

TURING

图灵程序设计丛书

PEARSON

Embedded Linux Primer

A Practical Real-World Approach

Second Edition

嵌入式Linux 基础教程 (第2版)

[美] Christopher Hallinan 著
周鹏 译



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

Embedded Linux Primer

A Practical Real-World Approach **Second Edition**

嵌入式Linux 基础教程 (第2版)

[美] Christopher Hallinan 著
周鹏 译



人民邮电出版社
北京

图书在版编目 (C I P) 数据

嵌入式Linux基础教程：第2版 / 哈利南
(Hallinan, C.) 著；周鹏译. — 北京：人民邮电出版
社, 2012. 5

(图灵程序设计丛书)

书名原文: Embedded Linux Primer: A Practical
Real-World Approach, Second Edition
ISBN 978-7-115-27827-2

I. ①嵌… II. ①哈… ②周… III. ①
Linux操作系统—程序设计 IV. ①TP316.89

中国版本图书馆CIP数据核字(2012)第056904号

内 容 提 要

本书是嵌入式 Linux 的经典教程, 介绍了引导加载程序、系统初始化、文件系统、闪存和内核、应用程序调试技巧等, 还讲述了构建 Linux 系统的工作原理, 用于驱动不同架构的配置, Linux 内核源码树的特性, 如何根据需求配制内核运行时的行为, 如何扩展系统功能, 用于构建完整嵌入式 Linux 发行版的常用构建系统, USB 子系统和系统配置工具 udev 等内容。更重要的是, 本书阐述了如何修改系统使之满足读者自身的需求, 确保读者能够从中学习一些嵌入式工程中非常有用的提示和技巧。

本书适合 Linux 程序员阅读, 也可作为高等院校相关专业师生的参考读物。

图灵程序设计丛书

嵌入式Linux基础教程 (第2版)

◆ 著 [美] Christopher Hallinan

译 周 鹏

责任编辑 朱 巍

执行编辑 刘美英

◆ 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 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京鑫正大印刷有限公司印刷

◆ 开本: 800×1000 1/16

印张: 29.5

字数: 697千字

2012年5月第1版

印数: 1-4 000册

2012年5月北京第1次印刷

著作权合同登记号 图字: 01-2011-1848号

ISBN 978-7-115-27827-2

定价: 89.00元

读者服务热线: (010)51095186转604 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 *Embedded Linux Primer: A Practical Real-World Approach, Second Edition*, 978-0-13-701783-6 by Christopher Hallinan, published by Pearson Education, Inc., publishing as Prentice Hall, Copyright ©2011 by 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 © 2012.

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。

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。

第2版序

智能手机、PDA、家用路由器、智能电视、智能蓝光播放器以及智能悠悠球……好吧，或许智能悠悠球算不上。不管是用于工作也好，还是消遣也罢，越来越多的家用和办公用品中嵌入了计算机，这些计算机都运行着GNU/Linux。

你也许是一位习惯了在英特尔架构的台式电脑（或笔记本）上工作的GNU/Linux开发人员，或者你有可能是一位嵌入式系统开发人员，习惯于那些更传统的嵌入式或实时操作系统，无论你的背景如何，一旦进入嵌入式Linux开发的世界，多萝西^①所说的“托托，我想我们再也回不去堪萨斯了”就会在你身上应验。欢迎来参加这个探险之旅！

多萝西有目标，有一些好朋友，但是没有向导。然而，你的情况要好一些，因为你正拿着一本极好的“野外生存指南”，它能带领你进入嵌入式Linux开发的世界。Christopher Hallinan为你展示了这一切——怎么做，从何处着手，为什么要这样做以及哪些是不用做的。本书将让你免受磨练之苦，带领你轻松而快捷地打造自己的产品。

不必惊讶，本书在这个领域是首屈一指的。第2版更上一层楼，包含了最新的内容，融会了作者更多的经验。

能将本书纳入我们的“开源软件开发系列”，我深感荣幸。但更重要的是，你会为自己感到自豪，因为读了这本书，你会开发出更好的产品。学习愉快！

Arnold Robbins
(著名Linux专家)

