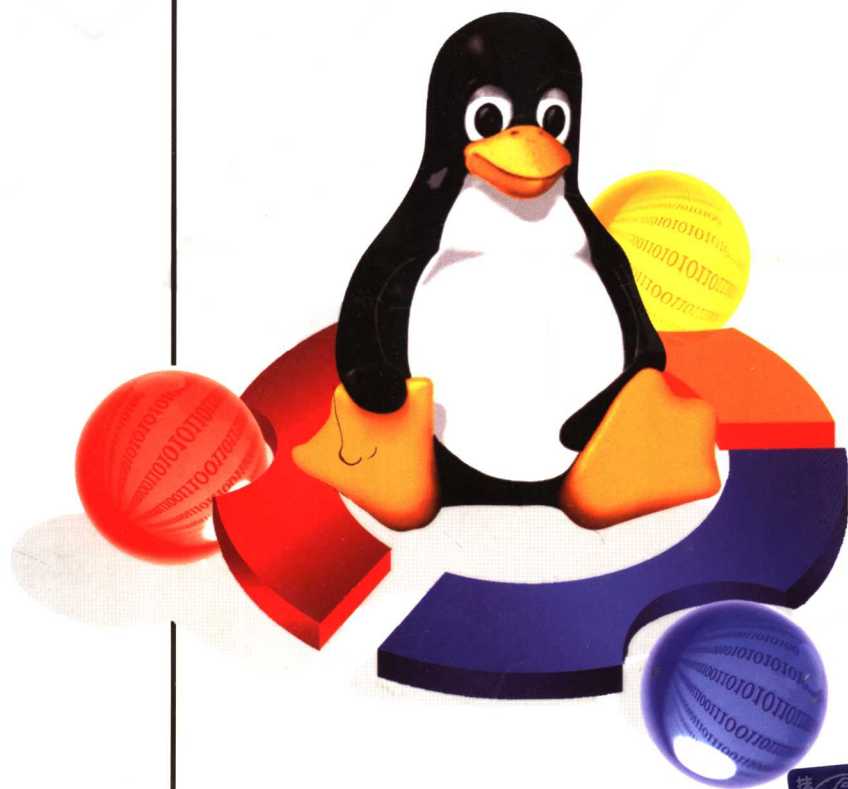


ARM 嵌入式

Linux 系统开发 从入门到精通

李亚锋 欧文盛 等编著



附光盘

- 3个开发工具源码包
- 14个项目实例源文件



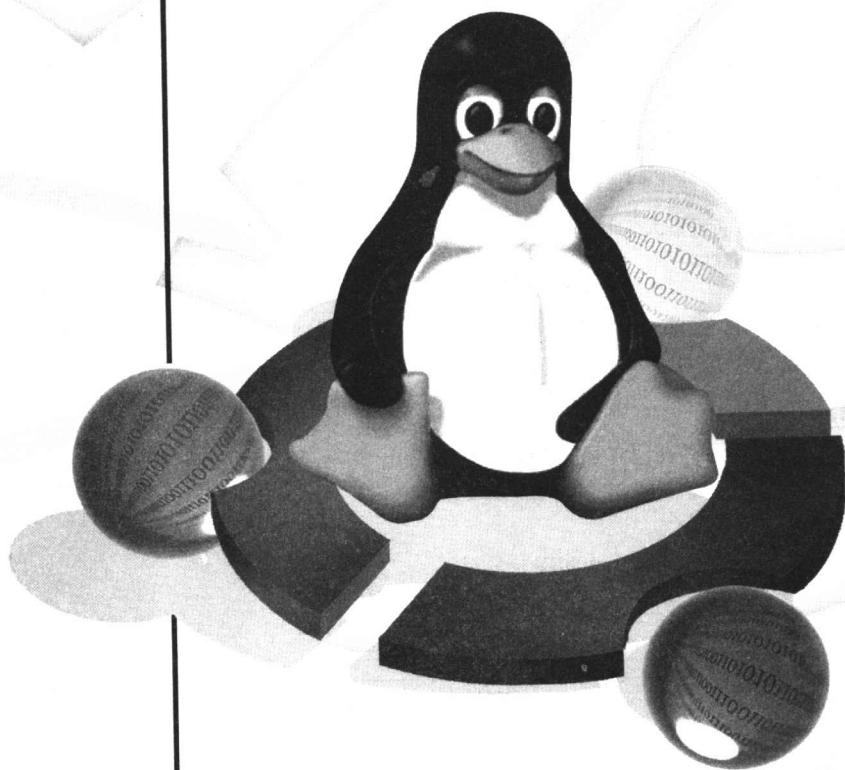
清华大学出版社

ARM嵌入式

Linux 系统开发

从入门到精通

李亚锋 欧文盛 等编著



附光盘

- 3个开发工具源码包
- 14个项目实例源文件

清华大学出版社
北京

内 容 简 介

Linux 是一个成熟而稳定的开放源代码操作系统, 将 Linux 植入嵌入式设备具有众多的优点。本书以应用最广泛的新一代 ARM9 处理器为对象, 介绍了 ARM Linux 系统移植, 包括嵌入式系统开发入门, 交叉编译器的构建, BootLoader 的移植与实现及 Linux 2.6 内核的编译与下载; ARM Linux 的驱动程序开发, 包括最常见的字符设备驱动的分析, 块设备驱动的分析以及网络设备驱动的分析; Qt GUI 开发, 包括 Qt 的具体安装, Qt 的核心技术以及最新的 Qtopia Core 开发环境; 最后通过实例来讲述 Qtopia Core 开发过程。

