

由浅入深，全面解析基于ARM体系结构的嵌入式Linux系统开发  
抽丝剥茧，采用典型实例，剖析嵌入式开发的技术原理与过程



# ARM 嵌入式

# Linux系统开发详解

弓雷 等编著



DVD-ROM

## 9小时多媒体语音视频讲解

另外赠送34.5小时Linux专题视频、Ubuntu安装文件

- ◎ 由浅入深：从嵌入式系统的软硬件开始讲解，逐步深入到实际开发
- ◎ 内容全面：涵盖嵌入式Linux系统开发的大部分知识点和典型应用
- ◎ 注重原理：对每个知识点都从概念和基本原理进行详细、透彻的分析
- ◎ 实践性强：结合实际应用，讲解时列举了大量典型示例和应用实例
- ◎ 重点突出：着重介绍了Bootloader功能及移植、Linux内核及移植等内容
- ◎ 兼顾难度：结合5个典型案例，介绍了Linux嵌入式驱动开发技术
- ◎ 视频讲解：专门录制了9小时多媒体教学视频讲解书中的重点内容

Linux 典藏大系

ChinaUnix.net



ARM 嵌入式

Linux 系统开发详解

弓雷 等编著

清华大学出版社

北京

## 内 容 简 介

嵌入式系统是目前最流行的计算机应用技术之一。本书由浅入深地讲解基于 ARM 体系结构的嵌入式 Linux 系统开发, 内容包括嵌入式系统的基本概念、应用领域等基础知识; ARM 处理器的体系结构和功能特点; Linux 系统内核结构和移植方法、Bootloader 功能和移植; Linux 应用程序开发, 包括如何操作文件、管理内存、访问串口、多线程和多进程技术, 并且剖析了应用程序的结构和工作流程; Linux 系统设备驱动程序开发, 包括如何编写内核模块、驱动程序分类和功能特点, 并且给出了网络设备驱动、Flash 设备驱动和 USB 设备驱动的实例分析。

本书附带 1 张 DVD 光盘, 内容为专门为本书录制的 9 小时视频、本书涉及的源代码、Ubuntu 安装光盘的镜像文件和另外赠送的 35 小时 Linux 专题学习视频。

本书适合广大从事嵌入式 Linux 系统开发人员、对嵌入式 Linux 系统开发有兴趣的计算机爱好者, 以及大中专院校学生阅读。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

### 图书在版编目 (CIP) 数据

ARM 嵌入式 Linux 系统开发详解 / 弓雷等编著. —北京: 清华大学出版社, 2010.1  
(Linux 典藏大系)

ISBN 978-7-302-20729-0

I. A… II. 弓… III. ①微处理器, ARM—系统设计 ②Linux 操作系统—系统设计  
IV. TP332 TP316.89

中国版本图书馆 CIP 数据核字 (2009) 第 141354 号

责任编辑: 夏兆彦

责任校对: 徐俊伟

责任印制: 孟凡玉

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京密云胶印厂

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 31.25 字 数: 776 千字

(附光盘 1 张)

版 次: 2010 年 1 月第 1 版 印 次: 2010 年 1 月第 1 次印刷

印 数: 1~5000

定 价: 59.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 034378-01

# 前言

随着超大规模集成电路的发展，计算机处理器技术不断提高，计算机芯片的处理能力越来越强，体积越来越小，计算机技术应用到生活的方方面面。与人们日常生活打交道最多的就是嵌入式系统，从目前广泛使用的手机、MP3 播放器到家用电器，嵌入式系统的应用无处不在。嵌入式系统的开发占整个计算机系统开发的比重也越来越高。

嵌入式系统开发与传统的 PC 程序开发不同。嵌入式系统开发涉及软件硬件的开发，是一个协同工作的统一体。目前，已经有许多的嵌入式系统硬件和操作系统软件，其中应用最广泛的是 ARM 嵌入式处理器和 Linux 系统，本书也将以 ARM 处理器和 Linux 作为讲解对象，帮助读者进入嵌入式开发领域。

目前嵌入式开发的书籍种类繁多，但是内容都比较有针对性。主要可以分成 3 类，一类是侧重嵌入式硬件开发；一类侧重嵌入式操作系统原理和设备驱动开发；还有一类是针对嵌入式系统应用程序开发。由于以上 3 类图书都是针对某个特定领域编写的，内容针对性强，不便于初学者学习。

