

TURING

图灵程序设计丛书

Linux/Unix系列

PRENTICE
HALL

“这是我读过的最全面的Linux设备驱动程序著作。”

—— Alan Cox, Linux内核维护者

Essential Linux Device Drivers

精通Linux 驱动程序开发 (英文版)

[印] Sreekrishnan Venkateswaran 著



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

Linux/Unix系列

Essential Linux Device Drivers
精通Linux
驱动程序开发
(英文版)

[印] Sreekrishnan Venkateswaran

江苏工业学院图书馆

藏书章

人民邮电出版社
北京

人民邮电出版社

样书

专用章

图书在版编目 (CIP) 数据

精通 Linux 驱动程序开发—Essential Linux Device
Drivers: 英文 / (印) 温卡特斯瓦兰 (Venkateswaran,
S.) 著. —北京: 人民邮电出版社, 2009.6
(图灵程序设计丛书)
ISBN 978-7-115-20647-3

I. 精… II. 温… III. Linux 操作系统—程序设计—英
文 IV. TP316.89

中国版本图书馆CIP数据核字 (2009) 第048918号

内 容 提 要

本书是 Linux 设备驱动程序领域的权威著作。全书基于 2.6 内核, 不仅透彻讲解了基本概念和技术, 更深入探讨了其他书没有涵盖或辄止浅尝的许多重要主题和关键难点, 如 PCMCIA、I²C 和 USB 等外部总线、视频、音频、无线连网和闪存等技术。在解释每一个技术时, 均讲解了相关的内核源码文件, 并给出了完整的开发实例。

本书适合中高级 Linux 开发人员阅读。

图灵程序设计丛书

精通Linux驱动程序开发 (英文版)

- ◆ 著 [印] Sreekrishnan Venkateswaran
责任编辑 傅志红
- ◆ 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>
北京顺义振华印刷厂印刷
- ◆ 开本: 800×1000 1/16
印张: 45.75
字数: 879千字
印数: 1—3 000册
- 2009年6月第1版
2009年6月北京第1次印刷
- 著作权合同登记号 图字: 01-2009-1813号

ISBN 978-7-115-20647-3/TP

定价: 99.00元

读者服务热线: (010)51095186 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

Authorized translation from the English language edition, entitled *Essential Linux Device Drivers*, 0131679848 by Sreekrishnan Venkateswaran, published by Pearson Education, Inc, publishing as Prentice Hall, Copyright © 2008 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by Pearson Education Asia Ltd., and Posts & Telecom Press Copyright ©2009.

本书中文简体字版由 Pearson Education Asia Ltd.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

仅限于中华人民共和国境内（香港、澳门特别行政区和台湾地区除外）销售发行。

本书封面贴有 Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

版权所有，侵权必究。

本书献给千千万万印度盲人。本书作者的所有收入将捐给他们。

序

拿到这本书，你也许会问自己：为什么还要另一本Linux设备驱动程序的书呢？这样的书不是已经有一堆了吗？

答案是：相对于其他图书，本书是一个巨大的飞跃。

首先，本书与时俱进，基于最新的2.6内核进行讲解。其次，也是更重要的，本书对驱动程序的讲解非常透彻。大多数设备驱动程序的图书仅仅讲解标准Unix内核图书或操作系统图书中论述的主题，譬如串口、磁盘驱动和文件系统等，如果你运气好，它可能也会讲解网络协议栈。

本书前进了一大步，它没有避重就轻，而是知难而上，探讨了在现代PC和嵌入式系统中必须面对的难点，比如PCMCIA、USB、I²C、视频、音频、闪存、无线通信等。你可以这样定位这本书：Linux内核包含了什么，这本书就会告诉你什么。