本书通俗易懂, 可作为高等院校电子类、电气类、控制类、计算机类专业本科生、研究生学习嵌入式 Linux 的参考书目或自学教材, 也可供广大希望转入嵌入式领域的科研和工程技术人员参考使用, 还可作为广大嵌入式培训班的教材和教辅材料。

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 13501256678 13801310933

图书在版编目 (CIP) 数据

ARM 嵌入式 Linux 系统开发从入门到精通 / 李亚锋等编著. —北京: 清华大学出版社, 2007. 8
ISBN 978-7-302-15687-1

I. A… II. 李… III. Linux 操作系统 IV. TP316.89

中国版本图书馆 CIP 数据核字 (2007) 第 105492 号

责任编辑: 夏兆彦 王冰飞

责任校对: 张 剑

责任印制: 王秀菊

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

c-service@tup.tsinghua.edu.cn

社 总 机: 010-62770175 邮购热线: 010-62786544

投稿咨询: 010-62772015 客户服务: 010-62776969

印 刷 者: 北京鑫丰华彩印有限公司

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 203 × 260 印 张: 20 字 数: 520 千字

附光盘 1 张

版 次: 2007 年 8 月第 1 版 印 次: 2007 年 8 月第 1 次印刷

印 数: 1 ~ 4000

定 价: 39.00 元

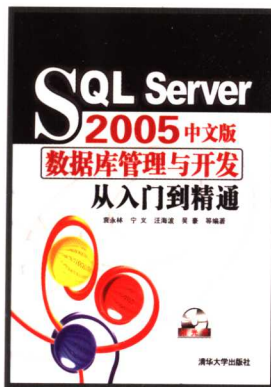
本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 025184-01



ISBN 978-7-302-14983-5
 定价:49.00元(附光盘1张)



ISBN 978-7-302-15080-0
 定价:59.00元(附光盘1张)

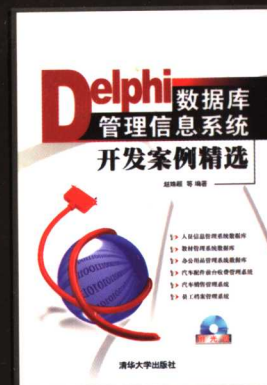


ISBN 978-7-302-15137-1
 定价:86.00元(附光盘1张)



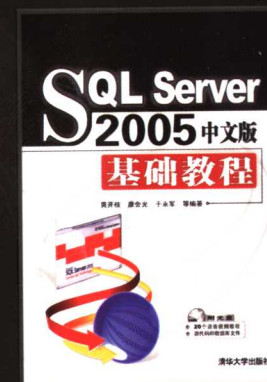
ISBN 978-7-302-14088-7

定价:49.00元(附光盘1张)



ISBN 978-7-302-14030-6

定价:42.00元(附光盘1张)



ISBN 978-7-302-14972-9

定价:59.80元(附光盘1张)

封面设计:  子时文化 (010)86390064 · 俞兆君

FOREWORD

前言

由于芯片、软件、网络和传感器等技术的不断发展，嵌入式系统正在成为未来社会的“数字基因”。如今，人类已经进入了后 PC 时代，嵌入式技术已被广泛应用于科学研究、工程设计、军事技术以及文艺、商业等方面，成为后 PC 时代的主力军。与此同时，嵌入式 Linux 操作系统也在嵌入式领域蓬勃发展，它不仅继承了 Linux 的源码开放、内核稳定性强、软件丰富等特点，而且还支持几乎所有的主流处理器和硬件平台。嵌入式硬件系统和 Linux 系统的有机结合，成为后 PC 时代计算机最普遍的应用形式。嵌入式 Linux 技术在中国有巨大的发展潜力和市场需求。有数据显示，未来两年里，在计算机、消费电子、通信、汽车电子、工业控制和军事国防这六大主要应用领域，嵌入式 Linux 产品将达到 80 亿美元的市场规模，可见这个行业的前景是非常乐观的。当然，Linux 嵌入式操作系统本身也有一定的局限性，就是开发难度过高，对于企业需要很高的技术实力。这就要求 Linux 系统厂商们不光要利用 Linux，更要掌握 Linux。此外，社会需要更多人加入到学习和使用 Linux 的行列中来。