本书从嵌入式初学者的角度出发，力求由浅入深地介绍嵌入式系统开发的各部分知识，帮助初学者快速进入嵌入式开发领域，掌握嵌入式开发的基本技术。

## 本书特色

### 1. 配视频讲解光盘

为了让读者更加快速、直观地学习本书内容，作者专门为本书录制了大量多媒体语音视频进行讲解，读者可以结合视频讲解更高效地学习，学习效果更好。

### 2. 循序渐进，由浅入深

为了让初学者快速进入嵌入式系统开发领域，本书一开始对嵌入式系统的软件和硬件做了全面的介绍，让读者对嵌入式系统有一个基本的认识。同时，在书中讲解的知识点都配备了完整的实例，读者可以通过实例学习嵌入式系统开发的相关知识。

### 3. 技术全面，内容充实

作者本人从事嵌入式系统开发多年，深入了解嵌入式系统开发的各个方面，在书中讲解了嵌入式 Linux 开发的各个要点，包括 Linux 内核的构成、工作流程、驱动程序开发、文件系统、程序库等知识，使读者全面了解嵌入式 Linux 开发的各个知识点。

### 4. 实例讲解，理解深刻

嵌入式 Linux 开发书籍众多，很多书籍偏重理论。本书所有的实例都经过作者验证，

并且有详细的操作过程和实验结果。其次，本书的操作实例有完整的实验环境描述，读者可以通过实例加深对知识点的理解。

## 5. 化整为零，深入剖析

嵌入式系统开发涉及知识面广，技术复杂。本书剥茧抽丝，力求找出开发过程中关键的知识点。从关键点入手，通过简单易懂的例子剖析技术原理，帮助读者掌握复杂的技术。

## 本书内容

第1章：介绍嵌入式系统的基本概念。本章从日常生活的例子入手，给读者对嵌入式系统一个初步认识。然后介绍了嵌入式系统的发展历程，以及基本的嵌入式系统开发知识。

第2章：嵌入式软件和硬件知识。从模拟电路和数字电路的对比入手，介绍嵌入式系统硬件的基本知识。在此基础上，讲解软件的基本概念和开发流程，最后是操作系统的基础知识。

第3章：全面介绍 ARM 处理器。先比较了微控制器和微处理器的异同，然后介绍 ARM 嵌入式处理器。讲解 ARM 处理器的指令集和结构，最后是处理器选型需要考虑的因素。

第4章：引入本书的重点——嵌入式 Linux 操作系统。本章可以分成两部分，前半部分介绍常见的几种嵌入式操作系统，通过对比得出 Linux 系统在嵌入式领域的优势。后半部分讲解嵌入式 Linux 相关的知识。

第5章：本章介绍如何搭建一个嵌入式 Linux 系统的开发环境。嵌入式 Linux 系统开发可以在 Linux 系统上进行，也可以在其他系统（如 Windows 系统）上进行。因此，本章介绍了在两种平台下搭建开发环境的知识点，并且给出了详细的操作步骤。

第6章：由一个最基本的 Linux 应用程序入手，讲解应用程序的工作原理和工作流程。最后还介绍了 Linux 管理编译程序的方法。

第7章：讲解 Linux 应用程序开发的关键技术之一，如何操作内存和文件。内存和文件是应用程序使用最多的两种资源。

第8章：讲解多线程和多进程程序开发。多线程和多进程程序可以提高应用程序的并发处理能力，是现代计算机应用中重要的技术。

第9章：讲解目前应用最广泛的网络功能，如何在应用程序中通过套接字在网络中传输数据。

第10章：串口是嵌入式系统应用最广泛的基本数据传递方式之一，本章讲解串口的基本工作原理、在应用程序中使用串口传递数据。

第11章：图形界面是嵌入式系统的一个热点。本章讲解了如何开发嵌入式系统的图形界面，以应用广泛的 Qt 图形程序库为例，讲解了嵌入式系统移植和开发 Qt 程序的全过程。

第12章：嵌入式系统开发通常涉及多个人相互协作，软件管理的必要性尤为突出。本章讲解了嵌入式系统开发中常见的集中软件管理方法。包括文档管理、代码管理等。每种管理方法都是基于实际的管理软件，并且给出了实际的操作过程。