它毫无遗漏，应有尽有，不留死角。

进而，本书的作者赢得了他的荣耀：读到他在20世纪90年代末期就将Linux移植到某种智能手表上的故事就像在进行一次惊险的奇幻之旅。

本书能成为Prentice Hall出版社开源软件开发系列图书中的一本，我感到非常兴奋。在开源领域发生这样的事情，令人振奋，它将成为开源领域的光辉典范。我希望你能从本书中找到你在进行内核开发时需要的东西，并且也能享受这一过程。

Arnold Robbins

（资深 Linux 技术专家，Gawk 维护者）

前言

20世纪90年代末，我们IBM的一群同事进行了一项将Linux内核移植到一种智能手表上的工作。目标设备虽然微不足道，但是移植Linux的任务却相当艰巨。在当时，内核中还不存在MTD（Memory Technology Device，内存技术设备）子系统，这意味着在文件系统能够运行在这种手表的闪存之前，我们不得不从头开发必要的存储驱动器。由于当时内核的输入事件驱动接口尚未诞生，因此手表的触摸屏与用户应用程序的接口非常复杂。让X Windows运行在手表的LCD上十分困难，因为X Windows和帧缓冲设备驱动程序搭配得并不好。如果你戴着一块防水的Linux智能手表，却不能躺在浴缸里实时获得股票行情，那么这块手表还有什么用？Linux几年前就已集成了蓝牙技术，而当时我们却花费了数月的时间将一种专有的蓝牙协议栈移植到手表上，从而使得这种手表具备了Internet连网能力。电源管理系统可以让这种手表的电池多运行几个小时，因此我们又设计了一个省电方案。那时候，Linux红外项目Linux-Infrared还不稳定，我们十分小心地用红外协议栈，设计出使用红外键盘作为手表的数据输入设备。最后，由于当时还没有能应用于消费类电子产品的成型的编译器发行版，我们也不得不编译出编译器，并交叉编译出一个紧凑的应用程序集。

时光飞逝，当年嗷嗷待哺的企鹅宝宝已经成长为一名健壮的少年。过去我们编写了成千上万行代码并耗时一年完成的任务，若采用现在的内核，只需要几天的时间就可以完成。但是，要成为一名能巧妙地解决多种问题的高级内核工程师，你需要理解今天的Linux内核提供的各种功能和设施。

关于本书

在Linux内核源代码树提供的各个子系统中，*drivers/*目录是其中最大的一个分支，它比其他子系统大数倍。随着各种新技术的广泛应用，内核中新的设备驱动程序的开发工作正在稳步加速。最新的Linux内核支持多达70余种设备驱动程序的庞大家族。

本书主要讲解Linux设备驱动程序，介绍了目前内核所支持的主要设备类型的设计与开发，其中包括当年我在开发Linux-on-Watch项目时未遇到的设备。本书在讲解每种设备驱动程序家族的时候，先介绍与该驱动程序相关的技术，接着给出一个实际的开发例子，最后列出相关的内核源代码文件。在踏入Linux设备驱动程序领域之前，本书先介绍了内核以及Linux 2.6的重要特性，重点讲解了设备驱动程序编写者感兴趣的内核知识。

读者对象

本书面向渴望在Linux内核上开发新设备驱动程序的中级程序员。要阅读本书，你需要具备操作系统相关的基本概念。比如，要知道什么是系统调用，理解为什么在内核开发中需要关注并发问题。本书假定你已经下载了Linux，浏览过Linux内核源代码，并至少浏览过一些相关的文档。另外，你必须能非常熟练地使用C语言。

各章概述

前4章为你阅读本书剩余部分打下了基础，接下来的16章讨论了不同的Linux设备驱动程序家族，之后的第21章描述了设备驱动程序的调试技术，第22章讲解了维护和交付设备驱动程序的相关事宜，最后一章给出了当你接到一个新设备驱动程序开发任务的时候，要首先查验的项目清单。

第1章带你走入Linux的世界。这一章教你下载内核源代码、进行小的代码修改并建立可启动的Linux内核映像。

第2章引导读者轻松地进入Linux内核的内部结构，讲解一些必要的内核概念。这一章首先讲述了内核的启动进程，接下来描述了与驱动程序开发相关的内核API，譬如内核定时器、并发管理以及内存分配等。