嵌入式 Linux 属于一个交叉学科，并且也是一个高起点的学科，它涵盖了微电子技术、电子信息技术、计算机软件和硬件等多项技术领域的应用。另外，学习嵌入式 Linux 最好具备相应的嵌入式开发板和软件，还需要有经验的人进行指导开发，目前国内大部分高校都很难达到这种要求，这也造成了目前国内嵌入式 Linux 开发人才极其缺乏的局面。

很多希望学习嵌入式 Linux 的人已经具备了一定的硬件知识，并且对操作系统原理、数据结构等都有相当的了解，但在 Linux 技术方面都是零起点。目前嵌入式 Linux 的书籍也是非常之多，但大部分都要求读者有一定的 Linux 使用基础，对于初学者来说真的非常困难。写这本书的主要目的就是能对那些没有 Linux 开发经验的初学者有个很好的指导参考作用，让他们少走弯路。

其次，笔者希望通过写书来总结这几年在工作中的项目经验，与更多的读者分享自己的技术，也是对自己所做项目的一个巩固。通过写这本书，让笔者更加清楚了实践与理论之间的联系，书中的每个章节都结合自己的亲身经验和教训进行讲解。

本书特点

首先，本书涵盖了嵌入式 Linux 系统中最重要的 3 个部分：ARM Linux 系统移植、ARM Linux 驱动程序开发以及 Qt GUI 开发，这在同类书籍中比较少见。其次，本书的讲述不依赖于具体某个厂家的开发板，这样读者使用任意一款类似的开发板就可以进行实践学习，同时对于没有开发板的读者也可以学到更多的知识。另外，本书提供了书中出现的所有实例的源代码，便于读者参考使用，更重要的是读者不用手动输入这些代码，从而节

省了时间。

本书主要内容

本书分为3部分，共12章，每一部分由4章组成。

第一部分讲述 ARM Linux 系统移植，首先第1章讲述嵌入式系统开发入门，主要针对初学者，讲述嵌入式系统的概要、ARM 处理器、ADS 工具、Linux 开发环境，以及 Linux 内核源码等。接着第2章讲述交叉编译工具链的构建，主要讲述交叉工具链的作用、使用分步法构建交叉工具链和使用 Crosstool 工具构建交叉工具链。第3章讲述嵌入式系统的 BootLoader，主要讲述嵌入式 BootLoader 的作用、基于 S3C2410 开发板的 U-Boot 分析与移植以及自己设计 BootLoader 的方法。最后，第4章讲述嵌入式 Linux 内核移植，主要讲述移植的基本概念、内核配置、内核编译、内核下载以及构建根文件系统。

第二部分讲述 ARM Linux 驱动程序开发，首先，第5章讲述 ARM Linux 驱动程序开发入门，主要讲述嵌入式 Linux 驱动程序、简单的内核模块程序分析，以及 Linux 驱动开发的基本要点。第6章讲述字符设备驱动程序，主要讲述字符设备驱动相关的重要数据结构、字符设备驱动开发实例——触摸屏设备驱动开发。第7章讲述块设备驱动程序，主要讲述块设备相关的数据结构、块设备驱动开发实例——MMC/SD 设备驱动开发。最后，第8章讲述网络设备驱动程序，主要讲述网络设备驱动相关的重要数据结构、网络设备驱动开发实例——CS8900A 网卡驱动开发。

第三部分讲述 Qt GUI 开发，首先，第9章介绍 Qt 的概要知识，包括 Linux 桌面 GUI 系统、Qt/X11、Qtopia Core 等，使读者对 Qt 及其在 Linux GUI 系统中的作用有个大概了解。第10章讲述了 Qt/X11 的安装以及非常详细的应用实例，使读者可以轻松地编写基本的 Qt 程序。第11章深入讨论了一些 Qt 的核心技术，重点是以 Qt 对象模型为基础的信号和槽等机制，通过剖析 Qt 的源代码深入地学习 Qt 的这些核心技术，同时也为读者今后对 Qt 源代码的自行研习打下基础。最后，第12章重点讲述 Qtopia Core 和 Qt/X11 的一些不同之处，包括轻量级的窗口系统、QCOPI 进程间通信机制及调试工具 qvfb 等，使读者在熟悉了 Qt/X11 的基础上能够很快过渡到 Qtopia Core 开发。

本书的读者对象

本书通俗易懂，可作为高等院校电子类、电气类、控制类、计算机类专业本科生、研究生学习嵌入式 Linux 的参考书目或自学教材，也可供广大希望转入嵌入式领域的科研和工程技术人员参考使用，还可作为广大嵌入式培训班的教材和教辅材料。

本书的编写得到了很多人的帮助和支持。首先要感谢这本书的另外一位作者欧文盛，书中 Qt GUI 部分（第9章~第12章）主要由他来完成，他这几年一直在国际知名的通信公司从事 Qt 方面的开发工作。其次，要感谢我的妻子，很特殊的是我写这本书的时间正是我妻子怀孕的期间，其实在写这本书之前已经得知妻子怀孕的消息，本想放弃编写，但是妻子却很坚定地支持我写这本书。所以，我认为这本书的完成离不开她对我的默默支持。还要感谢我的岳父、岳母，是他们对我妻子这段时间的精心照顾，才使得我有更多的时间投入到写书中。