第13章：本章在第3章的基础上深入讲解 ARM 处理器。主要内容包括 ARM 处理器的体系结构、编程模型、内存管理、外部设备访问等关键技术。在最后给出 S3C2440 ARM

处理器的操作实例。

第 14 章：讲解 Bootloader 工作原理。介绍了嵌入式系统常见的集中 Bootloader，以 U-Boot 为例讲解了 Bootloader 的工作原理，移植 U-Boot 到 ARM 开发板等知识。

第 15 章：解析 Linux 内核。包括 Linux 内核的代码工程结构，主要工作原理等知识。本章内容理论较多，需要结合相关章节理解。

第 16 章：Linux 内核启动过程涉及到软硬件多方面知识，学习 Linux 内核启动过程对理解 Linux 内核结构有很大帮助。

第 17 章：讲解 Linux 文件系统工作原理和制作方法。以 ARM 平台为例制作嵌入式 Linux 系统常用的闪存文件系统。

第 18 章：讲解嵌入式 Linux 系统开发交叉编译工具链的创建和使用。

第 19 章：介绍嵌入式 Linux 常用的命令系统 BusyBox 工作原理、裁剪和交叉编译等知识。

第 20 章：讲解嵌入式 Linux 内核移植的全部过程。

第 21 章：介绍 Linux 内核和应用程序的调试技术。

第 22 章：讲解 Linux 系统设备驱动程序开发的基本知识，包括设备驱动的分类、特点、内核模块编写实例。在最后给出了一个字符设备开发实例。

第 23 章：在第 22 章的基础上详细讲解 Linux 网络设备驱动开发，描述 Linux 内核网络体系结构，并且用实例分析了 DM9000 网络适配器驱动程序。

第 24 章：这一章讲解嵌入式系统中广泛应用的 Flash 设备驱动程序开发，用实例分析了 NOR 和 NAND Flash 设备驱动。

第 25 章：讲解 USB 体系结构和设备驱动开发。内容包括 USB 接口标准、工作流程等，最后给出了 USB 设备驱动开发实例。

## 本书读者对象

- 嵌入式系统移植程序员
- 嵌入式系统应用程序开发人员
- 大中专院校的学生
- 社会培训学生

## 本书作者及编委会成员

本书由弓雷主笔编写。其他参与编写和资料整理的人员还有蔡成立、陈晓莉、陈燕、冯国良、高岱明、黄会、纪奎秀、江莹、李凌、李胜君、李雅娟、刘惠萍、刘水珍、马月桂、秦兰、汪文君、文龙、尹承印、于欣、张国强、张景君、赵书山、周伶俐、朱娜敏。在此对大家表示感谢！

本书编委会成员有欧振旭、陈杰、陈冠军、项宇峰、张帆、陈刚、程彩红、毛红娟、聂庆亮、王志娟、武文娟、颜盟盟、姚志娟、尹继平、张昆、张薛。

编著者

# 目 录

## 第 1 篇 Linux 嵌入式开发基础篇

第 1 章 嵌入式系统入门	2
1.1 什么是嵌入式系统	2
1.2 嵌入式系统应用领域	2
1.2.1 家用电器	2
1.2.2 交通工具	3
1.2.3 公共电子设施	3
1.3 嵌入式系统发展	4
1.3.1 嵌入式微控制器	4
1.3.2 嵌入式微处理器	5
1.3.3 未来嵌入式系统发展的方向	6
1.4 典型的嵌入式系统组成	6
1.5 小结	7
第 2 章 嵌入式软硬件系统	8
2.1 电路基本知识	8
2.1.1 什么是模拟电路	8
2.1.2 什么是数字电路	8
2.1.3 数制转换	9
2.2 计算机组成原理	10
2.2.1 计算机体系的发展	10
2.2.2 中央处理器	10
2.2.3 存储系统	11
2.2.4 总线系统	12
2.2.5 输入输出系统	12
2.3 软件基础知识	12
2.3.1 什么是软件	13
2.3.2 软件开发流程	13
2.3.3 常见的软件开发模型	14
2.3.4 计算机编程语言	14



