华清远见 FARSIGHT 嵌入式培训专家

一本揭秘Linux内核学习方法的图书

Linux内核修炼秘籍大揭秘
每天万余人争相阅读

Linux 内核修炼之道

CSDN Blog专家
fudan_abc
最新力作

华清远见嵌入式培训中心 任桥伟 编著

十余位Linux内核专家联合推荐（排名不分先后）

- Chris DiBona（谷歌开源程序经理 开源软件大师）
- 吴雄昂（ARM中国区总经理）
- 栾跃（谷歌开发技术推广部 中国市场主管经理）
- 郭淳学（中国软件行业协会嵌入式系统分会副理事长兼秘书长）
- 何小庆（北京麦克泰软件技术有限公司董事长）
- 马忠梅（北京理工大学副教授）
- 史应生（红帽专家服务团队 资深咨询顾问，RHCA、RHCSS 中国第一人）
- 陈莉君（西安邮电学院教授）
- 向农（电子工程世界总编）
- 陈渝（清华大学副教授）
- 伍鹏（Linux Kernel Developer）
- 李泽帆（富士通南大软件技术有限公司）

《Linux内核修炼之道》讲座视频、PPT免费下载

人民邮电出版社
POSTS & TELECOM PRESS

Linux

内核修炼之道

华清远见嵌入式培训中心 任桥伟 编著

人民邮电出版社
北京

图书在版编目（CIP）数据

Linux内核修炼之道 / 华清远见嵌入式培训中心编著
. -- 北京 : 人民邮电出版社, 2010.7
ISBN 978-7-115-22585-6

Ⅰ. ①L… Ⅱ. ①华… Ⅲ. ①Linux操作系统 Ⅳ.
①TP316.89

中国版本图书馆CIP数据核字(2010)第048635号

内 容 提 要

本书重点介绍 Linux 2.6.23 内核的工作原理以及学习方法。将 Linux 内核的修炼之道总结为四个层次："全面了解抓基本，兴趣导向深钻研，融入社区做贡献，坚持坚持再坚持"。第一层次目的是对 Linux 以及内核有个全面的认识和了解，掌握 Linux 内核源代码的分析方法。第二个层次讨论了内核中系统初始化、系统调用、中断处理、进程管理及调度、内存管理、文件系统以及设备驱动等主要部分，目的是希望读者以兴趣为导向，寻找一个子系统或模块，对其代码深入钻研和分析。第三个层次介绍了内核开发与调试的一些基本信息，目的是希望读者能够融入到内核的开发社区，做出自己的贡献。第四个层次浓缩为两个字——坚持。

本书语言通俗易懂，内容覆盖了内核的学习方法到内核设计与实现等各方面内容，能够带领读者快速走入 Linux 内核的世界，适合对 Linux 内核学习茫然的初学者，也适合各类希望深入理解 Linux 内核的读者。

Linux 内核修炼之道

◆ 编　　著　华清远见嵌入式培训中心　任桥伟
责任编辑　黄　焱

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
三河市海波印务有限公司印刷

◆ 开本：787×1092　1/16
印张：23
字数：537 千字　　2010 年 7 月第 1 版
印数：1 – 5 000 册　　2010 年 7 月河北第 1 次印刷

ISBN 978-7-115-22585-6

定价：49.00 元

读者服务热线：(010)67132692　印装质量热线：(010)67129223
反盗版热线：(010)67171154

序

曾几何时，那只黄嘴巴黄脚蹼的企鹅走进了我们的生活，不经意间 Linux 已经在 IT 行业得到了广泛的应用。

要充分地了解 Linux，我们需要走进 Linux 内核中去，通过对 Linux 内核的学习与研究，了解 Linux 是如何运作的，并从 Linux 内核的源码中体会到代码的艺术。

Linux 内核是庞大复杂的，初学者大多会不自觉地迷失在 Linux 内核的迷宫里。为了更好地帮助读者深入了解 Linux，任老师编写了本书，本书以一种独特的视角，将 Linux 内核的学习划分为以下 4 个层次：