最后，要感谢威盛电子的李松、易宏宇、周志勇、张磊等，他们为本书的完成也提供了很多的帮助。

鉴于作者水平有限，加之时间仓促，书中一定有不少不足与疏漏之处，希望得到广大读者的批评与指正。有兴趣的读者可以发送 E-mail 到 lyf99526@yahoo.com.cn 或登录笔者的个人 Blog 来做技术上的交流 (<http://mike2linus.blog.com.cn/>)。

作者
2007年6月

CONTENTS

目 录

第一部分 ARM Linux 系统移植

第 1 章 嵌入式系统开发入门	2
1.1 嵌入式系统介绍	2
1.1.1 嵌入式系统概述	2
1.1.2 嵌入式系统组成	4
1.2 ARM 介绍	5
1.2.1 ARM 处理器介绍	5
1.2.2 ARM 处理器的选型	6
1.2.3 S3C2410 微处理器介绍	7
1.3 ADS 集成开发环境介绍	8
1.3.1 ADS 软件组成	8
1.3.2 使用 Code Warrior IDE	13
1.3.3 使用 AXD IDE	16
1.4 嵌入式 Linux 开发介绍	19
1.4.1 Linux 历史	19
1.4.2 Linux 开发环境	20
1.4.3 GCC 介绍	21
1.4.4 GNU Make 介绍	23
1.4.5 GDB 介绍	26
1.4.6 ARM Linux 系统开发流程	29
1.5 Linux 内核介绍	30
1.5.1 Linux 内核目录结构	31
1.5.2 如何阅读 Linux 内核源代码	33
1.6 本章小结	34
1.7 常见问题	34
第 2 章 交叉编译工具链的构建	36
2.1 交叉编译工具链介绍	36
2.2 构建 ARM Linux 交叉编译工具链	36
2.2.1 分步构建交叉编译链	37
2.2.2 用 Crosstool 工具构建交叉工具链	43
2.3 本章小结	46
2.4 常见问题	46
第 3 章 嵌入式系统的 BootLoader	48
3.1 BootLoader 概述	48
3.2 常用的嵌入式 Linux BootLoader	49

3.2.1 U-Boot	49	3.6 常见问题	85
3.2.2 VIVI	50	第4章 嵌入式 Linux 内核移植	86
3.2.3 Blob	51	4.1 移植的基本概念	86
3.2.4 RedBoot	51	4.2 内核移植的准备	87
3.2.5 ARMboot	51	4.3 内核移植	88
3.2.6 DIY	51	4.3.1 内核配置	88
3.3 基于 S3C2410 开发板的 BootLoader 实现	52	4.3.2 内核编译	97
3.3.1 S3C2410 开发板介绍	52	4.3.3 内核下载	98
3.3.2 U-Boot 分析与移植	53	4.4 建立 Linux 根文件系统	98
3.4 基于 S3C2410 开发板自己编写 BootLoader	77	4.4.1 根文件系统的基本介绍	99
3.4.1 设计系统的启动流程	77	4.4.2 建立根文件系统	102
3.4.2 BootLoader 的具体实现	78	4.5 本章小结	105
3.5 本章小结	84	4.6 常见问题	106

第二部分 ARM Linux 设备驱动程序开发

第5章 ARM Linux 驱动程序开发入门	110	6.2.2 S3C2410 触摸屏工作原理	149
5.1 嵌入式 Linux 驱动程序介绍	110	6.2.3 S3C2410 的 ADC 和触摸屏接 口特殊寄存器	150
5.1.1 驱动程序的作用	110	6.2.4 触摸屏驱动概要设计	154
5.1.2 Linux 设备驱动程序分类	111	6.2.5 触摸屏驱动程序分析	155
5.2 最简单的内核模块举例	112	6.2.6 配置和编译驱动程序	164
5.2.1 编写 Hello World 模块	113	6.2.7 测试触摸屏驱动程序	165
5.2.2 编写 Hello World 模块的 Makefile	114	6.2.8 触摸屏的校准	166
5.2.3 加载和卸载 Hello World 模块	115	6.3 本章小结	167
5.3 Linux 驱动程序开发要点	116	6.4 常见问题	167
5.3.1 内存与 I/O 端口	116	第7章 块设备驱动程序	169
5.3.2 并发控制	120	7.1 块设备驱动介绍	169
5.3.3 阻塞与非阻塞	125	7.1.1 块设备驱动相关的重要结构	169
5.3.4 中断处理	126	7.1.2 请求处理	174
5.3.5 内核调试	133	7.2 块设备驱动开发实例	176
5.4 本章小结	137	7.2.1 MMC/SD 介绍	176
5.5 常见问题	137	7.2.2 S3C2410 提供的 SDI 接口	179
第6章 字符设备驱动程序	139	7.2.3 SDI 相关的寄存器	180
6.1 字符设备驱动介绍	139	7.2.4 MMC/SD 驱动概要设计	184
6.1.1 字符设备驱动相关的重要结构	139	7.2.5 MMC 驱动程序分析	186
6.1.2 主、次设备号	145	7.2.6 S3C2410 SDI 接口驱动分析	192
6.2 字符设备驱动开发实例	148	7.2.7 配置和编译驱动程序	193
6.2.1 四线电阻式触摸屏原理	148		

