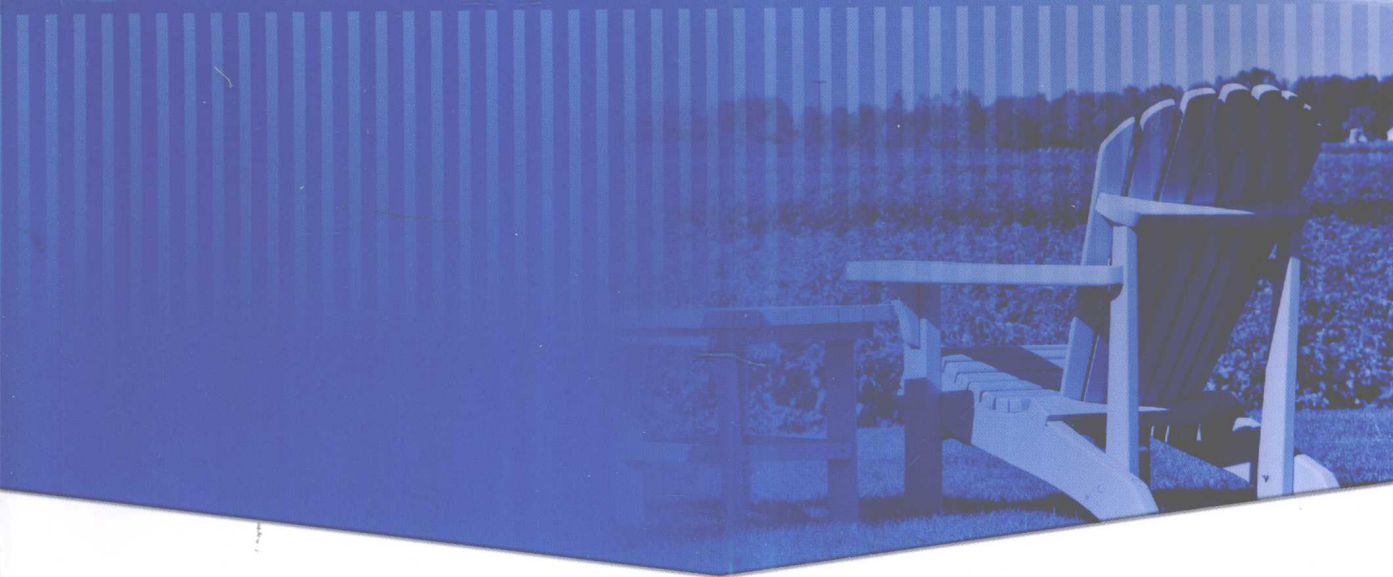




 嵌入式系统系列教材

嵌入式Linux系统 实训教程

 徐成 谭曼琼 徐署华 刘豪 朱雪庆 编著
李仁发 主审



 人民邮电出版社
POSTS & TELECOM PRESS

■ 嵌入式系统系列教材

嵌入式Linux系统 实训教程

■ 徐成 谭曼琼 徐署华 刘豪 朱雪庆 编著
李仁发 主审

人民邮电出版社

北京

图书在版编目 (CIP) 数据

嵌入式Linux系统实训教程 / 徐成等编著. — 北京:
人民邮电出版社, 2010.4
(嵌入式系统系列教材)
ISBN 978-7-115-22377-7

I. ①嵌… II. ①徐… III. ①Linux操作系统—程序设计—教材 IV. ①TP316.89

中国版本图书馆CIP数据核字(2010)第025461号

内 容 提 要

本书首先对嵌入式系统的开发做了简单的概述, 然后介绍 Super-ARM 开发平台及如何在该平台上安装开发环境, 接下来结合开发平台上丰富的资源及接口重点介绍一些典型案例, 包括 LED、键盘、UART、A/D、GPS、GPRS 等。在这些案例中, 深入解析 Linux 操作系统的移植与驱动程序的设计, 同时介绍 QT 嵌入式图形界面工具, 在本书的最后介绍了一个大型的综合型实验。这个实验是在前面的基础上衍生出来的, 融合了已经介绍的技术, 贯穿整个嵌入式开发的流程。通过前面各案例的学习及该大型实验的训练, 读者将具备基本的嵌入式开发能力。