- 全面了解，掌握基本功；
- 兴趣导向，选择重点深度钻研；
- 融入社区，参与开发做贡献；
- 坚持，坚持，再坚持。

本书为每个层次都设立了特定的目标与修炼的方法，从而为广大的 Linux 内核学习者打开了那扇原本掩着的门。希望广大读者能够从这本书中有所收获，在任老师幽默而犀利的文笔带领下走进 Linux 内核的世界。

作为《Linux 内核修炼之道》一书的策划和组稿者，我们希望通过本书的出版，让国内的众多 Linux 内核技术爱好者能够有机会跟随业内知名专家一起全面剖析、构建嵌入式 Linux 内核开发，了解大量 Linux 内核的技术内幕。

华清远见教育集团总裁　季久峰

Preface

Open Source and Innovation

Google wouldn't be the company it is today without Open Source Software .As a leading Internet technology company, we heavily use and release Open Source code into existing communities and creating new projects like Chrome, Android and others. As such, we value and try to encourage Open Source development and we see the participatory model of technology development as a key factor in the development of the Internet and in computer science.

I congratulate Farsight on the publishing of "The Zen of Linux Internals" and it is my hope that this book will foster further interest in Linux and Open Source among the developer communities in China. I look forward to seeing what new Chinese open source developers there have in store for the us all.

Best wishes to you all, developers in China!

Chris DiBona
Open Source Program Manager
Google
May 24th, 2010

推荐序

移动与云计算的发展推动了越来越多的新技术、新应用和新产品的涌现，推动了嵌入式电子产品世界的不断更新和快速发展。作为嵌入式行业最著名的厂商之一，20 多年来 ARM 除了不断地加大研发投资，开发最新的微处理器、图形技术、物理 IP 和开发工具，为产业升级搭建了最佳的开发架构；同时，也一直致力于建设一个开放的、具有强大生命力和发展前景的 ARM 嵌入式生态系统，使得每个存在于这个生态系统的成员都能发挥各自的特长，通过有效的产业分工和协作开发出高性能、低功耗、人性化的嵌入式产品服务于广大的消费者。

在这个生态系统中，嵌入式操作系统是必不可少的重要环节，是“链接”底层硬件和上层应用软件的纽带。其中 Linux 作为开源的嵌入式操作系统，多年来一直受到广大工程师朋友的喜爱，特别是在基于 Linux 内核的 Android 操作系统发布以来，Linux 的应用和发展到了一个崭新的高度。ARM 作为应用最广泛的嵌入式处理器，对 Linux 操作系统的发展也做出了大量的支持与贡献。

本书从 Linux 历史开始，深入浅出地讲述了 Linux 内核裁减和编译，系统启动和调用，中断处理，进程管理和调度，内存管理，文件系统，设备驱动和内核开发调试等，为有志于学习 Linux 内核相关领域的工程师朋友们提供了指导和素材。

吴雄昂
ARM 中国区总经理
2010 年 6 月

推荐序

利用开源代码软件的优势推动创新

开源代码作为软件开发的一种模式，具有与此相对立的基于某一家公司技术的封闭式代码软件开发模式所没有的、无法替代的优势。利用开源代码，这些优势能够对技术的发展和创新带来极大的生机。

开源代码的优势非常明显，主要如下。

（1）技术和知识的分享有利于创新。

开源代码的中心理念是技术和知识的分享。任何开发者和公司可以用开源代码的软件作为基础进行继续的开发，举一反三地在前人的基础上继续进行技术的提高和创新，有利于不断地创新。

（2）对软件和技术的选择自由、灵活。

开发技术不受某一家公司垄断性的控制和锁定，技术的使用者也不受限制于某家供应商。开发者还可以自由地对需要使用的技术进行整合、加以利用。

（3）低成本。

开源代码的免费的源代码分布模式让开发者们不用付所谓的软件使用授权费而使用代码，另外，有大量志愿者参与的社区型开发和维护模式也意味着开发上的低成本，因此它是一个很经济的开发模式。

（4）技术的透明带来的良好安全性和可靠性。

由开发者社区参与的对代码的维护和更新保证了技术的透明性，有利于有效地发现各种安全漏洞和解决各种隐患，提高代码的稳定性、可靠性以及总体的质量。

（5）技术的透明带来的可审计性

开源代码的另一个独特优点是：使用者可以通过对公布的源代码进行审核建立起使用的信任和信心。而封闭式的商业软件由于看不到源代码，各种性能或安全究竟如何，是否有安全漏洞或后门，无法验证供应商的任何说法，只能听任其说。