7.3 本章小结	194	8.2 网络设备驱动开发实例	200
7.4 常见问题	194	8.2.1 CS8900A 介绍	200
第 8 章 网络设备驱动程序	195	8.2.2 CS8900A 网卡驱动概要设计	202
8.1 网络设备驱动介绍	195	8.2.3 CS8900A 适配器驱动程序分析	204
8.1.1 网络设备驱动相关的重要结构	195	8.3 本章小结	213
8.1.2 常见的网络术语	198	8.4 常见问题	214

第三部分 Qt GUI 开发

第 9 章 Qt 概述	216	10.4 本章小结	249
9.1 Linux 下的 GUI 介绍	216	10.5 常见问题	249
9.1.1 Linux 桌面 GUI 系统	216	第 11 章 Qt 核心技术	251
9.1.2 嵌入式 Linux 下的 GUI 系统	220	11.1 信号 (Signals) 和槽 (Slots)	251
9.2 Qt/X11 介绍	221	11.1.1 常见的 GUI 组件通信方式	251
9.2.1 Qt 的历史和 Qt/X11 的由来	221	11.1.2 Qt 中的信号和槽 (Signals and Slots)	254
9.2.2 Qt/X11 的版权问题	222	11.1.3 自定义信号和槽的小例子	256
9.2.3 Qt/X11 及 Qt/Windows 的系统架构图对比	222	11.2 Qt 对象模型	259
9.2.4 Qt 的特性简介	222	11.2.1 元对象系统	259
9.3 Qtopia Core 介绍	223	11.2.2 信号和槽机制的实现	266
9.3.1 Qtopia Core 与 Qt/Embedded	223	11.2.3 元对象编译器 moc	275
9.3.2 Qtopia Core 的体系结构	224	11.3 Qt 的窗口系统	279
9.4 本章小结	225	11.3.1 窗口部件之间的树型结构	279
9.5 常见问题	225	11.3.2 窗口部件的布局管理	281
第 10 章 Qt/X11 初步	226	11.4 国际化	285
10.1 Qt/X11 的安装	226	11.4.1 Qt 国际化的基本步骤	285
10.1.1 Qt/X11 的下载及双重授权问题的说明	226	11.4.2 动态改变语言的小例子	286
10.1.2 Qt/X11 的安装详解	227	11.4.3 一些注意事项	291
10.2 Qt 下的 Hello World	229	11.5 本章小结	293
10.3 温度转换的小例子	230	11.6 常见问题	293
10.3.1 背景知识	230	第 12 章 Qtopia Core	294
10.3.2 Quit 按钮	231	12.1 Qtopia Core 的安装	294
10.3.3 摄氏温度的显示	234	12.2 Frame Buffer 和 qvfb	295
10.3.4 华氏温度的显示	237	12.2.1 Frame Buffer	296
10.3.5 华氏温度和摄氏温度之间的转换	240	12.2.2 编译 qvfb	298
10.3.6 保存当前的数值	244	12.2.3 在 qvfb 上运行 Qtopia Core 程序	298

12.3 移植 Qt/X11 程序到 Qtopia Core 中.....	299	12.7 常见问题.....	307
12.4 轻量级的窗口系统.....	301		
12.5 进程间通信.....	304	参考文献	309
12.6 本章小结.....	307		

第一部分

ARM Linux 系统移植

为了能让读者快速地了解嵌入式 ARM Linux 系统的开发过程，本书第一部分讲述 ARM Linux 的系统移植，这部分内容在实际工作中比较常见，是嵌入式 Linux 开发人员应该掌握的技能。该部分由 4 章组成：第 1 章为 ARM 嵌入式 Linux 系统概述，作为嵌入式开发入门的一章，非常适合初学者阅读，它的内容包括嵌入式系统、ARM、ADS 集成开发工具、嵌入式 Linux 开发以及 Linux 内核，对那些刚接触嵌入式 Linux 开发的读者来说，通过本章学习将会对嵌入式 Linux 开发具备大概的了解和认识；第 2 章介绍交叉编译工具链的制作，对于非 X86 硬件平台的设备开发通常使用交叉编译工具链在 X86 机器上进行，该章内容是编译目标内核和程序的基础，它包括对交叉工具链、使用分步法构建交叉工具链和使用 Crosstool 构建交叉工具链。通过本章学习，读者将会对交叉工具链有深刻的认识，并可以构建自己的交叉工具链；第 3 章讲述 ARM Linux 的引导程序——BootLoader，这是内核移植的关键，没有良好的 BootLoader 来引导内核工作，再强大、稳定的内核也不能正常工作，它包括对 BootLoader 的介绍、U-boot 的移植与分析以及如何自己设计 BootLoader。通过本章学习，读者将会对 BootLoader 的作用有更清楚的认识，以及学会如何移植和设计 BootLoader；第 4 章讲述嵌入式 Linux 内核移植，也是实际工作中非常重要的内容，它包括移植的基本概念、内核配置、内核编译，以及根文件系统的构建，通过本章学习，读者将会对内核移植以及构建根文件系统有更深入的理解。