2.3.5	数据结构	15
2.4	操作系统知识	15
2.4.1	什么是操作系统	16
2.4.2	操作系统的发展历史	16
2.4.3	操作系统由什么组成	16
2.4.4	几种操作系统的设计思路	17
2.4.5	操作系统分类	17
2.5	小结	18
第3章	ARM 处理器	19
3.1	微处理器和微控制器	19
3.2	ARM 处理器介绍	19
3.2.1	ARM 微处理器的应用领域	20
3.2.2	ARM 的功能特点	20
3.3	ARM 指令集	21
3.3.1	算术运算指令	21
3.3.2	逻辑运算指令	22
3.3.3	分支指令	23
3.3.4	数据传送指令	23
3.4	ARM 的结构	24
3.4.1	ARM 体系结构的命名方法	24
3.4.2	处理器系列划分	25
3.4.3	处理器工作模式	25
3.4.4	存储系统	26
3.4.5	寻址方式	27
3.5	ARM 的功能选型	29
3.5.1	ARM 的选型原则	29
3.5.2	几种常见的 ARM 核处理器选型参考	32
3.6	小结	32
第4章	嵌入式 Linux	33
4.1	常见的嵌入式操作系统	33
4.1.1	VxWorks	33
4.1.2	Windows CE	34
4.1.3	PalmOS	34
4.1.4	Symbian	34
4.2	嵌入式 Linux 操作系统	34
4.2.1	什么是 Linux	35
4.2.2	Linux 与 UNIX 的不同——GPL 版权协议介绍	35
4.2.3	Linux 发行版	36
4.2.4	常见的嵌入式 Linux 系统	37



---

4.3	小结	37
<b>第 5 章</b>	<b>建立软件开发环境</b>	<b>38</b>
5.1	独立的 Linux 系统	38
5.1.1	安装 Ubuntu Linux	38
5.1.2	关键一步——系统的网络设置	46
5.1.3	其他必要的设置	46
5.1.4	怎样安装卸载软件	48
5.1.5	怎样配置系统服务	48
5.1.6	安装主要的开发工具	48
5.1.7	安装其他的开发工具和文档	49
5.2	运行在 Windows 上的 Linux 系统	50
5.2.1	什么是 Cygwin	50
5.2.2	如何安装 Cygwin	50
5.2.3	安装开发环境	54
5.3	Linux 常用工具	54
5.3.1	Linux shell 和常用命令	55
5.3.2	文本编辑工具 vi	56
5.3.3	搜索工具 find 和 grep	57
5.3.4	FTP 工具	61
5.3.5	串口工具 minicom	63
5.4	Windows 常用工具	65
5.4.1	代码编辑管理工具 Source Insight	65
5.4.2	串口工具 XShell	71
5.5	ARM 集成开发环境 ADS	72
5.5.1	ADS 集成开发环境介绍	73
5.5.2	配置 ADS 调试环境	73
5.5.3	建立自己的工程	75
5.6	小结	78
<b>第 6 章</b>	<b>第一个 Linux 应用程序</b>	<b>79</b>
6.1	向世界问好——Hello,World!	79
6.1.1	用 vi 编辑源代码文件	79
6.1.2	用 gcc 编译程序	80
6.1.3	执行程序	80
6.2	程序背后做了什么	80
6.2.1	程序执行的过程	81
6.2.2	窥视程序执行中的秘密	82
6.2.3	动态库的作用	83
6.3	程序如何来的——编译的全部过程	84
6.3.1	编译源代码	84

6.3.2	连接目标文件到指定的库	85
6.4	更简单的办法——用 Makefile 管理工程	86
6.4.1	什么是 Makefile	87
6.4.2	它是如何工作的	87
6.4.3	如何使用 Makefile	87
6.4.4	好的源代码管理习惯	88
6.5	小结	88

## 第 2 篇 Linux 嵌入式开发应用篇

第 7 章	Linux 应用程序编程基础	90
7.1	内存管理和使用	90
7.1.1	堆和栈的区别	90
7.1.2	内存管理函数 malloc()和 free()	93
7.1.3	实用的内存分配函数 calloc()和 realloc()	93
7.1.4	内存管理编程实例	94
7.2	ANSI C 文件管理	97
7.2.1	文件指针和流	97
7.2.2	存储方式	98
7.2.3	标准输入、标准输出和标准错误	98
7.2.4	缓冲	98
7.2.5	打开关闭文件	99
7.2.6	读写文件	100
7.2.7	文件流定位	103
7.2.8	ANSI C 文件编程实例	103
7.3	POSIX 文件 I/O 编程	104
7.3.1	底层的文件 I/O 操作	104
7.3.2	文件描述符	104
7.3.3	创建/打开/关闭文件	105
7.3.4	读写文件内容	108
7.3.5	文件内容定位	109
7.3.6	修改已打开文件的属性	110
7.3.7	POSIX 文件编程实例	110
7.4	小结	112
第 8 章	开发多进程/线程程序	113
8.1	多进程开发	113
8.1.1	什么是进程	113
8.1.2	进程环境和属性	114