第3章讲解了对驱动程序开发有用的一系列内核API。这一章首先介绍了内核线程（它提供了一种在内核空间运行后台任务的能力），接下来讲解了一系列的辅助API（如链表、工作队列、完成函数、通知链等）。这些辅助API能简化代码、剔除内核中的冗余并有助于内核的长期维护。

第4章为你打下驾驭Linux设备驱动程序开发艺术的基础。这一章通过呈现一般的PC兼容系统和嵌入式设备的体系结构的鸟瞰图介绍了设备和驱动程序，并讲解了中断处理和内核设备模型等基本的驱动程序概念。

第5章介绍了Linux字符设备驱动程序的体系结构。这一章引入了几个新概念，譬如轮询、异步通知和I/O控制等。由于本书后面介绍的大多数设备都可以看作“超级”字符设备，所以这些概念也与后续章节密切相关。

第6章讲解了内核串口设备驱动程序的层次结构。

第7章讨论了内核中为键盘、鼠标和触摸屏控制器等输入设备服务的输入子系统。

第8章讲解了通过I²C总线或SMBus总线与系统连接的设备（如EEPROM）的驱动程序。同时，这一章也介绍了SPI总线和1-wire总线等其他串行接口。

第9章分析了PCMCIA子系统。这一章将教会你如何编写含PCMCIA或Compact Flash组件的设备的驱动程序。

第10章描述了内核对PCI及其衍生总线设备的支持。

第11章探讨了USB的体系结构，并讲解了如何利用Linux内核USB子系统的API来开发USB设备驱动程序。

第12章讲解了Linux视频子系统。这一章分析了内核提供的帧缓冲结构的优点，并给出了帧缓冲设备驱动程序的编写方法。

第13章描述了Linux音频子系统的架构，并给出了音频设备驱动程序的实现方法。

第14章集中描述存储设备（如硬盘）的驱动程序。在这一章里，你也将学习到Linux块子系统所支持的几种不同的I/O调度策略。

第15章分析了网络设备驱动程序，你将学习内核中与网络相关的数据结构以及网络设备驱动程序与协议层接口的实现方法。

第16章描述了各种无线网络设备的驱动程序，如蓝牙、红外、无线局域网WiFi和蜂窝通信等。

第17章讲解了如何让闪存在嵌入式设备上运行起来，这一章最后讲解了PC上的FWH（Firmware Hub，固件中心）的驱动程序。

第18章步入嵌入式Linux的世界。这一章介绍嵌入式设备中的引导加载程序（bootloader）、内核以及设备驱动程序等主要的固件组成。由于Linux在嵌入式领域越来越受欢迎，你很有可能将从本书中学到的Linux驱动程序开发技能应用于嵌入式领域。

第19章讲解了如何在用户空间驱动各种设备。一些设备驱动程序（尤其是那些重策略、轻性能的设备）更适合在用户空间被驱动。这一章也分析了Linux进程调度对用户空间设备驱动程序响应时间的影响。

第20章描述了之前尚未论及的设备驱动程序系统，如错误侦测和校验（EDAC）、火线接口以及ACPI等。

第21章讲解了用来调试Linux内核代码的各种调试工具。在这一章中，你也将学习到跟踪（trace）工具、内核探测点（probe）、崩溃转储（crash-dump）和剖析器（profiler）的使用方法。当你在开发Linux驱动程序的时候，请用本章学到的驱动调试技能武装自己。

第22章给出了设备驱动程序软件开发生命周期的概况。

第23章给出了当你开始进行一个新设备驱动程序开发工作时，应该查验的工作项目清单。本书最后是对“下一步怎么做”的思考。

设备驱动程序中有时需要以汇编语言实现一些代码片段，因此，附录A介绍了Linux汇编编程的不同方面。x86系统上的一些设备驱动程序直接或间接地依赖于BIOS，因此，附录B讲解了Linux如何与BIOS交互。附录C描述了2.6内核提供的seq文件——用于监控和追踪数据点的辅助接口。