总之，读者通过对这部分内容的学习，将会了解构建嵌入式 Linux 系统所需要的一些工作，比如交叉工具链、BootLoader、内核配置、内核编译等。

第 1 章

本章学习目标:

- 了解嵌入式系统基本概念
- 了解嵌入式系统的基本组成
- 了解 ARM 处理器的特点
- 学会使用 ADS 集成开发工具 (Code Warrior 和 AXD)
- 熟悉 Linux 开发环境
- 学会如何有效阅读 Linux 内核代码

1.1 嵌入式系统介绍

俗话说的好“良好的开始是成功的一半。”，虽然这句话并不是真理，但是希望读者在学习任何东西之前都应该有坚定的学习态度和持之以恒的信念，同样学习本书也要有个良好的开端。本章作为 ARM Linux 系统移植的第一章，也是本书的第一章，首先介绍嵌入式系统的概念。

1.1.1 嵌入式系统概述

随着嵌入式系统在消费类电子、工业控制、航空航天、汽车电子、医疗保健、网络通信等各个领域的广泛应用，嵌入式系统这个名词已经被各行各业的人所熟悉，嵌入式系统已经走进了人们的生活，它正在以各种不同的形式改变着人们的生产、生活方式。嵌入式系统，英文为 **Embedded System**，从广义上讲，凡是带有微处理器的专用软、硬件系统都可称为嵌入式系统。如各类单片机和 DSP 系统，这些系统在完成较为单一的专业功能时具有简洁高效的特点。但是由于它们没有使用操作系统，所以管理系统硬件和软件的能力有限，在实现复杂的多任务功能时往往困难重重，甚至无法实现。从狭义上讲，是指那些使用嵌入式微处理器构成的独立系统，并且有自己的操作系统，具有特定功能，用于特定场合的系统。本书中所说的嵌入式系统是指狭义上的嵌入式系统。到目前为止，对于嵌入式系统还没有一个明确的定义。嵌入式系统的核心是嵌入式微处理器，该处理器都是 RISC (Reduce Instruction Set Computing, 精简指令集计算机)^[注 1]的处理器内核。

注 1 RISC 和 CISC (Complex Instruction Set Computing, 复杂指令集计算机) 是当前 CPU 的两种架构。它们的区别在于不同的 CPU 设计理念和方法。早期的 CPU 全部是 CISC 架构，它的设计目的是要用最少的机器语言指令来完成所需的计算任务。比如对于乘法运算，在 CISC 架构的 CPU 上，可能需要这样一条指令：`MUL ADDRA, ADDR B` 就可以将 `ADDRA` 和 `ADDR B` 中的数相乘并将结果储存在 `ADDRA` 中。将 `ADDRA, ADDR B` 中的数据读入寄存器，相乘和将结果²回内存的操作全部依赖于 CPU 中设计的逻辑来实现。这种架构会增加 CPU 结构的复杂性和对 CPU 工艺的要求，但对于编译器的开发十分有利。比如 C 程序中的 `a*b` 就可以直接编译为一条乘法指令。今天只有 Intel 及其兼容 CPU 还在使用 CISC 架构。RISC 架构要求软件来指定各个操作步骤。上面的例子如果要在 RISC 架构上实现，将 `ADDRA, ADDR B` 中的数据读入寄存器，相乘和将结果²回内存的操作都必须由软件来实现，比如：`MOV A, ADDRA; MOV B, ADDR B; MULA, B; STR ADDRA, A`。这种架构可以降低 CPU 的复杂性以及允许在同样的工艺水平下生产出功能更强大的 CPU，但对于编译器的设计有更高的要求。

1. 嵌入式微处理器的主要特点

- 对实时多任务有很强的支持能力，能完成多任务并且有较短的中断响应时间，从而使内部的代码和实时内核的执行时间减少到最低限度。
- 具有功能很强的存储区保护功能。这是由于嵌入式系统的软件结构已模块化，而为了避免在软件模块之间出现错误的交叉作用，需要设计强大的存储区保护功能，同时也有利于软件诊断。
- 可扩展的处理器结构，以便最迅速地开发出满足应用的高性能的嵌入式设备。
- 嵌入式微处理器必须功耗很低，尤其是用于便携式的无线及移动的计算和通信设备中靠电池供电的嵌入式系统更是如此。