本书适合计算机应用技术、通信工程、软件工程等相关专业的大专(高职)学生阅读, 也可作为社会办学机构中面向企业嵌入式应用开发的定向培养、培训课程的学员的参考教材。

嵌入式系统系列教材

嵌入式 Linux 系统实训教程

-
- ◆ 编 著 徐 成 谭曼琼 徐署华 刘 豪 朱雪庆
主 审 李仁发
责任编辑 王建军
执行编辑 李 静
 - ◆ 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>
北京昌平百善印刷厂印刷
 - ◆ 开本: 787×1092 1/16
印张: 19
字数: 470 千字 2010 年 4 月第 1 版
印数: 1—3 000 册 2010 年 4 月北京第 1 次印刷

ISBN 978-7-115-22377-7

定价: 39.00 元(附光盘)

读者服务热线: (010)67119329 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前 言

背景

随着计算机技术、微处理器技术、电子技术、通信技术、集成电路技术的发展，嵌入式系统已成为计算机技术和应用领域的一个重要组成部分，嵌入式产品在人们的日常生活中也扮演着越来越重要的角色。

近年来，Linux 在嵌入式领域的成功应用引起了人们广泛的兴趣和热烈的回应，越来越多的开发者开始学习和研究嵌入式 Linux 系统。但是嵌入式 Linux 的入门却不简单，进行嵌入式 Linux 的开发需要具备完善的知识结构，不仅要了解嵌入式设备相关的硬件知识，比如常用嵌入式处理器的体系结构、接口技术等，还需要有一定的 Linux 操作系统作为基础。

目前，针对嵌入式 Linux 开发的书籍相当多，也不乏经典。然而大多数书中介绍的实例依赖于平台却又脱离平台，使得其可操作性反而不强；互联网的发展使得共享的网络资源琳琅满目，面对这么多的书籍和资料，刚入门的开发者们往往比较茫然。

本书面向那些想要在未来或者目前的计划中使用 Linux 的有一定经验的嵌入式系统设计者，以及想要熟悉嵌入式 Linux 系统开发工具和技术但没有经验的嵌入式系统开发者，以现在嵌入式领域使用相当广泛的 ARM 处理器为开发平台，在一个具体的 S3C2410 开发板上，从开发平台的介绍、开发环境的搭建、Linux 系统的移植，到最终成功运行一个应用实例，带领读者一步一步掌握嵌入式 Linux 的开发过程。

本书的作者具有多年的嵌入式开发经验，了解嵌入式 Linux 的开发过程，又熟悉 ARM 的体系结构及底层硬件接口电路。特别地，笔者在编写本书之前针对 S3C2410 做过很多的实例开发，解决了不少在开发过程中遇到的问题，也积累了很多经验，因此笔者希望借助本书与大家分享这些经验，目的只是为了让大家在嵌入式 Linux 这条路上少走弯路。

为了让本书更加贴近初学者，笔者根据自己的经验，精心设计了一系列的实验范例，并选定了一个具体的开发平台——深圳旋极力通有限公司的 Super-ARM 实验教学平台。书中所有实验范例的开发都在该平台上完成。之所以选择 Super-ARM 平台，是因为该平台的处理器及其接口电路等都很具有代表性。笔者专门安排了一章来介绍这个平台，只是为了让读者在充分了解平台的基础上更好地理解书中的实例，这样当读者面对其他平台的时候更能够举一反三。另外，本书的所有实验范例都经过多位初学者的再三验证，这足以证明本书实例的可操作性。

预备知识

本书编写依赖 3 个假定：假定读者已经具备一定的 Linux 基础，包括 Linux 的安装、一些基本命令的使用以及在 PC 的 Linux 环境下使用 C 或者 C++ 开发简单的应用程序；假定读者对 ARM 的体系结构有所了解，包括 ARM 处理器的工作原理和接口技术，如 I/O、中断与异常机制等，同时对设备驱动程序的开发以及嵌入式图形界面的开发有一定认识；假定读者有一定硬件电路基础，能看懂一些简单的接口电路，如 LED、数码管等。