8.1.3	创建进程	115
8.1.4	等待进程结束	117
8.1.5	退出进程	118
8.1.6	常用进程间通信的方法	120
8.1.7	进程编程实例	124
8.2	多线程开发	125
8.2.1	线程的概念	126
8.2.2	进程和线程对比	126
8.2.3	创建线程	127
8.2.4	取消线程	128
8.2.5	等待线程	129
8.2.6	使用 pthread 库线程操作实例	129
8.3	小结	131
<b>第 9 章</b>	<b>网络通信应用</b>	<b>132</b>
9.1	网络通信基础	132
9.1.1	TCP/IP 协议族	132
9.1.2	IP 协议	133
9.1.3	TCP 协议	135
9.1.4	UDP 协议	135
9.1.5	学习分析协议的方法	135
9.2	Socket 通信基本概念	137
9.2.1	创建 socket 对象	138
9.2.2	面向连接的 Socket 通信实现	138
9.2.3	面向连接的 Echo 服务编程实例	140
9.2.4	无连接的 Socket 通信实现	144
9.2.5	无连接的时间服务编程实例	145
9.3	Socket 高级应用	148
9.3.1	Socket 超时处理	148
9.3.2	使用 Select 机制处理多连接	149
9.3.3	使用 Poll 机制处理多连接	151
9.3.4	多线程环境 Socket 编程	152
9.4	小结	152
<b>第 10 章</b>	<b>串行口通信编程</b>	<b>154</b>
10.1	串口介绍	154
10.1.1	什么是串口	154
10.1.2	串口工作原理	154
10.1.3	串口流量控制	155
10.2	开发串口应用程序	156
10.2.1	操作串口需要用到的头文件	156

10.2.2	串口操作方法	157
10.2.3	串口属性设置	157
10.2.4	与 Windows 串口终端通信	161
10.3	串口应用实例——手机短信发送	163
10.3.1	PC 与手机连接发送短信的物理结构	164
10.3.2	AT 指令介绍	164
10.3.3	GSM AT 指令集	165
10.3.4	PDU 编码方式	165
10.3.5	建立与手机的连接	167
10.3.6	使用 AT 指令发送短信	167
10.4	小结	171
<b>第 11 章</b>	<b>嵌入式 GUI 程序开发</b>	<b>172</b>
11.1	Linux GUI 介绍	172
11.1.1	Linux GUI 的发展	172
11.1.2	常见的嵌入式 GUI	173
11.2	开发图形界面程序	174
11.2.1	安装 Qt 开发环境	174
11.2.2	建立简单的 Qt 程序	176
11.2.3	Qt 库编程结构	178
11.3	深入 Qt 编程	179
11.3.1	使用 Widget	179
11.3.2	对话框程序设计	181
11.3.3	信号与槽系统	185
11.4	移植 Qtopia 到 ARM 开发板	186
11.4.1	需要的资源	186
11.4.2	准备工作	187
11.4.3	编译主机 Qt 工具	187
11.4.4	交叉编译 qtopia	189
11.5	小结	190
<b>第 12 章</b>	<b>软件项目管理</b>	<b>191</b>
12.1	源代码管理	191
12.1.1	什么是软件的版本	191
12.1.2	版本控制的概念	192
12.2	版本控制系统 Subversion	193
12.2.1	在 Linux 系统下使用 Subversion 服务端	193
12.2.2	在 Windows 系统下使用 TortoiseSVN 客户端	195
12.3	开发文档管理——常见的开发文档	199
12.3.1	可行性研究报告	200
12.3.2	项目开发计划	200