^① 多萝西 (Dorothy) 是童话故事《绿野仙踪》的主人公。——译者注

第 1 版 序

计算机无处不在！

在过去大约25年中，只要不是与世隔绝的人就肯定不会对此感到惊讶。现在，计算机不仅占据了我们的桌面，进驻了我们的厨房，而且越来越多地进入到我们的生活，即便是在微波炉、电烤箱、移动电话和便携式数字音乐播放器中也出现了它的身影。

选择本书的读者肯定已经了解了不少，但还想学习更多的嵌入式系统知识。

就在不久前，嵌入式系统还不是很强大，它们运行具有特殊目的、专用的操作系统，而这些操作系统与工业标准的系统有很大不同（而且，它们也更难开发）。现在，嵌入式系统即使在功能上不比家用计算机强大，但至少也与其相当（如高端游戏机）。

伴随着这种强大的功能，运行Linux等成熟操作系统的能力也呼之欲出，在嵌入式产品中使用Linux这样的操作系统意义重大。一个庞大的开发者社区更使这一切成为可能。开发环境和部署环境惊人相似，这也让程序员的生活变得更轻松。现在我们既有由虚拟内存系统提供的保护地址空间的安全性，又有多用户、多进程操作系统的能力和灵活性。这真可谓应有尽有了。

出于这个原因，世界上很多公司都选择把Linux放在自己的设备中，如PDA、家庭娱乐系统，甚至——不管你信不信——手机里！

这本书很令我振奋。它为那些想在嵌入式系统中使用Linux的开发人员提供了极好的学习路线指导。本书内容简洁、准确，组织合理，字里行间渗透出Christopher的学问和见解，你不仅能从中得到很多信息和帮助，也能获得阅读的乐趣。

我希望你在学习的同时也能感受到这种乐趣，我自己已经感受到了。

Arnold Robbins
(著名Linux专家)

前言

虽然讲Linux的好书很多，但是本书汇集了专门针对嵌入式Linux开发人员的各方面信息和建议。实际上，有关Linux内核、Linux系统管理等方面的优秀书籍已经有很多了。本书也参考了我认为在同类书籍中最好的几本。

本书的很多内容来自实际的问题，有些问题是我作为嵌入式Linux顾问时一些开发工程师提出的，还有些问题是我直接参与商业嵌入式Linux开发时遇到的。

嵌入式Linux会给经验丰富的软件工程师带来一些独特的挑战。首先，那些有多年老式实时操作系统（Real Time Operating System, RTOS）开发经验的工程师会发现，他们的思维习惯很难从旧的环境转换到Linux；其次，经验丰富的应用程序开发人员常常难以理解多种开发环境的相对复杂性。

虽然这是一本面向嵌入式Linux开发初学者的基础教程，但我相信即使是经验丰富的嵌入式Linux开发人员也能从中获益，书中包含了我多年积累的实用建议和技巧。

给实际的嵌入式开发人员的实用建议

本书介绍了嵌入式工程师怎样才能迅速掌握嵌入式Linux环境的新知识。书中没有重点讲解Linux内核原理，而是在讲解内核的章节侧重从项目角度介绍内核。你可以阅读专门介绍内核原理的优秀图书来了解相关知识。你可以从本书学到内核源码树的组织和布局，了解组成内核镜像的二进制组件和加载它们的方法，以及它们在嵌入式系统中的作用。

在本书中，你会学到Linux内核构建系统的工作原理，以及怎样将满足项目需求的具体变化融合到系统中。你会了解到Linux系统初始化的细节，包括内核空间初始化和用户空间初始化。你还能看到很多对嵌入式项目有益的建议和技巧，涵盖引导加载程序、系统初始化、文件系统和闪存，以及高级的内核与应用程序调试技术。第2版中新增了很多内容，很多章节都有更新，比如讲开源构建系统、USB和udev的几章都是新的，而且有相当篇幅探讨如何在嵌入式Linux项目中配置和使用这些复杂的系统。