本书内容及组织方式

本书分为 4 篇，共 17 章，组织如下。

第一篇：开发环境篇

本篇包括第 1 章到第 4 章，介绍如何搭建嵌入式开发的软硬件环境。

第 1 章：概述

本章主要介绍在嵌入式 Linux 开发过程中会涉及的一些基本概念和基础知识等。

第 2 章：开发平台介绍

本章主要介绍本书实例使用的 Super-ARM 实验教学平台，使读者对开发平台的硬件资源有一定的了解。

第 3 章：ADS 开发环境搭建

本章主要讲解 ADS 开发环境的安装过程，并结合一个简单实例介绍其使用方法，为后续开发中 BootLoader 移植做好准备。

第 4 章：嵌入式 Linux 开发环境搭建

本章介绍主机 Linux 开发环境的安装，包括系统的安装和相关开发工具的配置等。

第二篇：系统移植篇

本篇包括第 5 章到第 7 章，介绍 Linux 系统的移植过程。

第 5 章：BootLoader 移植

本章主要介绍 BootLoader 的一些基础知识以及 U-Boot 的详细移植过程。

第 6 章：内核移植

本章主要介绍 Linux 内核的编译、配置及安装过程。

第 7 章：文件系统移植

本章主要讲解根文件系统的制作、移植和访问以及 NFS 文件系统的挂载。

第三篇：基础实验篇

本篇包括第 8 章到第 14 章，主要介绍一些简单硬件的驱动程序的编写、移植及测试。

第 8 章：Linux 设备驱动概述

本章主要介绍 Linux 设备驱动的相关概念，包括设备驱动的作用、分类及特点，以及驱动模块的加载与卸载等。

第 9 章：LED 实例

本章结合 LED 驱动实例介绍简单字符设备驱动的程序结构，并编写简单的应用程序对设备驱动进行测试，通过本章的学习，读者可以了解开发驱动程序的大致流程，为以后学习其他设备驱动程序的开发打下基础。

第 10 章：按键中断实例

本章介绍 S3C2410 的中断机制及嵌入式 Linux 中断驱动程序的结构。通过按键驱动实例的开发，带领大家一步一步实现该驱动程序的设计与测试过程。

第 11 章：数码管实例

本章主要介绍数码管驱动的开发过程。通过本章的学习，读者可以了解数码管的显示原理及其驱动程序的设计方法。

第 12 章：4×4 矩阵键盘实例

本章主要介绍键盘驱动的开发及测试。通过本章的学习，读者可以了解键盘扫描的原理，

从而为自己的系统添加键盘设备驱动程序。

第 13 章：LCD 驱动实例

本章主要介绍 LCD 显示设备的工作原理以及在 Linux 中移植 LCD 设备驱动的过程。

第 14 章：触摸屏实例

本章主要介绍触摸屏的工作原理，以及在 Linux 中移植触摸屏输入设备驱动的过程。

第四篇：图形应用篇

本篇包括第 15 章到第 17 章，主要介绍 QT/Embedded 的安装和编译，并通过开发一个简单的 QT 实例，使学生逐步了解嵌入式图形界面编程，最后完成本书的一个综合实例。

第 15 章：嵌入式 QT 环境搭建

本章首先对嵌入式 GUI 编程做个简单介绍，然后在宿主机上完成 QT/Embedded 开发环境的搭建。

第 16 章：嵌入式 QT 之 HelloWorld

本章指导学生编写一个最简单的 QT 程序：在一个主窗口中添加一个按钮，点击该按钮会弹出一个对话框，让该程序分别运行在桌面 PC 和实验箱上。

第 17 章：综合实例

作为本书的最后一章，本章结合基础实验篇中的一些实例，用 QT/Embedded 编写一个简单图片浏览程序。该程序具有如下功能：

- (1) 能读取文件系统系统所有目录下的图片，在实验箱的 LCD 屏幕上显示出来。
- (2) 具有单张查看和循环播放两种模式。
- (3) 在循环播放模式下，可以通过实验箱的数码管显示播放时间间隔（以 s 为单位）。
- (4) 在单张查看模式下可以通过点击触摸屏切换到下一张或上一张。

附件

为方便读者高效地理解并完成本书的实例，笔者为部分章节提供了相关的源代码、工程文件以及参考资料等，以附件的形式发布在随书光盘中。

作者

本书第一篇由徐成、徐署华、刘豪、谭曼琼和朱雪庆编写，第二篇由徐成、谭曼琼和朱雪庆编写，第三篇由谭曼琼编写，第四篇由谭曼琼、朱雪庆和邱金波编写。全书由谭曼琼整理和统稿，由李仁发教授主审。

徐成教授多年从事嵌入式系统教学，有丰富的嵌入式系统开发和教学经验，徐署华、刘豪、谭曼琼等也从事嵌入式开发和教学多年，本书是所有作者教学与开发经验的积累和总结。

本书的所有内容和实例都经过作者的仔细编排与验证。这里要特别感谢刘冬、张嵎桐、张泰忠、曾娟丽等人的无私帮助，他们本着认真负责的态度对本书的内容，特别是实例部分进行了验证，并提出了宝贵的意见。我们尽管再三校对，但书中肯定还存在错误和不足，恳请读者批评指正。您宝贵的意见和建议是我们的期待，也是我们继续前进的动力。我们的联系方式：xu_cheng@yeah.net, heima251210490@163.com。