（6）高效的技术使用和部署。

在开源代码基础上进行的产品开发，可以进行高速的技术验证和原型开发，敏捷地进行产品的阶段性开发、测试、部署，不再需要等待某个垄断性的供应商的产品发行周期。开源的透明性又能帮助公司或机构之间进行高效的开发协作，提高部署和使用的效率。

（7）社区化的技术支持和参与的环境。

互联网上大量的开源代码社区和论坛，提供了一个世界范围内的互相支持的环境。开发者们通过社区的协作和公开的讨论，还可以影响和推动各种技术的发展以及相应的公开标准。

开源代码在中国软件开发业界的使用目前还不是很广泛，但是世界上一些领先的技术公

司，特别是谷歌公司，已经在利用开源代码的模式推动互联网技术的发展上取得了巨大的成果。

谷歌公司不仅利用开源代码作为自己很多产品开发的基础，例如采用基于开源代码的 Web Kit 作为 Chrome 浏览器的引擎，使 Chrome 浏览器具备了领先业界的高速运行的性能；谷歌公司还将自己正在开发的新一代产品通过开源代码回馈给开发者社区，例如把未来的 Chrome OS 操作系统通过 Chromium 项目，将整个操作系统产品的设计构架和代码通过开源模式与开发者们进行分享，以此来推动更多的开发者利用它进行更多的创新，推动网络应用的进一步发展。可以预见，在不远的未来，很多开发者将会在此基础上开发出更多极其富有创意的新型上网本操作系统的产品。

谷歌的另一个与开源代码有着紧密联系的产品是新型的手机和移动设备平台的操作系统 Android。它已经得到了全世界范围的开发者的热烈支持，大量的应用被不停地开发出来。开发者和市场对它的兴趣和支持的程度已经远远超过其他手机操作系统。这些都是开源代码和由开发者社区参与的开发模式对创新带来好处的极好的例子。

华清远见嵌入式培训中心金牌讲师任桥伟所著的《Linux 内核修炼之道》是帮助中国的软件开发业界进一步了解和学习 Linux 和开源代码的力作。希望本书会让更多的中国软件开发者了解 Linux 和开源代码，并参加和融入到世界范围的各个开源代码社区中去，让未来的相关技术的发展有更多的来自中国的开发者的声音和建议，并对 Linux、开源代码、网络技术等在中国的进一步发展起到推动作用。

谷歌 开发技术推广部 中国市场主管经理　栾跃

2010 年 5 月

专家推荐

（排名不分先后）

Linux 内核是 Linux 系统最核心的部分，学习 Linux 内核对于 Linux 开发有重要意义。本书将 Linux 内核的学习分为 4 个层次，以轻松、简洁的语言讲解了 Linux 内核学习的方法，是学习 Linux 内核很好的选择。

——中国软件行业协会嵌入式系统分会副理事长兼秘书长　郭淳学

Linux 内核的浩瀚而深邃相信是我们每个人在学习它时的感悟，开源世界的自由和潇洒也让习惯了传统教育的莘莘学子无法适应。本书内容新颖、文笔简练，是一本让人愿意读并读得懂的 Linux 书籍。

——北京麦克泰软件技术有限公司董事长　《单片机与嵌入式系统应用》杂志副主编　何小庆

《Linux 内核源代码情景分析》只有少数计算机专业高手才能问津。作者坚持、坚持再坚持的内核修炼之心得化作风趣、通俗易懂的语言，能引发广大 Linux 爱好者的学习兴趣。这是本书的价值所在。

——北京理工大学副教授　马忠梅

十年磨一剑，躬行始大成。任老师把自己十年的内核学习心得融入这本力作中。由浅入深，有理论更重实践；涉及全面，更有深入解剖。从这本书我们得到的不仅是知识，更是一种迈向大师的修炼之道。

——红帽专家服务团队 资深咨询顾问，RHCA、RHCSS 中国第一人　史应生

“修炼”需要一个场景，如果是一马平川，十年修炼也只是耗费时间而已。如果你有幸踏入 Linux 内核的修炼场，本书风趣贴心的表达方式、独具匠心的引领地图，无疑让你在修炼的路途中，少走弯路并多份坚持。