目标读者

本书需要读者具有一定的C语言编程基础，对局域网和因特网有基本的了解，理解IP地址的

概念以及IP地址在简单局域网中的用法，还需要理解十六进制和八进制编码方式以及它们常见的用法。

本书也涉及一些C语言编译和链接中较为深入的概念，因此你要是能粗略复习一下C语言链接器的概念就更好了。同时，了解GNU make操作和语法对于阅读本书也很有帮助。

本书不是什么

本书不是一本详细介绍硬件的指南。硬件设备种类繁多是嵌入式开发者所面临的一大困难。集成了一些外围设备的现代32位处理器，其用户手册动辄就有3000页。没有捷径可走。作为程序员，要想理解硬件设备就必须花费大量时间研读硬件数据手册和参考指南，同时要花费更多的时间针对这些硬件设备编写和测试代码。

这也不是一本讲述Linux内核或内部原理的书。本书不会深入讨论用来实现Linux虚拟内存管理策略和过程的内存管理单元（MMU）。已经有许多关于这个主题的优秀图书了，建议你读一读每章后面的“补充阅读建议”。

排版约定

命令和代码语句使用Courier New字体。用户输入的命令使用加粗的**Courier New**字体。新名词和重要概念使用楷体加以强调。

路径名前面有3个点表示这是大家熟知但未明确指定的顶层目录。上下文不同，顶层目录也会不同，但一般都是指顶层的Linux源码目录。例如，`.../arch/powerpc/kernel/setup_32.c`指位于Linux源码树体系结构分支中的文件`setup_32.c`。实际的文件路径有可能是`~/sandbox/linux.2.6.33/arch/power/kernel/setup_32.c`。

本书结构

第1章简要介绍了促使Linux迅速应用于嵌入式环境的因素，同时也介绍了与嵌入式Linux相关的几个重要标准和组织。

第2章介绍了很多嵌入式Linux相关的概念，这些概念是后续几章的基础。

第3章概述了几种较流行的用于搭建嵌入式Linux系统的处理器和平台，介绍了几款主要处理器厂商的重要产品，涉及所有主要的处理器体系结构。

第4章从另一角度探讨了Linux内核。这里没有重点讲解内核理论或其内部原理，只是介绍了内核的结构、布局和构建结构，目的是使读者从一开始就能以自己的方式学习这个庞大的软件工程项目。更重要的是，要知道哪些内容是必须重点关注的。这一章还详细讲解了内核构建系统。

第5章详细说明了Linux内核的初始化过程：把与体系结构和引导加载程序相关的镜像组件拼接成适合下载到闪存的内核镜像，最终通过嵌入式系统的引导加载程序启动。这一章的知识将帮助你定制Linux内核，使之满足你自己的嵌入式应用的需求。

第6章继续详细介绍初始化过程。当Linux内核完成自身的初始化后，应用程序将根据预先确定的方式继续初始化过程。读完这一章，你就具备了定制用户空间应用程序启动顺序的知识。

第7章主要介绍引导加载程序及其在嵌入式Linux系统中的作用。这一章以现在流行的开源引导加载程序U-Boot为例说明了移植的概念；还简要介绍了其他几种现在使用的引导加载程序，以使用户有特殊需求时可以有多种选择。

第8章介绍了Linux设备驱动程序模型，提供了很多开发设备驱动程序的背景资料，这些资料都在每章结尾的“补充阅读建议”中列出了。

第9章列举了目前嵌入式系统中使用的一些流行的文件系统，包括闪存设备上最常用的JFFS2文件系统。这一章还简要介绍了如何创建你自己的文件系统镜像，这也是嵌入式Linux开发人员所面临的一项艰巨任务。

第10章介绍了MTD（Memory Technology Device，内存技术设备）子系统。MTD是Linux文件系统和硬件内存设备（尤其是闪存）之间一种非常有效的抽象层。

第11章介绍了BusyBox，它是构建小型嵌入式系统最常用的工具。这一章讲述如何根据特殊需求来配置和构建BusyBox，随后介绍了仅使用BusyBox环境完成系统初始化的全过程。附录A列举了最新版本BusyBox提供的命令。

第12章详细介绍了典型交叉开发环境的特殊需求。这一章所介绍的一些技术能有效地提高嵌入式开发人员的工作效率，例如强大的NFS根目录挂载开发配置。

第13章介绍了一些有用的开发工具。这一章介绍了使用gdb进行调试，包括核心转储分析；并通过示例介绍了strace、ltrace、top和ps，以及内存性能评测工具mtrace和dmalloc。这一章最后介绍了几个更重要的二进制实用工具，如强大的readelf程序。