12.3.3	软件需求说明书	200
12.3.4	概要设计	200
12.3.5	详细设计	200
12.3.6	用户手册	201
12.3.7	其他文档	201
12.4	使用 trac 管理软件开发文档	201
12.4.1	安装 trac	201
12.4.2	配置 trac 基本设置	202
12.4.3	配置 trac 全局脚本	204
12.4.4	设置 trac 的 Web 界面	205
12.5	Bug 跟踪系统	208
12.5.1	Bug 管理的概念和作用	208
12.5.2	使用 Bugzilla 跟踪 Bug	208
12.6	小结	211

## 第 3 篇 Liux 系统篇

第 13 章	ARM 体系结构及开发实例	214
13.1	ARM 体系结构介绍	214
13.1.1	ARM 体系结构	214
13.1.2	ARM 指令集介绍	215
13.2	编程模型	216
13.2.1	数据类型	216
13.2.2	处理器模式	216
13.2.3	寄存器	217
13.2.4	通用寄存器	217
13.2.5	程序状态寄存器	219
13.2.6	异常处理	219
13.2.7	内存和内存 I/O 映射	220
13.3	内存管理单元	221
13.3.1	内存管理介绍	221
13.3.2	内存访问顺序	222
13.3.3	地址翻译过程	222
13.3.4	访问权限	223
13.4	常见接口和控制器	223
13.4.1	GPIO 接口	223
13.4.2	中断控制器	224
13.4.3	RTC 控制器	224
13.4.4	看门狗定时器	225

---

13.4.5	试验：使用 GPIO 点亮 LED .....	226
13.5	小结 .....	227
<b>第 14 章</b>	<b>深入 Bootloader .....</b>	<b>228</b>
14.1	初识 Bootloader .....	228
14.1.1	PC（个人电脑）上的 Bootloader .....	228
14.1.2	什么是嵌入式系统的 Bootloader .....	229
14.1.3	嵌入式系统常见的 Bootloader .....	229
14.2	U-Boot 分析 .....	230
14.2.1	获取 U-Boot .....	230
14.2.2	U-Boot 工程结构分析 .....	230
14.2.3	U-Boot 总体工作流程 .....	232
14.3	U-Boot 启动流程分析 .....	233
14.3.1	_start 标号 .....	233
14.3.2	reset 标号 .....	235
14.3.3	cpu_init_crit 标号 .....	237
14.3.4	lowlevel_init 标号 .....	238
14.3.5	relocate 标号 .....	239
14.3.6	start_armboot()函数 .....	240
14.3.7	main_loop()函数 .....	244
14.4	移植 U-Boot 到开发板 .....	248
14.4.1	U-Boot 移植的一般步骤 .....	248
14.4.2	移植 U-Boot 到目标开发板 .....	249
14.4.3	移植 U-Boot 的常见问题 .....	250
14.5	小结 .....	251
<b>第 15 章</b>	<b>解析 Linux 内核 .....</b>	<b>252</b>
15.1	基本知识 .....	252
15.1.1	什么是 Linux 内核 .....	252
15.1.2	Linux 内核版本 .....	253
15.1.3	如何获取 Linux 内核代码 .....	253
15.1.4	编译内核 .....	253
15.2	Linux 内核的子系统 .....	260
15.2.1	系统调用接口 .....	260
15.2.2	进程管理子系统 .....	261
15.2.3	内存管理子系统 .....	262
15.2.4	虚拟文件系统 .....	263
15.2.5	网络堆栈 .....	264
15.2.6	设备驱动 .....	265
15.2.7	依赖体系结构的代码 .....	266
15.3	Linux 内核代码的工程结构 .....	266

15.3.1	源代码目录布局	266
15.3.2	几个重要的 Linux 内核文件	267
15.4	内核编译系统	268
15.4.1	内核编译系统基本架构	269
15.4.2	内核顶层 Makefile 分析	269
15.4.3	内核编译文件分析	274
15.4.4	目标文件清除机制	278
15.4.5	编译辅助程序	279
15.4.6	KBuild 变量	281
15.5	小结	282
<b>第 16 章</b>	<b>嵌入式 Linux 启动流程</b>	<b>283</b>
16.1	Linux 内核初始化流程	283
16.2	PC 的初始化流程	284
16.2.1	PC BIOS 功能和作用	284
16.2.2	硬盘的数据结构	285
16.2.3	完整的初始化流程	286
16.3	嵌入式系统的初始化	286
16.4	Linux 内核初始化	287
16.4.1	解压缩内核映像	287
16.4.2	进入内核代码	289
16.5	启动 init 内核进程	291
16.6	根文件系统初始化	292
16.6.1	根文件系统介绍	292
16.6.2	挂载虚拟文件系统	294
16.7	内核交出权限	300
16.8	init 进程	300
16.9	初始化 RAM Disk	301
16.9.1	RAM Disk 介绍	301
16.9.2	如何使用 RAM Disk	302
16.9.3	实例：使用 RAM Disk 作为根文件系统	302
16.10	小结	303
<b>第 17 章</b>	<b>Linux 文件系统</b>	<b>304</b>
17.1	Linux 文件管理	304
17.1.1	文件和目录的概念	304
17.1.2	文件的结构	305
17.1.3	文件的类型	305
17.1.4	文件系统的目录结构	307
17.1.5	文件和目录的存取权限	308
17.1.6	文件系统管理	310