2. 嵌入式处理器内核按照体系结构的分类

(1) MIPS 处理器

MIPS 处理器是由美国 MIPS 公司研发出来的一套处理器体系，MIPS 公司是一家设计制造高性能、高档次及嵌入式 32 位和 64 位处理器的厂商，在 RISC 处理器方面占有重要地位。目前 MIPS 公司的战略发生变化，把重点放在嵌入式系统上。1999 年，MIPS 公司发布 MIPS32 和 MIPS64 架构标准，为未来 MIPS 处理器的开发奠定了基础。新的架构集成了所有原来的 MIPS 指令集，并且增加了许多更强大的功能。MIPS 公司陆续开发了高性能、低功耗的 32 位处理器内核（core）MIPS32 4Kc 与高性能 64 位处理器内核 MIPS64 5Kc。2000 年，MIPS 公司发布了针对 MIPS32 4Kc 的版本以及 64 位 MIPS64 20Kc 处理器内核。

(2) ARM 处理器

ARM (Advanced RISC Machines) 处理器是由只设计内核的英国 ARM 公司研发出来的一套处理器体系。ARM 是微处理器行业的一家知名企业，设计了大量高性能、廉价、低能耗的 RISC 处理器，开发了许多相关技术及软件。其技术具有性能高、成本低和能耗省的特点，适用于多种领域，比如嵌入控制、消费/教育类多媒体、DSP 和移动式应用等。ARM 将其技术授权给世界上许多著名的半导体、软件和 OEM 厂商，每个厂商得到的都是一套独一无二的 ARM 相关技术及服务。利用这种合伙关系，ARM 很快成为许多全球性 RISC 标准的缔造者。目前，总共有 30 家半导体公司与 ARM 签订了硬件技术使用许可协议，其中包括 Intel、IBM、三星电子、LG 半导体、NEC、SONY、飞利浦和国民半导体等许多这样的大公司。至于软件系统的合伙人，则包括微软、升阳和 MRI 等一系列知名公司。ARM 架构是面向低预算市场设计的第一款 RISC 微处理器。

(3) PowerPC 处理器

20 世纪 90 年代，IBM (国际商用机器公司)、Apple (苹果公司) 和 Motorola (摩托罗拉) 公司开发 PowerPC 芯片成功，并制造出基于 PowerPC 的多处理器计算机。PowerPC 架构的特点是可伸缩性好、方便灵活。第一代 PowerPC 采用 0.6 微米的生产工艺，晶体管的集成度达到单芯片 300 万个。MPC860 和 MPC8260 是最经典的两款 PowerPC 内核的嵌入式处理器。

(4) 68K/ColdFire 处理器

68K/ColdFire 处理器是 Motorola 公司独有的处理器体系。68K 内核是最早在嵌入式领域广泛应用的内核，其最著名的代表芯片是 68360。ColdFire 继承了 68K 的特点并继续兼容它。

由于嵌入式系统一般具有芯片集成度高、软件代码小、高度自动化、响应速度快等特点，特别适合要求实时性和多任务的体系。RTOS（Real-Time Operating System，实时操作系统）是根据操作系统的工作特性而言的，实时是指物理进程的真实时间。实时操作系统是指具有实时性，能支持实时控制系统工作的操作系统。首要任务是调度一切可利用的资源完成实时控制任务，其次才着眼于提高计算机系统的使用效率，主要特点是要满足对时间的限制和要求。一般 Windows、UNIX、Linux 等桌面系统都属于分时操作系统，在此有必要说明一下实时操作系统与分时操作系统的区别。具体地说，对于分时操作系统，软件的执行在时间上的要求并不严格，时间上的错误，一般不会造成灾难性的后果；而对于实时操作系统，主要任务是对事件进行实时的处理，虽然事件可能在无法预知的时刻到达，但是软件必须在事件发生时能够在严格的时限内作出响应，即使是在尖峰负荷下，也应该如此，系统时间响应的超时就意味着致命的失败。另外，实时操作系统的重要特点是具有系统的可确定性，即系统能对运行情况的好坏和最坏等的情况能做出精确的估计。

1.1.2 嵌入式系统组成

嵌入式系统一般由硬件平台和软件平台两部分组成，如图 1.1 所示。其中硬件平台由嵌入式微处理器和外围硬件设备组成，而软件平台由嵌入式操作系统和应用软件组成。

随着芯片技术的不断发展，嵌入式处理器的主频也越来越高，通常主频都在 40MHz 以上，有的甚至高达 500MHz。多处理器、多核处理器平台也逐渐应用在嵌入式领域，不过现在大量使用的还是 32 位单处理器组成的平台。一个典型的硬件平台如图 1.2 所示。