本书总体上根据设备和总线的复杂度进行组织，同时也结合了章与章之间互相依赖的客观情况。我们从讲解基本的设备类型（如字符设备、串口和输入设备）开始，紧接着介绍简单的串行总线（如I²C和SMBus），之后介绍了PCMCIA、PCI和USB等外部I/O总线。由于视频、音频、块和网络设备通常通过这些I/O总线与处理器连接，因此在介绍完这些总线之后，

介绍了这些设备的驱动程序。书中接下来的部分面向嵌入式Linux，讲述了无线连网和闪存等技术。最后讨论了用户空间的设备驱动程序。

内核版本

本书总体上紧跟2.6.23/2.6.24内核版本，书中列出的大部分代码都在2.6.23上进行过测试。如果你正在使用更新的版本，请通过类似lwn.net的Linux网站了解内核自2.6.23/2.6.24后进行了哪些更改。

本书网站

我特意建立了elinuxdd.com这个网站来提供与本书相关的更新、勘误等信息。

本书约定

源代码、函数名和shell命令使用等宽字体。shell提示符为**bash>**。文件名为斜体。另外，新名词也使用斜体表示。

为了实现代码示例，一些章节对原始的内核源代码进行了修改。为便于指出这些修改，新添加的代码前添加了“+”，删除的代码前则添加了“-”。

有时候，为了简化，本书使用了通用的路径名。因此，当你遇到`arch/your-arch/`目录时，应该根据你当前的编译情况进行转换。例如，如果你正在为x86体系结构编译内核，它应该转换为`arch/x86/`。类似地，如果你正在为ARM体系结构编译内核，`include/asm- your-arch/`就应该转换为`include/asm-arm/`。本书偶尔在文件名中使用*和X作为通配符。因此，如果书中要求查看`include/linux/time*.h`文件，你就应该查看`include/linux/`下的`time.h`、`timer.h`、`times.h`和`timex.h`所有这些头文件。同样地，如果书中包含类似`/dev/input/eventX`或`/sys/devices/platform/i8042/serioX`这样的文件名，其中的X指在当前系统配置情况下内核分配给你的设备的接口号。

→符号有时候会插入在命令或内核的输出之间，以便附加更多的解释。

为了紧凑地列出函数原型，本书偶尔使用了一些简单的正则表达式。例如，第10章的Direct Memory Access一节，就用`pci_[map|unmap|dma_sync]_single()`指代了`pci_map_single()`、`pci_unmap_single()`和`pci_dma_sync_single()`。

有几章提到了用户空间的配置文件。例如，描述启动进程的一节打开了`/etc/rc.sysinit`，讨论蓝牙的一章引用了`/etc/bluetooth/pin`。这些文件的确切名称和位置都有可能由于你使用的Linux发行版的不同而不同。

致谢

首先感谢Prentice Hall出版社负责本书的编辑们：Debra Williams Cauley、Anne Goebel和Keith Cline。没有他们的支持，本书就不可能完成。感谢Mark Taub对这个项目的兴趣，是他发起了它。

过去的10年里，很多人和事对我的研究给予了巨大帮助，包括：在各个Linux项目里一起共事的同事们、强健的内核源代码、邮件列表和Internet。这些都对我完成本书发挥了重大作用。

*Linux Magazine*的Martin Streicher邀请我参与该杂志的“超级发烧友”内核栏目，从而把我从一名全职程序员变成了一名兼职写手。在技术写作过程中，我从他身上学习到了许多，在此表示感谢！

我要特别感谢我的技术审稿人。Vamsi Krishna耐心地读完了初稿的每一章，他提出了很多建设性的建议，令本书更进一步。Jim Lieb对书中的几章提供了有价值的反馈。Arnold Robbins浏览了开始的几章并且提供了富有见地的意见。