第14章深入探讨了很多Linux内核的调试技术，介绍了内核调试器KGDB的用法，提出了许多gdb和KGDB组合使用的调试技巧。这一章涉及的内容还包括硬件JTAG调试器的用法，以及当内核无法启动时的一些故障分析技巧。

第15章把调试环境从内核转移至应用程序。这一章继续完善前两章用到的gdb示例，讲述了多线程和多进程的调试技巧。

第16章取代了第1版的第16章（移植Linux）。那一章的内容已经过时了，如果要在现代内核中恰当地讨论其主题，则需要专门写一本书。我觉得你会对新版的第16章有兴趣的，这一章涵盖了常用的构建完整嵌入式Linux发行版的构建系统。我们会介绍OpenEmbedded，它已经在商业和其他开源项目中获得了极大的关注。

第17章介绍了嵌入式Linux中一个令人激动的发展：通过PREEMPT_RT选项来配置系统的实时性。这里介绍的特性通过RT选项得以实现，同时还介绍了如何在设计中使用这些特性。这一章也介绍了衡量应用程序配置延时的技巧。

第18章以简单易懂的语言描述了USB子系统。我们介绍了一些概念和USB拓扑结构，接着给出几个USB配置的例子。我们会详细分析sysfs和USB的作用，以帮助你理解这个功能强大的系统。我们还会介绍几个有助于理解USB和解决USB故障的工具。

第19章解密了udev这个强大的系统配置工具。我们分析udev的默认行为，并以此为基础来理

解如何对它进行定制。我们会给出几个实际的例子。对于BusyBox的使用者，我们会考察BusyBox自带的mdev工具。

本书附录包含U-Boot配置命令、BusyBox命令、SDRAM接口注意事项、开源开发者的资源、BDI-2000调试器的配置文件范例。BDI-2000是目前很流行的硬件JTAG调试器。

边看边做

如果你能边看书边在你喜欢的Linux工作站上动手实验，将会从书中得到最大的收获。你可以找一台较旧的x86计算机完成嵌入式系统实验。如果有条件能连接其他体系结构的单板计算机进行实验就更好了。BeagleBorad开发板是一个可以进行实验的物美价廉的平台，书中的好几个例子都基于这个平台。通过学习这个大型代码库（Linux内核）的布局和组织结构，你将获益良多，并且能够在研究内核和边学边做的过程中获得大量的知识和经验。

看一下本书使用的代码并试着理解书中的示例，要使用不同的设置方案、配置选项和不同的硬件设备进行实验。除可获得丰富的知识，还充满了乐趣！如果你也这么想，请登录本书网站www.embeddedlinuxprimer.com免费注册一个账号，添加些内容和评论，在这个逐步壮大的Linux社区中分享你自己的成功故事和解决方案。你分享的内容会帮助其他人学习。这是一项不断完善的工作，你的参与会使其成为一个有价值的社区资源。

GPL 版权声明

本书使用的部分开源代码的版权归很多个人或公司所有。复制代码遵循了GPL，即GNU公共许可。

第 2 版致谢

首先，我必须感谢高级策划编辑Debra Williams Cauley，他指导有方，经验丰富，而且很有耐心。没有他，本书就不会出版。

非常感谢我的评审团队：Robert P.J. Day、Sandy Terrace、Kurt Lloyd、Jon Masters和丛书编辑Arnold Robbins。他们对本书质量提升所做的贡献，让我难以言表。

还要感谢Mark A. Yoder教授和他的嵌入式Linux课程，他在课堂上使用了本书的手稿，全面检验了这些内容。

特别感谢飞思卡尔半导体公司提供了硬件平台，书中的很多例子都基于这个平台。如果没有Kalpesh Gala帮忙安排，我就享受不到这样的支持。

还要感谢Embedded Planet和Tim Van de Walle，他们也提供了硬件平台供本书使用。

本书撰写过程中，有几个人给予了我很大帮助，他们提出建议并解答我的问题：Cedric Hommbourger、Klaas van Gend、George Davis、Sven-Thorsten Dietrich、Jason Wessels和Dave Anders（排名没有先后次序）。