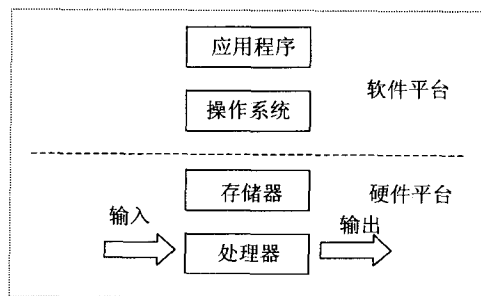


图 1.1 嵌入式系统的一般架构

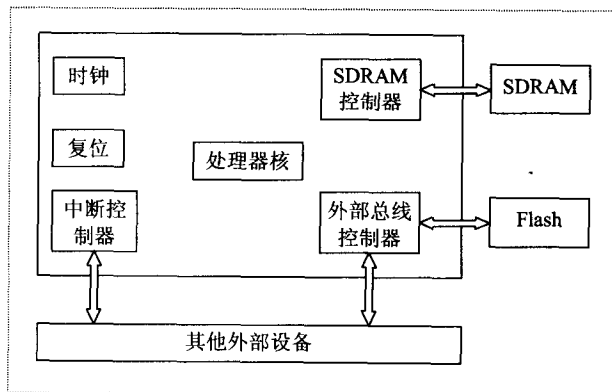


图 1.2 嵌入式硬件平台基本组成结构

嵌入式软件平台主要由嵌入式操作系统与应用软件组成。目前流行的嵌入式操作系统可以分为两类，一类是从运行在个人电脑上的操作系统向下移植到嵌入式系统中，形成的嵌入式操作系统，如微软公司的 Windows CE，SUN 公司的 Java 系统，朗讯科技公司的 Inferno，嵌入式 Linux 等。这类系统经过个人电脑或高性能计算机等产品的长期运行考验，技术日趋成熟，其相关的标准和软件开发方式已被用户普遍接受，同时积累了丰富的开发工具和应用软件资源。另一类是实时操作系统，如 WindRiver 公司的 VxWorks，ISI 的 pSOS，QNX 系统软件公司的 QNX，ATI 的 Nucleus，中国科学院凯思集团的 Hopen 嵌入式操作系统等，这类产品在操作系统的结构和实现上

都针对所面向的应用领域，对实时性高可靠性等进行了精巧的设计，而且提供了独立而完备的系统开发和测试工具，较多地应用在军用产品和工业控制等领域中。目前常见的嵌入式系统有 Linux、uClinux、Windows CE、PalmOS、Symbian、eCos、uCOS-II、VxWorks、pSOS、Nucleus、ThreadX、Rtems、QNX、INTEGRITY、OSE、C Executive 等。嵌入式操作系统的发展也必将带动新一轮的科技竞争。

应用程序运行在嵌入式操作系统之上，一般情况下应用程序和操作系统是分开的。当处理器上带有 MMU (Memory Management Unit, 存储器管理单元) 时，它可以从硬件上将应用程序和操作系统分开编译和管理，Linux、Windows CE 就是这种分离机制。这样做的好处就是系统安全性更高，可维护性更强，更有利于各功能模块的划分。很多情况下在没有 MMU 的处理器时，如 ARM7TDMI，经常应用程序和操作系统是编译在一起运行的，对于开发人员来说，操作系统更像一个函数库。

1.2 ARM 介绍

ARM 是 Advanced RISC Machines (高级精简指令系统处理器) 的缩写，它既是一种微处理器知识产权 (IP) 核，也是一个公司的名称。在上节中对 ARM 公司作了大致介绍，这里不再赘述。

1.2.1 ARM 处理器介绍

ARM 处理器已经成功地广泛应用于无线通信、工业控制、消费类电子产品、网络产品等领域，并且保持持续增长的势头。目前，基于 ARM 技术的微处理器应用约占据了 32 位 RISC 微处理器 75% 以上的市场份额。采用 RISC 架构的 ARM 微处理器一般具有如下特点。

- 体积小、低功耗、低成本、高性能；
- 支持 Thumb (16 位) / ARM (32 位) 双指令集，能很好的兼容 8 位/16 位器件；
- 大量使用寄存器，指令执行速度更快；
- 大多数数据操作都在寄存器中完成；
- 寻址方式灵活简单，执行效率高；
- 指令长度固定。

ARM 微处理器目前包括下面几个系列，每一个系列的 ARM 微处理器都有各自的特点和应用领域。

(1) **ARM7 系列** 一般包括 ARM7TDMI、ARM7TDMI-S、ARM720T、ARM7EJ 几种内核。ARM7TDMI 是目前使用最广泛的 32 位嵌入式 RISC 处理器之一，主要应用于工业控制、Internet 设备、网络和调制解调器设备、移动电话等多种多媒体和嵌入式应用。TDMI 的基本含义如下。

- **T** 支持 16 位压缩指令集 Thumb；
- **D** 支持片上 Debug；
- **M** 内嵌硬件乘法器 (Multiplier)；
- **I** 嵌入式 ICE，支持片上断点和调试点。