——西安邮电学院教授　陈莉君

不积跬步，无以致千里；不读内核，无以成高手。本书是作者多年 Linux 内核学习心得的精华集粹，它扬弃了教科书式的说教，从资深工程师的角度，引领初学者走过内核编程的入门阶段。深入浅出，娓娓道来，在不知不觉中引领你攀登系统内核的科学高峰，向高手的境界迈进。

——电子工程世界总编　向农

《Linux 内核修炼之道》用比较轻松的叙述，通过分析最新的 Linux 源代码，解剖了内核启动、系统调用、中断处理、进程管理、调度、内存管理、文件系统和设备驱动等主要操作系统组成的设计原理和实现细节，并结合一定的动手实践的例子来增强对上述实现的理解。对广大 Linux 爱好者而言，是一本值得推荐的好书。

——清华大学计算机系副教授　陈渝

我刚开始知道 fudan_abc 是由于在网络上盛传其 Linux 内核的博客《Linux 内核修炼之道》。看过之后，非常惊讶，发现 Linux 内核相关的枯燥技术话题居然可以写得如此生动有趣。过了一些日子，发现书籍出版在即，感觉内容更加丰富。

我一直从事 Linux 内核开发，看过很多 Linux 内核的书籍，大都是技术性非常强的，而且很多不太适合初学者阅读。《Linux 内核修炼之道》用当下非常流行的诙谐调侃的语气，从实际的开发经验入手，介绍了整个 Linux 内核的方方面面，可以看到作者的技术功底和对 Linux 内核开发非常强烈的热爱之情。

这本书我觉得非常适合 Linux 内核开发者参考，并且很适合初学者入门学习。

——Linux Kernel Developer　伍鹏（Bryan Wu）

近年来 Linux 内核邮件列表中的中国人面孔可谓屡见不鲜，但由于内核越来越复杂，学习曲线越来越陡峭，又加上语言障碍和文化差异，新手，特别是业余爱好者，面对内核和内核社区常常觉得望其门而不得入。对此，这本书当有其价值。

——富士通南大软件技术有限公司　Linux 内核开发工程师　李泽帆

前　言

至此落笔之际，恰至 Linux 问世 18 周年，18 年的成长，如梦似幻，风雨颇多，感慨颇多。

犹自忆起多年以前一位前辈训导时的箴言：今天的必然正是由之前一系列的偶然所决定的。过去的某年某月，我偶然初识 Linux 就身陷其中，至今仍找不到出去的路。那么，当你偶然地拿起这本书，偶然地看到这段话，你是否会问自己：这样的偶然又会导致什么样的必然？

如果你依然决定继续这次的偶然之旅，那么首先请认识一个人和一只企鹅。这个人自然就是 Linus Torvalds，我们称他为 Linus 或李纳斯，正是这位来自芬兰的天才，在 1991 年 1 月 2 日，攒着在圣诞节和生日得到的钱，偶然地做出了一个重大的财政决定，分期 3 年买一台价格为 3500 美元的相貌平平的计算机，从而 Linux 开始了。

企鹅则是 Linux 的标志，很多人可能不知道 Linus，但是却可能知道这只企鹅，这是一个奇怪的现象，就像很多人知道微软，却不知道比尔盖茨。不管怎么说，是 Linus 塑造了这只企鹅，并让它有一副爽透了的样子，就像刚刚吞下一扎啤酒。除此之外，这只企鹅还要很特别，其他的企鹅都是黑嘴巴黑脚蹼，但它却是黄嘴巴黄脚蹼。

在继续阅读之前，我还想问一个问题：你在强迫自己学习内核吗？我很希望你能回答不是，但希望与现实往往都有段不小的距离，因为很多时候，我会发现身边的人是因为觉得内核很高深而强迫自己喜欢的。强迫自己去喜欢一个人是多么痛苦的事情。或许，针对这个问题，最让人愉悦的回答是“说实话，我学习的热情从来都没有低落过”。正如 Linus 在自己的自传《Just for Fun》中希望的那样。