我还想感谢这本书的制作团队，我的日程安排不够准确给他们添了麻烦。他们是Alexandra Maurer、Michael Thurston、Jovana San Nicolas Shirley、Gayle Johnson、Heather McNeill、Tricia Bronkella和Sarah Kearns。

出版本书是个很大的项目，有无数人提供了帮助，他们及时回答问题，跟我在交谈中提出想法。这些人太多了，无法一一列举。但他们对我有意或无意的帮助都是我真心感激的。

在本书第1版中，我特别感谢了Cary Dillman，因为每当写完一章，她就会不辞劳苦地审读。她现在是我心爱的妻子了。在第2版的写作过程中，Cary继续支持我，给了我必要的灵感、耐心，作出了不少牺牲。

第 1 版致谢

我由衷地敬佩开源软件工程师的崇高精神，深深地折服于我们社区中远远超过我的天才们。在本书的撰写过程中，我向Linux和开源社区的很多人提出了大量问题，大多数问题都能很快得到回答，而且还经常获得鼓励。我要向Linux和开源社区中帮我解答问题的朋友致以真挚的谢意（排名不分先后）：

Dan Malek为第2章的部分内容提供了创作灵感。

Dan Kegel和Daniel Jacobowitz耐心地帮我解答了关于工具链的问题。

Scott Anderson 提供了第14章中gdb宏的最初思想。

Brad Dixon不断用他所掌握的知识挑战和扩展我的技术洞察力。

George Davis帮我解答了ARM的问题。

Jim Lewis为我提供了关于MTD的意见和建议。

Cal Erickson帮我解答了关于gdb用法的问题。

John Twomey就第3章内容给出了建议。

Lee Revell、Sven-Thorsten Dietrich和Daniel Walker就实时Linux的内容提供了建议。

非常感谢AMCC、Embedded Planet、Ultimate Solutions和United Electronic Industries公司，它们提供了示例硬件。感谢我的公司Monta Vista Software，因为写这本书偶尔会从工作中分点儿心，而且公司还为某些示例提供了软件支持。在创作过程中，还有很多人贡献了他们的想法，并给予我鼓励和支持，我也非常感激！

我要诚挚地感谢最初审阅本书的团队，他们迅速阅读了每一章，提供了极好的反馈、意见和想法。谢谢Arnold Robbins、Sandy Terrace、Kurt Lloyd和Rob Farber。还要感谢Arnold教给我这个写作新手撰写技术图书的规则。虽然我已经努力减少错误，但错误肯定还会存在，这都得归咎于我。

感谢Mark L.Taub使本书得以完成，感谢他的鼓励和无限的耐心。还要感谢制作团队，包括Kristy Hart、Jennifer Cramer、Krista Hansing和Cheryl Lenser。

最后，把最特别、最衷心的感谢献给Cary Dillman，在我撰写本书时她阅读了每一章，整个创作过程中都有她的不断鼓励和重要的贡献。

目 录

第 1 章 入门	1	2.4.2 打造自己的 Linux 发行版	24
1.1 为什么选择 Linux	1	2.5 小结	24
1.2 嵌入式 Linux 现状	2	第 3 章 处理器基础	26
1.3 开源和 GPL	2	3.1 独立处理器	26
1.4 标准及相关组织	3	3.1.1 IBM 970FX	27
1.4.1 Linux 标准基础	3	3.1.2 英特尔奔腾 M	27
1.4.2 Linux 基金会	4	3.1.3 英特尔凌动™	28
1.4.3 电信级 Linux	4	3.1.4 飞思卡尔 MPC7448	28
1.4.4 移动 Linux 计划: Moblin	4	3.1.5 配套芯片组	28
1.4.5 服务可用性论坛	5	3.2 集成处理器: 片上系统	30
1.5 小结	5	3.2.1 Power 架构	30
第 2 章 综述	6	3.2.2 飞思卡尔 Power 架构	30
2.1 嵌入与非嵌入	6	3.2.3 飞思卡尔 PowerQUICC I	31
2.2 剖析嵌入式系统	7	3.2.4 飞思卡尔 PowerQUICC II	32
2.2.1 典型的嵌入式 Linux 开发环境	8	3.2.5 PowerQUICC II Pro	32
2.2.2 启动目标板	9	3.2.6 飞思卡尔 PowerQUICC III	33
2.2.3 引导内核	10	3.2.7 飞思卡尔 QorIQ™	34
2.2.4 内核初始化: 概述	12	3.2.8 AMCC Power 架构	36
2.2.5 第一个用户空间进程: init	13	3.2.9 MIPS	38
2.3 存储	14	3.2.10 Broadcom MIPS	38
2.3.1 闪存	14	3.2.11 其他 MIPS	39
2.3.2 NAND 型闪存	15	3.2.12 ARM	40
2.3.3 闪存的用途	16	3.2.13 德州仪器 ARM	40
2.3.4 闪存文件系统	17	3.2.14 飞思卡尔 ARM	42
2.3.5 内存空间	17	3.2.15 其他 ARM 处理器	43
2.3.6 执行上下文	19	3.3 其他架构	43
2.3.7 进程虚拟内存	20	3.4 硬件平台	43
2.3.8 交叉开发环境	21	3.4.1 CompactPCI	43
2.4 嵌入式 Linux 发行版	22	3.4.2 ATCA	44
2.4.1 商业 Linux 发行版	24	3.5 小结	44