最后，我要感谢我的父母和妻子，感谢他们的爱与支持。我还要感谢我的宝宝，我的乖女儿，正是她不断地提醒我把时间投入到这本书上来，她摇摇晃晃地走来走去，那离奇的步伐极似企鹅。

Contents

Chapter 1 Introduction.....	1	Creating a Kernel Thread.....	56
Evolution.....	2	Process States and Wait Queues.....	61
The GNU Copyleft.....	3	User Mode Helpers	63
Kernel.org	4	Helper Interfaces.....	65
Mailing Lists and Forums	4	Linked Lists	65
Linux Distributions	5	Hash Lists.....	72
Looking at the Sources.....	6	Work Queues.....	72
Building the Kernel.....	10	Notifier Chains	74
Loadable Modules.....	12	Completion Interface	78
Before Starting	14	Kthread Helpers	81
		Error-Handling Aids.....	83
		Looking at the Sources.....	85
Chapter 2 A Peek Inside the Kernel.....	17	Chapter 4 Laying the Groundwork	89
Booting Up.....	18	Introducing Devices and Drivers	90
Kernel Mode and User Mode.....	30	Interrupt Handling.....	92
Process Context and Interrupt Context	30	Interrupt Context.....	92
Kernel Timers.....	31	Assigning IRQs.....	94
HZ and Jiffies.....	31	Device Example: Roller Wheel.....	94
Long Delays	33	Softirqs and Tasklets	99
Short Delays.....	36	The Linux Device Model	103
Pentium Time Stamp Counter.....	36	Udev	103
Real Time Clock	37	Sysfs, Kobjects, and Device Classes.....	106
Concurrency in the Kernel.....	39	Hotplug and Coldplug.....	110
Spinlocks and Mutexes	39	Microcode Download.....	111
Atomic Operators.....	45	Module Autoload	112
Reader-Writer Locks.....	46	Memory Barriers.....	114
Debugging.....	48	Power Management	114
Process Filesystem.....	49	Looking at the Sources.....	115
Allocating Memory	49		
Looking at the Sources.....	52	Chapter 5 Character Drivers.....	119
Chapter 3 Kernel Facilities	55	Char Driver Basics	120
Kernel Threads.....	56	Device Example: System CMOS.....	121

Driver Initialization.....	122
Open and Release.....	127
Exchanging Data.....	129
Seek.....	136
Control.....	137
Sensing Data Availability.....	139
Poll.....	139
Fasync.....	142
Talking to the Parallel Port.....	145
Device Example: Parallel Port LED Board.....	146
RTC Subsystem.....	156
Pseudo Char Drivers.....	157
Misc Drivers.....	160
Device Example: Watchdog Timer.....	160
Character Caveats.....	166
Looking at the Sources.....	167
Chapter 6 Serial Drivers.....	171
Layered Architecture.....	173
UART Drivers.....	176
Device Example: Cell Phone.....	178
RS-485.....	191
TTY Drivers.....	192
Line Disciplines.....	194
Device Example: Touch Controller.....	195
Looking at the Sources.....	205
Chapter 7 Input Drivers.....	207
Input Event Drivers.....	210
The Evdev Interface.....	210
Input Device Drivers.....	216
Serio.....	217
Keyboards.....	217
Mice.....	220
Touch Controllers.....	227
Accelerometers.....	228
Output Events.....	228
Debugging.....	230
Looking at the Sources.....	231