目 录

第一篇 开发环境篇

第 1 章 概述	2
1.1 基本概念	2
1.1.1 嵌入式 Linux	2
1.1.2 ARM 处理器	3
1.1.3 宿主机和目标机	5
1.1.4 交叉编译	7
1.1.5 内核	9
1.1.6 根文件系统	9
1.1.7 系统引导程序 (BootLoader)	10
1.2 嵌入式 Linux 开发过程简介	11
第 2 章 开发平台介绍	13
2.1 Super-ARM 整体结构及资源	13
2.1.1 Super-ARM 整体结构	13
2.1.2 Super-ARM 教学系统主要资源	14
2.2 Super-ARM 各模块简介	15
2.2.1 底板	15
2.2.2 主板	17
2.2.3 核心板	19
2.2.4 LCD 模块	19
2.2.5 JTAG 仿真器 ProbeICE	20
2.2.6 连接电缆及配件	20
第 3 章 ADS 开发环境搭建	21
3.1 搭建硬件开发环境	21
3.1.1 Super-ARM 供电线路连接	21
3.1.2 JTAG 连接	22
3.2 搭建软件开发环境	22
3.2.1 ProbeICE-ARM 服务器配置	23
3.2.2 ProbeICE-ARM 仿真器配置	24
3.3 新建工程及仿真调试	26
3.3.1 新建工程	26
3.3.2 编译及连接	30

3.3.3 仿真调试	30
第 4 章 嵌入式 Linux 开发环境搭建	33
4.1 安装 Linux 操作系统	33
4.1.1 在 Windows 中安装虚拟机	33
4.1.2 在虚拟机上安装 Ubuntu	37
4.2 宿主机 Linux 系统中基本环境的配置与安装	39
4.2.1 网络配置	39
4.2.2 更新 Linux 操作系统	40
4.2.3 安装一些相关工具和程序库	40
4.2.4 实际项目工作空间目录的安排	41
4.2.5 配置 Linux 下的 Minicom	42
4.2.6 配置 Linux 下的 TFTP 服务器	45
4.3 在主机 Linux 系统中建立交叉编译环境	47
4.3.1 准备工具链软件	47
4.3.2 建立交叉编译工具链	48

第二篇 系统移植篇

第 5 章 BootLoader 移植	52
5.1 系统引导程序基础	52
5.1.1 BootLoader 的安装媒介 (Installation Medium)	52
5.1.2 BootLoader 工作模式	53
5.1.3 BootLoader 的启动过程	53
5.1.4 BootLoader 与宿主机之间进行文件传输所用的通信设备及协议	55
5.1.5 几种常用的 BootLoader	55
5.2 移植 U-Boot	58
5.2.1 U-Boot 源码结构	58
5.2.2 U-Boot 移植过程详解	59
5.2.3 烧写 U-Boot	68
5.3 U-Boot 使用	69
5.3.1 U-Boot 命令	69
5.3.2 U-Boot 命令使用实例	71
5.3.3 U-Boot 引导内核	74
5.4 U-Boot 的环境变量	74
5.4.1 查看环境变量	74
5.4.2 设置环境变量	75
5.4.3 保存环境变量	76
第 6 章 Linux 内核移植	77
6.1 选择 Linux 内核	77
6.2 Linux 内核移植前准备	79

6.2.1 Linux 内核源码	79
6.2.2 内核配置及方法	81
6.2.3 图形化内核配置选项	81
6.2.4 构建内核映像	83
6.2.5 构建模块	84
6.2.6 安装内核及内核模块	84
6.3 Linux 内核移植实例	84
6.3.1 配置编译内核的过程	84
6.3.2 从 NandFlash 引导内核	87
6.3.3 网络引导内核	88
第 7 章 根文件系统移植	90
7.1 文件系统概述	90
7.1.1 文件系统	90
7.1.2 根文件系统	92
7.2 制作根文件系统	93
7.2.1 BusyBox 工具	93
7.2.2 根文件系统制作实例	94
7.3 配置 NFS 服务	101
7.3.1 NFS 简介	101
7.3.2 在 Linux 下配置 NFS 服务器	103
7.4 目标机挂载 NFS 根文件系统	106
7.4.1 移植网卡驱动	106
7.4.2 挂载 NFS 根文件系统	108
7.5 Hello World 测试程序	113
7.6 文件系统移植	114
7.6.1 配置内核命令行参数	114
7.6.2 制作 cramfs 文件系统	115
7.6.3 将根文件系统烧写到 NandFlash	115