第4章 Linux 内核：不同的视角	46	第6章 用户空间初始化	100
4.1 背景知识	46	6.1 根文件系统	100
4.1.1 内核版本	47	6.1.1 FHS：文件系统层次结构标准	101
4.1.2 内核源码库	48	6.1.2 文件系统布局	101
4.1.3 使用 git 下载内核代码	49	6.1.3 最小化的文件系统	102
4.2 Linux 内核的构造	49	6.1.4 嵌入式根文件系统带来的挑战	103
4.2.1 顶层源码目录	49	6.1.5 试错法	104
4.2.2 编译内核	50	6.1.6 自动化文件系统构建工具	104
4.2.3 内核主体：vmlinux	52	6.2 内核的最后一些引导步骤	104
4.2.4 内核镜像的组成部分	53	6.2.1 第一个用户空间程序	106
4.2.5 子目录的布局	56	6.2.2 解决依赖关系	106
4.3 内核构建系统	57	6.2.3 定制的初始进程	107
4.3.1 .config 文件	57	6.3 init 进程	107
4.3.2 配置编辑器	58	6.3.1 inittab	109
4.3.3 Makefile 目标	61	6.3.2 Web 服务器启动脚本示例	111
4.4 内核配置	66	6.4 初始 RAM 磁盘	112
4.4.1 定制配置选项	68	6.4.1 使用 initrd 进行引导	113
4.4.2 内核 Makefile	71	6.4.2 引导加载程序对 initrd 的支持	113
4.5 内核文档	72	6.4.3 initrd 的奥秘所在：linuxrc	115
4.6 获得定制的 Linux 内核	72	6.4.4 initrd 探究	116
4.7 小结	73	6.4.5 构造 initrd 镜像	116
第5章 内核初始化	75	6.5 使用 initramfs	117
5.1 合成内核镜像：Piggy 及其他	75	6.6 关机	119
5.1.1 Image 对象	78	6.7 小结	120
5.1.2 与具体架构相关的对象	78	第7章 引导加载程序	122
5.1.3 启动加载程序	79	7.1 引导加载程序的作用	122
5.1.4 引导消息	80	7.2 引导加载程序带来的挑战	123
5.2 初始化时的控制流	83	7.2.1 DRAM 控制器	123
5.2.1 内核入口：head.o	84	7.2.2 闪存与 RAM	123
5.2.2 内核启动：main.c	85	7.2.3 镜像的复杂性	124
5.2.3 架构设置	86	7.2.4 执行环境	126
5.3 内核命令行的处理	87	7.3 通用引导加载程序：Das U-Boot	126
5.4 子系统初始化	92	7.3.1 获取 U-Boot	127
5.5 init 线程	94	7.3.2 配置 U-Boot	127
5.5.1 通过 initcalls 进行初始化	95		
5.5.2 initcall_debug	96		
5.5.3 最后的引导步骤	97		
5.6 小结	98		