Chapter 8 The Inter-Integrated Circuit Protocol.....	233
What's I ² C/SMBus?.....	234
I ² C Core.....	235
Bus Transactions.....	237
Device Example: EEPROM.....	238
Initializing.....	238
Probing the Device.....	241
Checking Adapter Capabilities.....	244
Accessing the Device.....	244
More Methods.....	246
Device Example: Real Time Clock.....	247
I2C-dev.....	251
Hardware Monitoring Using LM-Sensors.....	251
The Serial Peripheral Interface Bus.....	251
The 1-Wire Bus.....	254
Debugging.....	254
Looking at the Sources.....	255
Chapter 9 PCMCIA and Compact Flash.....	257
What's PCMCIA/CF?.....	258
Linux-PCMCIA Subsystem.....	260
Host Controller Drivers.....	262
PCMCIA Core.....	263
Driver Services.....	263
Client Drivers.....	264
Data Structures.....	264
Device Example: PCMCIA Card.....	267
Tying the Pieces Together.....	271
PCMCIA Storage.....	272
Serial PCMCIA.....	272
Debugging.....	273
Looking at the Sources.....	275
Chapter 10 Peripheral Component Interconnect.....	277
The PCI Family.....	278
Addressing and Identification.....	281
Accessing PCI Regions.....	285

Configuration Space.....	285	Frame Buffer Drivers	365
I/O and Memory.....	286	Device Example: Navigation System	365
Direct Memory Access.....	288	Console Drivers	380
Device Example: Ethernet-Modem Card.....	292	Device Example: Cell Phone Revisited	382
Initializing and Probing.....	293	Boot Logo	387
Data Transfer.....	301	Debugging.....	387
Debugging.....	308	Looking at the Sources.....	388
Looking at the Sources.....	308		
Chapter 11 Universal Serial Bus....	311	Chapter 13 Audio Drivers.....	391
USB Architecture	312	Audio Architecture.....	392
Bus Speeds	314	Linux-Sound Subsystem	394
Host Controllers	315	Device Example: MP3 Player.....	396
Transfer Types.....	315	Driver Methods and Structures	399
Addressing	316	ALSA Programming	409
Linux-USB Subsystem.....	317	Debugging.....	412
Driver Data Structures	317	Looking at the Sources.....	412
The usb_device Structure.....	318	Chapter 14 Block Drivers	415
USB Request Blocks.....	319	Storage Technologies	416
Pipes.....	321	Linux Block I/O Layer.....	421
Descriptor Structures	322	I/O Schedulers.....	422
Enumeration.....	324	Block Driver Data Structures and	
Device Example: Telemetry Card.....	324	Methods.....	423
Initializing and Probing.....	325	Device Example: Simple Storage	
Accessing Registers	332	Controller	426
Data Transfer.....	335	Initialization	427
Class Drivers.....	338	Block Device Operations	430
Mass Storage.....	339	Disk Access.....	432
USB-Serial.....	345	Advanced Topics.....	434
Human Interface Devices.....	348	Debugging.....	436
Bluetooth.....	348	Looking at the Sources.....	437
Gadget Drivers	348		
Debugging.....	349	Chapter 15 Network Interface	
Looking at the Sources.....	351	Cards.....	439
Chapter 12 Video Drivers	355	Driver Data Structures	440
Display Architecture	356	Socket Buffers.....	441
Linux-Video Subsystem.....	359	The Net Device Interface.....	443
Display Parameters	361	Activation.....	444
The Frame Buffer API.....	362	Data Transfer.....	444
		Watchdog	445