第三篇 基础实验篇

第 8 章 Linux 设备驱动概述	118
8.1 设备驱动的角色	118
8.2 设备驱动和操作系统	119
8.2.1 无操作系统时的设备驱动	119
8.2.2 有操作系统时的设备驱动	121
8.3 Linux 设备驱动	122
8.3.1 Linux 设备的分类及特点	122
8.3.2 不同设备的驱动设计概述	124
8.3.3 Linux 设备文件的创建	127

8.3.4	Linux 驱动程序的加载和卸载	128
8.3.5	学习 Linux 驱动程序的基础及方法	129
第 9 章	LED 实例	130
9.1	字符设备驱动程序的结构	130
9.1.1	常用的头文件	130
9.1.2	主次设备号	131
9.1.3	cdev 结构体	132
9.1.4	分配和释放设备号	133
9.1.5	File_operation 结构体	134
9.1.6	字符设备驱动程序一般结构	135
9.2	LED 设备驱动实例	137
9.2.1	实验目的	137
9.2.2	实验原理	137
9.2.3	实验任务	139
9.2.4	实验步骤	139
第 10 章	按键中断实例	147
10.1	S3C2410 中断机制	147
10.1.1	中断向量表	147
10.1.2	中断控制器	148
10.2	嵌入式 Linux 中断处理程序架构	154
10.2.1	Linux 中断处理程序的上半部与下半部机制	154
10.2.2	Linux 中断编程	155
10.3	按键中断驱动实例	158
10.3.1	实验目的	158
10.3.2	实验原理	158
10.3.3	实验任务	158
10.3.4	实验步骤	159
第 11 章	数码管实例	167
11.1	数码管显示原理	167
11.1.1	数码管简介	167
11.1.2	数码管显示原理	168
11.1.3	数码管驱动电路设计	169
11.2	数码管设备驱动实例	171
11.2.1	实验目的	171
11.2.2	实验原理	171
11.2.3	实验任务	172
11.2.4	实验步骤	172
第 12 章	4×4 矩阵键盘实例	183
12.1	键盘接口概述	183
12.1.1	键盘的分类	183

12.1.2	键盘的扫描	184
12.1.3	键盘的防抖	185
12.1.4	键盘的缓冲算法	186
12.2	4×4 键盘驱动实例	186
12.2.1	实验目的	187
12.2.2	实验原理	187
12.2.3	实验任务	188
12.2.4	实验步骤	189
第 13 章	LCD 驱动实例	204
13.1	LCD 的硬件原理	204
13.1.1	LCD 的显像原理	204
13.1.2	LCD 的分类	205
13.1.3	TFT LCD 的工作时序及参数设置	206
13.2	S3C2410 的 LCD 控制器及 IO 配置	209
13.2.1	LCD 控制器	209
13.2.2	LCD 控制器的寄存器设置	209
13.2.3	S3C2410 的 LCD 接口及配置	212
13.3	基于帧缓冲 (FrameBuffer) 的 LCD 驱动分析	213
13.3.1	FrameBuffer 设备驱动分析	214
13.3.2	基于 FrameBuffer 的 LCD 驱动分析	218
13.3.3	FrameBuffer 驱动与 LCD 驱动之间的关系	226
13.4	LCD 驱动移植实例	226
13.4.1	实验目的	226
13.4.2	实验原理	226
13.4.3	实验任务	227
13.4.4	实验步骤	227
第 14 章	触摸屏实例	230
14.1	触摸屏的硬件原理	230
14.2	S3C2410 的触摸屏控制	231
14.2.1	S3C2410 的 ADC 及触摸屏接口原理	231
14.2.2	S3C2410 的 ADC 和触摸屏接口模式	233
14.2.3	S3C2410 的 ADC 及触摸屏控制寄存器	234
14.2.4	S3C2410 的触摸屏中断控制	236
14.3	Linux 输入子系统	236
14.3.1	Input 子系统架构	236
14.3.2	Input 子系统数据结构	237
14.3.3	Input 设备驱动编写	237
14.3.4	Input 设备驱动范例	238
14.4	触摸屏驱动移植实例	239
14.4.1	实验目的	239