本书的组织形式

本书将 Linux 内核的学习分为 4 个层次：全面了解，掌握基本功；兴趣导向，选择重点深度钻研；融入社区，参与开发做贡献；坚持，坚持，再坚持。总结起来，就是“全面了解抓基本，兴趣导向深钻研；融入社区做贡献，坚持坚持再坚持。”（如果你是一个修真小说爱好者，可以将其与炼气、筑基、结丹和元婴等层次相对应）。

第一层次修炼的内容包括了前 3 章，目的是希望你能够对 Linux 以及内核有个全面的认识和了解，掌握分析 Linux 内核源代码的分析方法。

第 1 章主要介绍了 Linux 的 18 年成长史，或许你会乐意陪我一起缅怀这过去的 18 年。

第 2 章介绍内核的配置和编译过程，和任何大型软件源码的学习一样，学会编译和配置是第一步。

第 3 章介绍学习内核需要的基础，内核的体系结构、目录结构、代码特点，浏览内核代码的工具，最后，突出强调了内核源码分析过程中极为重要的两个角色——Kconfig 和

Makefile，并以 USB 子系统为例，演示了如何利用这两个角色进行代码分析。

第二层次的修炼包括了第 4～11 章，对内核多数部分的工作原理进行介绍。按照认识的发展规律，在第一层次修炼中已经对内核有个全局的认识和了解之后，接下来就应该以兴趣为导向，寻找一个子系统或模块，对其代码深入钻研和分析，不懂的地方就通过社区、邮件列表或者直接发 E-mail 给 maintainer 请教等途径弄懂，切勿得过且过，这样分析下来，对同步、中断等内核机制的掌握有很大好处，一通则百通就是这个道理。

因此第二层次的各个章节里，只是阐释重点的概念和工作原理，帮助你在分析该部分代码时进行理解，并不求详尽。

第 4 章讲解系统的初始化，万事开头难，系统的初始化是一个很复杂的过程，不过对于内核源码的学习来说，以这个部分开始应该是个不错的选择。特别是子系统初始化，应该是你选择任何内核子系统开始分析时都需要了解的内容。

第 5 章讲解系统调用，它是应用程序和内核间的桥梁，学习并理解它是我们走向内核的一个很好的过渡。

第 6 章讲解内核的中断处理机制，其中讲解了其他内核书籍都没有涉及的通用 IRQ 层。

第 7 章讲解进程的内核抽象，以及进程如何被创建和销毁。如果我们将计算机上运行的操作系统以及各种各样的软件看作一系列有机的生命体，而不是死的指令集合，那么这就是一个进程的世界，只不过与我们人类世界不同的是，进程世界里的个体是一个一个鲜活的进程，而不是人。人的世界有道德与法律去制约管理，进程的世界同样也有自己的管理机制，这就是第 7 章所要讲解的内容——进程管理。

第 8 章讲解进程的调度，重点讲解了在内核历史上具有重要地位的 O（1）调度器和最新的 CFS 调度器。

第 9 章讲解内存管理，内存就是进程的家，这里讲解内核如何为每个进程都分配一个家，并尽量地去做到“居者有其屋”，以及保证每个家的安全。

第 10 章讲解文件系统，主要是虚拟文件系统（VFS），它通过在各种具体的文件系统之上建立一个抽象层，屏蔽了不同文件系统间的差异。

第 11 章讲解设备驱动，对于驱动开发来说，设备模型的理解是根本，spec、datasheet 与内核源代码的利用是关键。

通过第二层次的修炼，你应该对至少一～两个部分有了很深入的理解，对内核代码采用的通用手法也已经很熟悉，那么你应该开始进入第三层次，努力融入内核开发社区，此时的你已经不再是社区中潜水的小白，已经可以针对某个问题发表自己的见解。你已经可以尝试参与到内核的开发中去，即使仅仅修改了内核中的一个错误单词，翻译了一份大家需要的文档，也是做出了自己的贡献，会得到大家的认可。