Statistics	445	Looking at the Sources.....	496
Configuration	446	Cellular Networking.....	496
Bus Specific	448	GPRS.....	496
Talking with Protocol Layers.....	448	CDMA.....	498
Receive Path.....	448	Current Trends	500
Transmit Path.....	449		
Flow Control	449	Chapter 17 Memory Technology	
Buffer Management and Concurrency		Devices.....	503
Control	450	What's Flash Memory?.....	504
Device Example: Ethernet NIC	451	Linux-MTD Subsystem	505
ISA Network Drivers	457	Map Drivers	506
Asynchronous Transfer Mode.....	458	Device Example: Handheld	506
Network Throughput.....	459	NOR Chip Drivers	511
Driver Performance.....	459	NAND Chip Drivers	513
Protocol Performance.....	461	User Modules	516
Looking at the Sources.....	461	Block Device Emulation	516
		Char Device Emulation.....	517
Chapter 16 Linux Without Wires....	465	JFFS2	517
Bluetooth.....	467	YAFFS2.....	518
BlueZ.....	469	MTD-Utils.....	518
Device Example: CF Card	471	Configuring MTD	519
Device Example: USB Adapter	471	eXecute In Place	520
RFCOMM.....	473	The Firmware Hub.....	520
Networking	475	Debugging.....	524
Human Interface Devices.....	477	Looking at the Sources.....	524
Audio.....	477		
Debugging.....	478	Chapter 18 Embedding Linux.....	527
Looking at the Sources.....	478	Challenges.....	528
Infrared.....	478	Component Selection.....	530
Linux-IrDA	480	Tool Chains	531
Device Example: Super I/O Chip	482	Embedded Bootloaders	531
Device Example: IR Dongle	483	Memory Layout	535
IrComm	486	Kernel Porting.....	537
Networking	486	Embedded Drivers	538
IrDA Sockets.....	487	Flash Memory	538
Linux Infrared Remote Control	488	UART	539
Looking at the Sources.....	489	Buttons and Wheels	539
WiFi	489	PCMCIA/CF	540
Configuration	490	SD/MMC.....	540
Device Drivers	494	USB.....	540

RTC	541	FireWire	588
Audio	541	Intelligent Input/Output	589
Touch Screen	541	Amateur Radio	590
Video	541	Voice over IP	590
CPLD/FPGA	542	High-Speed Interconnects	591
Connectivity	542	InfiniBand	592
Domain-Specific Electronics	542	RapidIO	592
More Drivers	543	Fibre Channel	592
The Root Filesystem	544	iSCSI	593
NFS-Mounted Root	544		
Compact Middleware	546	Chapter 21 Debugging Device	
Test Infrastructure	548	Drivers	595
Debugging	548	Kernel Debuggers	596
Board Rework	549	Entering a Debugger	597
Debuggers	550	Kernel Debugger (kdb)	598
		Kernel GNU Debugger (kgdb)	600
Chapter 19 Drivers in User		GNU Debugger (gdb)	604
Space	551	JTAG Debuggers	605
Process Scheduling and Response Times	553	Downloads	609
The Original Scheduler	553	Kernel Probes	609
The O(1) Scheduler	553	Kprobes	609
The CFS Scheduler	555	Jprobes	614
Response Times	555	Return Probes	617
Accessing I/O Regions	558	Limitations	619
Accessing Memory Regions	562	Looking at the Sources	620
User Mode SCSI	565	Kexec and Kdump	620
User Mode USB	567	Kexec	620
User Mode I ² C	571	Kexec with Kdump	621
UIO	573	Kdump	622
Looking at the Sources	574	Looking at the Sources	629
		Profiling	629
Chapter 20 More Devices and		Kernel Profiling with OProfile	629
Drivers	577	Application Profiling with Gprof	633
ECC Reporting	578	Tracing	634
Device Example: ECC-Aware Memory		Linux Trace Toolkit	634
Controller	579	Linux Test Project	638
Frequency Scaling	583	User Mode Linux	638
Embedded Controllers	584	Diagnostic Tools	638
ACPI	585	Kernel Hacking Config Options	639
ISA and MCA	587	Test Equipment	640

Chapter 22 Maintenance and Delivery 641

Coding Style.....	642
Change Markers.....	642
Version Control.....	643
Consistent Checksums.....	643
Build Scripts.....	645
Portable Code.....	647

Chapter 23 Shutting Down 649

Checklist.....	650
What Next?.....	651

Appendix A Linux Assembly 653

Debugging.....	659
----------------	-----

Appendix B Linux and the BIOS.... 661

Real Mode Calls.....	662
Protected Mode Calls.....	665
BIOS and Legacy Drivers.....	666

Appendix C Seq Files..... 669

The Seq File Advantage.....	670
Updating the NVRAM Driver.....	677
Looking at the Sources.....	679

Index.....	681
------------	-----