14.4.2 实验原理	239
14.4.3 实验任务	246
14.4.4 实验步骤	246

第四篇 图形应用篇

第 15 章 嵌入式 QT 环境搭建	250
15.1 嵌入式 GUI 简介	250
15.1.1 MiniGUI	251
15.1.2 OpenGUI	252
15.1.3 QT/Embedded	252
15.2 QT 开发环境搭建	254
15.2.1 Ubuntu 下安装 QT 开发环境	254
15.2.2 QT Assistant	255
15.2.3 QT Designer	256
15.2.4 QDevelop	258
15.3 QT/Embedded 环境的安装	259
15.3.1 安装 tslib	259
15.3.2 安装 QT/Embedded	260
15.3.3 环境配置及触摸屏校准	262
第 16 章 嵌入式 QT 之 HelloWorld	265
16.1 QT/E 信号与插槽机制	265
16.1.1 信号和插槽机制	265
16.1.2 信号和插槽机制的实现	266
16.2 HelloWorld 实例	269
16.2.1 实验目的	269
16.2.2 实验任务	269
16.2.3 实验步骤	270
第 17 章 综合实例	276
17.1 实验目的	276
17.2 实验任务	276
17.3 实验步骤	277
参考文献	291



第一篇 开发环境篇

本篇内容

概述

开发平台介绍

ADS 开发环境搭建

嵌入式 Linux 开发环境搭建

本篇目标

了解嵌入式 Linux 开发过程中会涉及的一些基本概念

了解本书实例所使用的硬件开发平台

熟悉 ADS 开发环境的安装过程及使用方法

掌握主机 Linux 开发环境中虚拟机的安装方法及相关开发工具详细配置

过程

本篇实例

实例 1: ADS 相关开发工具的安装及使用

实例 2: 虚拟机上 Linux 工作站的安装及配置

实例 3: Minicom 及 TFTP 的安装配置

实例 4: 交叉编译工具链的制作

第 1 章 概 述

本书的目的就是带领你完成一个任务——在一块空的嵌入式开发板上移植一个基于 Linux 的应用程序实例。在进行我们的嵌入式 Linux (Embedded Linux) 开发过程之前,也许你心中正在问:“嵌入式 Linux”,这究竟是一个怎样的世界?从这一刻开始,让我们一起步入这个世界吧!

当你已经拿到本书的时候,不管你是未入门的还是刚入门的,我想你一定是对“嵌入式 Linux”感兴趣的,所以假定你已经知道什么是嵌入式系统了,套用现在流行的定义“嵌入式系统是以应用为中心,以计算机技术为基础,软硬件可裁剪,对功能、可靠性、成本、功耗、体积等有严格要求的专用计算机系统”,而且我们几乎天天在跟嵌入式设备打交道,别不承认,你有几天离开过你的手机?你一个人听歌或者听英语的时候是不是要 MP3 呢?还有,公交车上和银行里几乎都存在嵌入式设备。同样,我也认为你已经知道 Linux 是什么了,甚至还在自己的 PC 上安装过 Linux,进行过 Linux 的开发。在此,就不多费口舌介绍“什么是 Linux”了。至于“什么是嵌入式 Linux”,恐怕很多读者都想得到答案,笔者在接下来的基本概念中将做简单介绍。

本章接下来向大家介绍嵌入式 Linux 开发中的一些基本概念,以及嵌入式 Linux 的开发过程,让读者对在开发过程中遇到的一些术语不至于感到陌生,同时也对嵌入式 Linux 的开发有个大致的了解。当然,你读完这一章之后对于某些概念,可能还是不甚理解,不妨先把疑问留下,相信随着开发过程的深入,你一定会得到答案的,到时候再回过头来阅读这些概念,便会明白了。

1.1 基本概念

1.1.1 嵌入式 Linux

嵌入式 Linux 是指对标准 Linux 经过小型化裁剪处理之后,能够固化在容量只有几千字节或者几兆字节的存储器芯片(如 Flash)或者单片机中,是适合于特定嵌入式应用场合的专用 Linux 操作系统。

嵌入式 Linux 同 Linux 一样,具有低成本、高性能、支持多种硬件平台和良好的网络支持等优点。另外,嵌入式 Linux 为了更好地适应嵌入式领域的开发,还在 Linux 基础上做了不少改进,例举部分如下。

(1) 改善的内核结构

整个 Linux 内核是一个单独的、非常大的程序,这种整体式结构虽然能够使系统的各个部分直接沟通,提高了系统响应速度,但与嵌入式系统存储容量小、资源有限的点不符合。因此,嵌入式系统中经常采用的是一种称为微内核的体系结构,即内核本身只提供一

本的操作系统功能，如任务调度、内存管理、中断处理等，而类似于文件系统和网络协议等附加功能则运行在用户空间中，并且可以根据实际需要进行裁剪。这样就大大减小了内核的体积，便于维护和移植。