本书中第三层次只包括了两章内容，这是因为内核的修炼之道越往后便越依赖于自己，任何参考书都替代不了自己的不断反思与总结。

第 12 章讲解参与内核开发需要了解的一些基础信息。

第 13 章讲解了内核的调试技术，与第 12 章一样，你可以仅仅将这些内容看成内核修炼中的一些 tips。

至于最后的第四层次，更是仅有两个字——坚持。能够在内核的修炼之道上走多远，取

决于我们能够坚持多久，或许用一个公式概括更为合适：心态+兴趣+激情+时间+X＝Y。

革命尚未成功，我等仍需努力——与君共勉之。

感谢

如果没有无数的建议和关怀，这本书不可能完成，谨在此感谢所有给予我鼓励和帮助的朋友和亲人。

首先要感谢责任编辑黄焱，是他鼓励我开始写作此书，并且在撰写目录到定稿的整个过程中给予我无私的帮助和指导。

其次要感谢在技术上给予我指导和帮助的老师和朋友，他们是杨树堂、冯启德、陈高鹏、王子祥、宋宝华、缪峥、吴赫、肖季东、肖林甫等。

再次要感谢所有对 Linux 抱有兴趣或从事 Linux 工作的人，没有你们提供的大量技术资料，本书便会成为无源之水。

最后，我要感谢我的父母和妹妹，感谢他们的爱。

我的博客地址是 http://blog.csdn.net/fudan_abc，欢迎大家访问、交流。

编　者

2010 年 6 月

目　录

第 1 章　初识 Linux ……1

1.1　Linux 史记 ……2
1.1.1　Linux 诞生记 ……2
1.1.2　泰坦尼克的狂潮 ……2
1.1.3　提前发生的革命 ……3
1.1.4　和平、爱情和 Linux ……4
1.1.5　Ubuntu 4.10 ……4
1.1.6　Richard Stallman 的征婚启事 ……5
1.1.7　一封公开信 ……6
1.1.8　首款 Android 手机 ……8
1.1.9　Linux 信用卡 ……8
1.2　内核的版本 ……9
1.3　获取内核源码 ……10
1.4　Linux 发行版 ……11
1.4.1　软件包管理器 ……11
1.4.2　流行的发行版 ……12
1.4.3　选择 Linux 发行版 ……13

第 2 章　配置与编译内核 ……15

2.1　配置内核 ……16
2.1.1　几种配置方式 ……16
2.1.2　.config 文件 ……19
2.1.3　配置选项详解 ……20
2.2　编译内核 ……26
2.2.1　准备工作 ……26
2.2.2　如何为内核打补丁 ……27
2.2.3　编译步骤 ……28
2.2.4　文档的编译 ……29
2.2.5　编译小技巧 ……29
2.3　自由软件的编译与安装 ……30
2.3.1　发布时的组织结构 ……30
2.3.2　配置 ……30
2.3.3　编译 ……32
2.3.4　安装 ……35

第 3 章　浏览内核源代码 ……37

3.1　内核学习的技术基础 ……38
3.2　内核体系结构 ……38
3.3　内核源码目录结构 ……40
3.4　浏览代码的工具 ……43
3.4.1　Source Insight ……43
3.4.2　Vim+Cscope ……43
3.4.3　LXR ……45
3.5　内核代码的特点 ……46
3.5.1　GCC 扩展 ……46
3.5.2　内嵌汇编 ……51
3.6　内核中的链表 ……51
3.7　Kconfig 和 Makefile ……53
3.7.1　Kconfig 结构 ……54
3.7.2　利用 Kconfig 和 Makefile 寻找目标代码 ……55
3.8　代码分析示例 ……57
3.8.1　分析 README ……57
3.8.2　分析 Kconfig 和 Makefile ……58
3.8.3　寻找初始化函数 ……60

第 4 章　系统初始化 ……64

4.1　引导过程 ……65
4.2　内核初始化 ……67
4.2.1　start_kernel 函数 ……68
4.2.2　reset_init 函数 ……71
4.2.3　kernel_init 函数 ……72
4.2.4　init_post 函数 ……73
4.3　init 进程 ……75

4.4 内核选项解析……75
4.4.1 内核选项……75
4.4.2 注册内核选项……77
4.4.3 两次解析……78
4.5 子系统的初始化……79
4.5.1 do_initcalls()函数……79
4.5.2 .initcall.init 节……80
4.5.3 分析示例……81