7.3.3 U-Boot 的监控命令	129	8.3.1 驱动程序中的文件系统 操作	166
7.3.4 网络操作	130	8.3.2 设备号的分配	169
7.3.5 存储子系统	132	8.3.3 设备节点和 mknod	169
7.3.6 从磁盘引导	133	8.4 综合应用	170
7.4 移植 U-Boot	133	8.5 在内核源码树外构建驱动	171
7.4.1 EP405 的 U-Boot 移植	133	8.6 设备驱动程序和 GPL	172
7.4.2 U-Boot Makefile 中的配置 目标	134	8.7 小结	173
7.4.3 EP405 的第一次构建	135	第 9 章 文件系统	174
7.4.4 EP405 处理器初始化	136	9.1 Linux 文件系统概念	175
7.4.5 与具体板卡相关的初始化	138	9.2 ext2	176
7.4.6 移植总结	141	9.2.1 挂载文件系统	177
7.4.7 U-Boot 镜像格式	141	9.2.2 检查文件系统的完整性	178
7.5 设备树对象 (扁平设备树)	143	9.3 ext3	180
7.5.1 设备树源码	145	9.4 ext4	182
7.5.2 设备树编译器	148	9.5 ReiserFS	182
7.5.3 使用 DTB 的其他内核镜像	148	9.6 JFFS2	183
7.6 其他引导加载程序	149	9.7 cramfs	185
7.6.1 Lilo	149	9.8 网络文件系统	187
7.6.2 GRUB	150	9.9 伪文件系统	190
7.6.3 其他更多的引导加载程序	151	9.9.1 /proc 文件系统	191
7.7 小结	152	9.9.2 sysfs	194
第 8 章 设备驱动程序基础	154	9.10 其他文件系统	196
8.1 设备驱动程序的概念	154	9.11 创建简单的文件系统	197
8.1.1 可加载模块	155	9.12 小结	199
8.1.2 设备驱动程序架构	155	第 10 章 MTD 子系统	201
8.1.3 最小设备驱动程序示例	156	10.1 MTD 概述	201
8.1.4 模块构建的基础设施	157	10.1.1 开启 MTD 服务	202
8.1.5 安装设备驱动程序	160	10.1.2 MTD 基础	203
8.1.6 加载模块	160	10.1.3 在目标板上配置 MTD	205
8.1.7 模块参数	161	10.2 MTD 分区	205
8.2 模块工具	162	10.2.1 使用 Redboot 分区表进行 分区	206
8.2.1 insmod	162	10.2.2 使用内核命令行传递分区 信息	210
8.2.2 lsmod	163	10.2.3 映射驱动	211
8.2.3 modprobe	163	10.2.4 闪存芯片驱动	213
8.2.4 depmod	164	10.2.5 与具体板卡相关的初始化	214
8.2.5 rmmod	165		
8.2.6 modinfo	165		
8.3 驱动程序方法	166		

10.3	MTD 工具	215	13.4	追踪和性能评测工具	261
10.4	UBI 文件系统	219	13.4.1	strace	261
10.4.1	配置 UBIFS	219	13.4.2	strace 命令行选项	264
10.4.2	构建 UBIFS 镜像	220	13.4.3	ltrace	266
10.4.3	使用 UBIFS 作为根文件 系统	222	13.4.4	ps	267
10.5	小结	222	13.4.5	top	269
第 11 章	BusyBox	224	13.4.6	mtrace	270
11.1	BusyBox 简介	224	13.4.7	dmalloc	272
11.2	BusyBox 的配置	225	13.4.8	内核 oops	274
11.3	BusyBox 的操作	227	13.5	二进制工具	276
11.3.1	BusyBox 的 init	230	13.5.1	readelf	277
11.3.2	rcS 初始化脚本示例	232	13.5.2	使用 readelf 查看调试信息	278
11.3.3	BusyBox 在目标板上的 安装	233	13.5.3	objdump	280
11.3.4	BusyBox 小应用	235	13.5.4	objcopy	281
11.4	小结	235	13.6	其他二进制实用程序	281
第 12 章	嵌入式开发环境	237	13.6.1	strip	281
12.1	交叉开发环境	237	13.6.2	addr2line	282
12.2	对主机系统的要求	241	13.6.3	strings	282
12.3	为目标板提供服务	242	13.6.4	ldd	282
12.3.1	TFTP 服务器	242	13.6.5	nm	283
12.3.2	BOOTP/DHCP 服务器	243	13.6.6	prelink	283
12.3.3	NFS 服务器	245	13.7	小结	284
12.3.4	目标板使用 NFS 挂载根 文件系统	247	第 14 章	内核调试技术	286
12.3.5	U-Boot 中使用 NFS 挂载根 文件系统的例子	248	14.1	内核调试带来的挑战	286
12.4	小结	250	14.2	使用 KGDB 进行内核调试	287
第 13 章	开发工具	252	14.2.1	KGDB 的内核配置	288
13.1	GNU 调试器 (GDB)	252	14.2.2	在开启 KGDB 时引导 目标板	289
13.1.1	调试核心转储	253	14.2.3	一些有用的内核断点	293
13.1.2	执行 GDB	255	14.2.4	KGDB 与控制台共享一个 串行端口	294
13.1.3	GDB 中的调试会话	257	14.2.5	调试非常早期的内核代码	295
13.2	数据显示调试器	258	14.2.6	主线内核对 KGDB 的支持	296
13.3	cbrowser/cscope	260	14.3	内核调试技术	297
			14.3.1	gdb 远程串行协议	298
			14.3.2	调试优化的内核代码	300
			14.3.3	GDB 的用户自定义命令	307
			14.3.4	有用的内核 GDB 宏	307
			14.3.5	调试可加载模块	315