(2) 提高的系统实时性

现有的 Linux 是一个通用的操作系统，虽然它也采用了许多技术来加快系统的运行和响应速度，但从本质上来说并不是一个嵌入式实时操作系统。因此，利用 Linux 作为底层操作系统，在其上进行实时化改造，从而构建出一个具有实时处理能力的嵌入式系统，如 RT-Linux 已经成功地应用于航天飞机的空间数据采集、科学仪器测控和电影特技图像处理等各领域。

嵌入式 Linux 同 Linux 一样，也有众多的版本（见表 1-1）。

表 1-1 嵌入式 Linux 的主要版本

版 本	简 单 介 绍
uCLinux	开放源码的嵌入式 Linux 的典范之作。它主要是针对目标处理器没有存储管理单元（MMU），其运行稳定，具有良好的移植性和优秀的网络功能，对各种文件系统有完备的支持，并提供标准丰富的 API
RT-Linux	由美国墨西哥理工学院开发的嵌入式 Linux 硬实时操作系统。它已有广泛的应用
Embedix	根据嵌入式应用系统的特点重新设计的 Linux 发行版本。它提供了超过 25 种的 Linux 系统服务，包括 Web 服务器等。此外还推出了 Embedix 的开发调试工具包、基于图形界面的浏览器等。可以说，Embedix 是一种完整的嵌入式 Linux 解决方案
XLinux	采用了“超字元集”专利技术，使 Linux 内核不仅能与标准字符集相容，还涵盖了 12 个国家和地区的字符集。因此，XLinux 在推广 Linux 的国际应用方面有独特的优势
PocketLinux	可以提供跨操作系统并且构造统一的、标准化的和开放的信息通信基础结构，在此结构上实现端到端方案的完整平台
红旗嵌入式 Linux	由北京中科院红旗软件公司推出的嵌入式 Linux，它是国内做得较好的一款嵌入式操作系统。目前，中科院计算机研究所自行开发的开放源码的嵌入式操作系统——Easy Embedded OS（EEOS）也已经开始进入实用阶段

表 1-1 所列出的嵌入式 Linux 的不同版本分别是针对不同的需要在内核等方面加入了特定的机制而定制的，在很多嵌入式 Linux 应用中，开发者一般在标准发行版 Linux 内核的基础上直接为自己的硬件平台定制裁剪嵌入式 Linux 系统。

正如上面所说，在“嵌入式开发”的语境中，嵌入式 Linux 通常指一个针对特定硬件设备的完整的系统；而在“嵌入式 Linux 厂商”的语境中，嵌入式 Linux 用于表示以嵌入式系统为需求对象的发行套件，如“嵌入式 Linux 发行套件”指的是为嵌入式系统与开发工具量身定制的软件套件，以便建立一个完整的系统。嵌入式 Linux 发行套件所提供的开发工具可能包括交叉编译器、调试器、引导映像生成器等（你可能对这些概念感到陌生，笔者将在开发中陆续介绍）。

1.1.2 ARM 处理器

(1) ARM 简介

ARM（Advanced RISC Machines），既可以被认为是一个公司的名字，也可以被认为是对一类微处理器的通称，还可以被认为是一种技术的名字。

ARM 公司主要出售芯片设计技术的授权。目前，采用 ARM 技术知识产权（IP）核的微处理器，即通常所说的 ARM 处理器，已遍及工业控制、消费类电子产品、通信系统、网络系统和无线系统等各类产品市场，基于 ARM 技术的微处理器应用占据 32 位 RISC 微处理器

75%以上的市场份额。

ARM 公司是专门从事基于 RISC 技术芯片设计开发的公司，作为知识产权供应商，它本身不直接从事芯片生产，靠转让设计许可，由合作公司生产各具特色的芯片，世界各大半导体生产商从 ARM 公司购买其设计的 ARM 处理器核，根据各自不同的应用领域，加入适当的外围电路，从而形成自己的 ARM 处理器芯片进入市场。目前，全世界有几十家大半导体公司都使用 ARM 公司的授权，因此既使得 ARM 技术获得更多的第三方工具、制造、软件的支持，又使整个系统成本降低，从而使产品更容易进入市场被消费者所接受，更具有竞争力。