第 5 章 系统调用……84

5.1 系统调用概述……85
5.1.1 系统调用、POSIX、C 库、系统命令和内核函数……85
5.1.2 系统调用表……87
5.1.3 系统调用号……87
5.1.4 系统调用服务例程……88
5.1.5 如何使用系统调用……88
5.1.6 为什么需要系统调用……90
5.2 系统调用执行过程……91
5.3 系统调用示例……93
5.3.1 sys_dup……93
5.3.2 sys_reboot……94
5.4 系统调用的实现……97
5.4.1 如何实现一个新的系统调用……97
5.4.2 什么时候需要添加新的系统调用……99

第 6 章 中断与中断处理……100

6.1 概述……101
6.1.1 中断分类……101
6.1.2 PIC vs APIC……102
6.1.3 中断号 vs 中断向量……104
6.1.4 中断描述符表……105
6.1.5 门……106
6.1.6 中断服务程序……107
6.2 重要数据结构……107
6.2.1 中断描述符 irq_desc……108
6.2.2 中断控制器描述符 irq_chip……109
6.2.3 中断服务程序描述符 irqaction……110
6.3 中断子系统初始化……111
6.3.1 中断描述符表的初始化……111
6.3.2 中断请求队列的初始化……116
6.4 中断或异常处理……117
6.4.1 中断控制器的工作……118
6.4.2 CPU 的工作……120
6.4.3 内核对中断的处理……120
6.4.4 内核对异常的处理……123
6.5 中断 API……123
6.5.1 注册和释放……123
6.5.2 激活和禁止……125
6.5.3 其他 API 函数……125
6.6 通用 IRQ 层……126
6.6.1 GenIRQ 的起源及发展……126
6.6.2 GenIRQ 的抽象层次……126
6.7 多处理器系统中的中断处理……128
6.7.1 处理器间中断……128
6.7.2 中断亲和力……128
6.7.3 中断负载均衡……129
6.8 中断的下半部……130
6.8.1 下半部的实现机制……130
6.8.2 下半部机制的选择……132

第 7 章 进程管理……133

7.1 概述……134
7.1.1 进程 vs 程序……134
7.1.2 进程 vs 线程……134
7.1.3 进程描述符……135
7.1.4 进程状态……136
7.1.5 进程标识符……137
7.1.6 进程间关系……138
7.1.7 进程 0 与进程 1……140
7.1.8 进程的内核栈……140
7.1.9 获取当前进程……142
7.2 进程创建……142
7.2.1 fork()、vfork()与 clone()……143
7.2.2 do_fork()……146
7.2.3 copy_process()……149

7.2.4 内核线程……150
7.3 进程退出……151
7.3.1 do_exit()……151
7.3.2 僵死进程……152
7.3.3 孤儿进程……153
7.4 后台进程……153

第 8 章 调度……155

8.1 概述……156
8.1.1 调度策略……156
8.1.2 进程调度的目标……157
8.1.3 进程的 nice 值……158
8.1.4 优先级……158
8.1.5 时间片……159
8.2 进程调度器的发展历史……160
8.2.1 Linux 2.4 的调度器……160
8.2.2 O(1)调度器……161
8.2.3 SD 调度器……163
8.2.4 RSDL 调度器……163
8.2.5 CFS 调度器……164
8.3 O(1)调度器……165
8.3.1 运行队列……165
8.3.2 优先级数组……168
8.3.3 计算时间片……169
8.3.4 平均休眠时间……170
8.3.5 判断交互性……170
8.3.6 计算优先级……171
8.3.7 休眠和唤醒……172
8.3.8 schedule()……173
8.3.9 负载均衡……174
8.3.10 软实时调度……176
8.4 CFS 调度器……176
8.4.1 完全公平与进程的权重……176
8.4.2 模块化……177
8.4.3 调度实体……179
8.4.4 CFS 运行队列……180
8.4.5 虚拟运行时间……181
8.4.6 CFS 调度类……182
8.4.7 schedule()……183
8.4.8 组调度……183
8.5 进程抢占与切换……185
8.5.1 用户抢占……185
8.5.2 内核抢占……186
8.5.3 进程切换……187