14.3.6	printk 调试	319	16.4.1	OpenEmbedded 的组成	355
14.3.7	Magic SysReq key	320	16.4.2	BitBake 元数据	355
14.4	硬件辅助调试	321	16.4.3	配方基础	356
14.4.1	使用 JTAG 探测器对闪存 进行编程	322	16.4.4	任务	358
14.4.2	使用 JTAG 探测器进行调 试	323	16.4.5	类	359
14.5	不能启动的情况	326	16.4.6	配置元数据	360
14.5.1	早期的串行端口调试输 出	326	16.4.7	构建镜像	361
14.5.2	转储 printk 的日志缓冲 区	327	16.5	小结	362
14.5.3	使用 KGDB 调试内核异常	328	第 17 章	实时 Linux	363
14.6	小结	329	17.1	什么是实时	363
第 15 章	调试嵌入式 Linux 应用程序	331	17.1.1	软实时	363
15.1	目标调试	331	17.1.2	硬实时	364
15.2	远程(交叉)调试	331	17.1.3	Linux 调度	364
15.3	调试共享程序库	335	17.1.4	延时	364
15.4	调试多个任务	340	17.2	内核抢占	365
15.4.1	调试多个进程	340	17.2.1	抢占的障碍	366
15.4.2	调试多线程应用程序	342	17.2.2	抢占模式	367
15.4.3	调试引导加载程序/闪存 代码	345	17.2.3	SMP 内核	368
15.5	其他远程调试选项	345	17.2.4	抢占延时的根源	369
15.5.1	使用串行端口进行调试	345	17.3	实时内核补丁	369
15.5.2	附着到运行的进程上	346	17.3.1	实时补丁的特性	370
15.6	小结	346	17.3.2	O(1)调度器	371
第 16 章	开源构建系统	348	17.3.3	创建实时进程	372
16.1	为什么使用构建系统	348	17.4	实时内核的性能分析	373
16.2	Scratchbox	349	17.4.1	使用 Ftrace 追踪内核行为	373
16.2.1	安装 Scratchbox	349	17.4.2	检测抢占被关闭的延时	374
16.2.2	创建一个交叉编译目标	350	17.4.3	检测唤醒延时	375
16.3	Buildroot	351	17.4.4	检测中断被关闭的延时	377
16.3.1	安装 Buildroot	352	17.4.5	检测 Soft Lockup	378
16.3.2	配置 Buildroot	352	17.5	小结	378
16.3.3	构建 Buildroot	353	第 18 章	通用串行总线	379
16.4	OpenEmbedded	354	18.1	USB 概述	379
			18.1.1	USB 的物理拓扑结构	379
			18.1.2	USB 的逻辑拓扑结构	381
			18.1.3	USB 版本	382
			18.1.4	USB 连接器	382
			18.1.5	USB 线缆	383