolynRodon、JoeyShaw、SarahStewart、JeremyVanDoren和他的家人、LuisVilla、SteveWeisberg和他的家人以及HelenWhisnant等。

媒体关注与评论

能够把linux内核的内容在300多页内叙述一遍，本身就是一件高难度的事情。
但《Linux内核设计与实现》确实做到了。
《Linux内核设计与实现》很少涉及具体实现，而是把握思想，讲解算法，读者可以学习到linux内核的知识，而不用纠缠于具体细节。
—豆瓣读者Googol，《Linux内核设计与实现》很适合系统学习了OS理论之后直接看代码详解又觉得暂且还不够功力的读者，它可以带你由理论学习阶段逐渐过渡到实践阶段。
这本书对Linux内核内容的范围和深度把握得恰到好处。
——豆瓣读者 纳兰经若

编辑推荐

《Linux内核设计与实现(原书第3版)》编辑推荐：畅销图书新版《Linux内核设计与实现》第3版翻译版、影印版同步上市，《Linux内核设计与实现(原书第3版)》详细描述Linux内核的主要子系统和特点，《Linux内核设计与实现(原书第3版)》涵盖Linux内核从理论到实践的方方面面。

<<Linux内核设计与实现>>

版权说明