第 9 章 内存管理……189

9.1 内存概述……190
9.1.1 地址空间……190
9.1.2 分页……191
9.2 内核的内存分配……192
9.2.1 内存结构……192
9.2.2 BUDDY 页面管理……195
9.2.3 SLAB 内存管理……198
9.2.4 SLUB/SLOB 内存管理……201
9.2.5 vmalloc 内存分配……203
9.2.6 高端内存映射……204
9.3 进程地址空间……204
9.3.1 内存描述符……205
9.3.2 内存映射……208
9.3.3 多级页表结构……209
9.3.4 缺页错误处理……210
9.4 页面缓存……211
9.4.1 页面缓存管理……212
9.4.2 Swap 内存交换……213
9.4.3 kswapd 和 pdflush……214
9.5 内存工具汇总……214
9.5.1 用 proc 接口查看内存信息……214
9.5.2 系统命令工具……215

第 10 章 文件系统……217

10.1 概述……218
10.1.1 文件……218
10.1.2 文件系统……218
10.1.3 虚拟文件系统……221
10.2 VFS 的数据结构……224
10.2.1 超级块……224
10.2.2 索引节点……227
10.2.3 目录项……232
10.2.4 文件……235

10.2.5 与文件系统相关的数据结构 ……238
10.2.6 与进程相关的数据结构 ……240
10.2.7 路径查找辅助结构 ……243
10.3 VFS 的缓存机制 ……244
10.3.1 索引节点缓存 ……244
10.3.2 目录项缓存 ……245
10.3.3 缓冲区缓存 ……247
10.4 文件系统的注册与安装 ……248
10.4.1 文件系统的注册 ……248
10.4.2 文件系统的安装 ……249
10.4.3 rootfs 的注册和安装 ……250
10.5 inotify 机制 ……251
10.5.1 inotify 数据结构 ……251
10.5.2 inotify 钩子函数 ……253
10.5.3 inotify 用户接口 ……254
10.5.4 应用示例 ……255
10.5.5 inotify 工具 ……256

第 11 章 设备驱动 ……258

11.1 概述 ……259
11.2 模块机制与“Hello World!” ……260
11.3 设备模型 ……262
11.3.1 设备模型的经济基础 ……263
11.3.2 设备模型的上层建筑 ……268
11.3.3 类（Class）与类设备（class_device） ……273
11.4 sysfs 文件系统 ……275
11.4.1 sysfs 与/sys ……276
11.4.2 sysfs 目录项 sysfs_dirent ……277
11.4.3 sysfs 目录和属性 ……278
11.5 spec、datasheet 与内核源代码 ……279
11.6 USB 子系统与 USB 驱动 ……279
11.6.1 USB 简史 ……279
11.6.2 USB 协议基础 ……281
11.6.3 USB 子系统与 sysfs ……284
11.6.4 内核中的 USB ……286
11.6.5 USB 设备基础 ……288
11.6.6 USB urb ……304
11.6.7 OTG 简介 ……310
11.6.8 USB 驱动程序 ……310

第 12 章 内核开发基础 ……316

12.1 相关资源 ……317
12.1.1 内核文档 ……317
12.1.2 经典书籍 ……318
12.1.3 内核社区 ……319
12.1.4 其他网络资源 ……319
12.2 编码风格 ……320
12.3 内核 API ……324
12.4 内核中的 Makefile ……325
12.5 如何添加自己的驱动程序 ……327
12.6 如何提交补丁 ……329
12.7 学会使用 Git ……332

第 13 章 调试 ……335

13.1 内核调试配置选项 ……336
13.2 二分法与 printk() ……337
13.2.1 二分查找法的基本原理 ……337
13.2.2 printk() ……338
13.3 获取内核信息 ……339
13.3.1 syslog 和 dmesg ……339
13.3.2 /proc ……340
13.3.3 /sys ……341
13.3.4 ioctl ……342
13.4 oops ……342
13.5 调试工具 ……343
13.5.1 gdb ……343
13.5.2 kgdb ……344
13.5.3 kdb ……346
13.5.4 kprobes ……346
13.5.5 systemtap ……347
13.5.6 kdump ……348
13.5.7 硬件工具 ……348
13.6 “神奇”的 SysRq ……349
13.7 使用模拟器与虚拟机 ……349