(2) ARM 处理器的应用领域及特点

到目前为止，ARM 处理器及技术的应用几乎已经深入到各个领域，并会在将来取得更加广泛的应用。

➤ 工业控制领域：作为 32 位的 RISC 架构，基于 ARM 核的微控制器芯片不但占据了高端微控制器市场的大部分份额，同时也逐渐向低端微控制器应用领域扩展，ARM 微控制器的低功耗、高性价比，已向传统的 8 位/16 位微控制器提出了挑战。

➤ 无线通信领域：目前已有 85% 以上的无线通信设备采用了 ARM 技术，ARM 以其高性能和低成本的特点，地位日益巩固。

➤ 网络应用：随着宽带技术的推广，采用 ARM 技术的 ADSL 芯片正逐步获得竞争优势。此外，ARM 在语音及视频处理上进行了优化，并获得广泛支持，也对 DSP 的应用领域提出了挑战。

➤ 消费类电子产品：ARM 技术在目前流行的数字音频播放器、数字机顶盒和游戏机中得到广泛应用。

ARM 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，能很好地兼容 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。

(3) ARM 处理器系列

下面所列的是 ARM 处理器的几个系列，以及其他厂商基于 ARM 体系结构的处理器，这些处理器除了具有 ARM 体系结构的共同特点以外，每一个系列的 ARM 处理器都有各自的特点和应用领域。

- ARM7 系列；
- ARM9 系列；
- ARM9E 系列；
- ARM10E 系列；
- SecurCore 系列；
- Intel 的 Xscale；
- Intel 的 StrongARM。

其中，ARM7、ARM9、ARM9E 和 ARM10E 为 4 个通用处理器系列，每一个系列提供一套相对独特的性能来满足不同应用领域的需求。SecurCore 系列专门为安全要求较高的应

用而设计。

下面详细介绍 ARM7 处理器及 ARM9 处理器的特点。

① ARM7 处理器

ARM7 系列微处理器为低功耗的 32 位 RISC 处理器，最适合用于对价位和功耗要求较高的消费类应用。ARM7 微处理器系列具有如下特点。

- 具有嵌入式 ICE—RT 逻辑，调试开发方便。
- 极低的功耗，适合对功耗要求较高的应用，如便携式产品。
- 能够提供 0.9MIPS/MHz 的三级流水线结构。
- 代码密度高并兼容 16 位的 Thumb 指令集。
- 对操作系统的支持广泛，包括 Windows CE、Linux、Palm OS 等。
- 指令系统与 ARM9 系列、ARM9E 系列和 ARM10E 系列兼容，便于用户的产品升级换代。

- 主频最高可达 130MIPS，高速的运算处理能力能胜任绝大多数的复杂应用。

ARM7 系列微处理器包括如下几种类型的核：ARM7TDMI、ARM7TDMI-S、ARM720T、ARM7EJ。其中，ARM7TMDI 是目前使用最广泛的 32 位嵌入式 RISC 处理器，属低端 ARM 处理器核。TDMI 的基本含义：

T 为支持 16 位压缩指令集 Thumb；

D 为支持片上 Debug；

M 为内嵌硬件乘法器 (Multiplier)；

I 为嵌入式 ICE，支持片上断点和调试点。

② ARM9 处理器

ARM9 系列微处理器在高性能和低功耗特性方面提供最佳的性能，具有以下特点。

- 5 级整数流水线，指令执行效率更高。
- 提供 1.1MIPS/MHz 的哈佛结构。
- 支持 32 位 ARM 指令集和 16 位 Thumb 指令集。
- 支持 32 位的高速 AMBA 总线接口。
- 全性能的 MMU，支持 Windows CE、Linux、Palm OS 等多种主流嵌入式操作系统。
- MPU 支持实时操作系统。
- 支持数据 Cache 和指令 Cache，具有更高的指令和数据处理能力。

ARM9 系列微处理器包含 ARM920T、ARM922T 和 ARM940T 这 3 种类型，以适用于不同的应用场合。本书的开发平台上使用的处理器就是 ARM920T。

1.1.3 宿主机和目标机

宿主机其实就是我们的开发主机，大多数时候指的就是日常使用的 PC，我们在宿主主机上进行应用程序的开发；而目标机就是我们的开发板，也叫做目标板，它为应用程序提供运行环境。

在开发过程中，宿主机与目标机必须进行信息交互，如在调试过程中，宿主机要向目标机发送调试控制命令，目标机则需要向宿主机返回调试状态信息，开发完成后，宿主机需要将代码下载到目标机上。为了实现信息交互，宿主机与目标机之间必须存在物理的连接，连接方式按交互方式的不同而改变，主要包括 JTAG 连接、串口连接、网络连接、USB 连接。