17.2	Linux 文件系统原理	311
17.2.1	非日志文件系统	311
17.2.2	日志文件系统	311
17.3	常见的 Linux 文件系统	312
17.3.1	ext2 文件系统	312
17.3.2	ext3 文件系统	315
17.3.3	ReiserFS 文件系统	315
17.3.4	JFFS 文件系统	316
17.3.5	cramfs 文件系统	318
17.4	其他文件系统	319
17.4.1	网络文件系统	319
17.4.2	/proc 影子文件系统	320
17.5	小结	323
<b>第 18 章</b>	<b>建立交叉编译工具链</b>	<b>324</b>
18.1	什么是交叉编译	324
18.2	需要哪些东西	325
18.3	手工创建工具链	325
18.3.1	准备工作——获取源代码	326
18.3.2	开始了——建立工作环境	326
18.3.3	建立 Linux 内核头文件	327
18.3.4	编译安装 binutils	330
18.3.5	编译安装 gcc 的 C 编译器	331
18.3.6	编译安装 glibc 库	332
18.3.7	编译安装 gcc 的 C、C++ 编译器	333
18.3.8	最后的工作	334
18.4	使用脚本创建工具链	334
18.5	更简便的方法——获取已编译好的交叉编译环境	336
18.6	小结	336
<b>第 19 章</b>	<b>强大的命令系统 BusyBox</b>	<b>337</b>
19.1	BusyBox 简介	337
19.1.1	简单易懂的 BusyBox	337
19.1.2	BusyBox 工作原理	338
19.1.3	安装 BusyBox	339
19.2	交叉编译 BusyBox	341
19.3	使用 BusyBox	343
19.3.1	BusyBox 初始化	344
19.3.2	目标板 BusyBox 安装	346
19.4	小结	348

第 20 章 Linux 内核移植 .....	349
20.1 Linux 内核移植要点 .....	349
20.2 平台相关代码结构 .....	350
20.3 建立目标平台工程框架 .....	351
20.3.1 加入编译菜单项 .....	352
20.3.2 设置宏与代码文件的对应关系 .....	352
20.3.3 测试工程框架 .....	353
20.4 建立目标平台代码框架 .....	354
20.4.1 ARM 处理器相关结构 .....	354
20.4.2 建立 machine_desc 结构 .....	356
20.4.3 加入处理函数 .....	356
20.4.4 加入定时器结构 .....	357
20.4.5 测试代码结构 .....	357
20.5 构建目标板代码 .....	357
20.5.1 处理器初始化 .....	358
20.5.2 端口映射 .....	360
20.5.3 中断处理 .....	361
20.5.4 定时器处理 .....	361
20.5.5 编译最终代码 .....	361
20.6 小结 .....	362
第 21 章 内核和应用程序调试技术 .....	363
21.1 使用 GDB 调试应用程序概述 .....	363
21.2 基本的调试技术 .....	363
21.2.1 列出源代码 .....	366
21.2.2 断点管理 .....	368
21.2.3 执行程序 .....	370
21.2.4 显示程序变量 .....	370
21.2.5 信号管理 .....	371
21.2.6 调试实例 .....	371
21.3 多进程调试 .....	372
21.4 调试意外终止的程序 .....	374
21.5 内核调试技术 .....	376
21.5.1 printk 打印调试信息 .....	376
21.5.2 使用/proc 虚拟文件系统 .....	377
21.5.3 使用 KDB 调试工具 .....	379
21.5.4 KDB 调试指令 .....	382
21.6 小结 .....	384