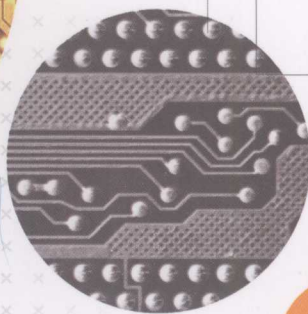
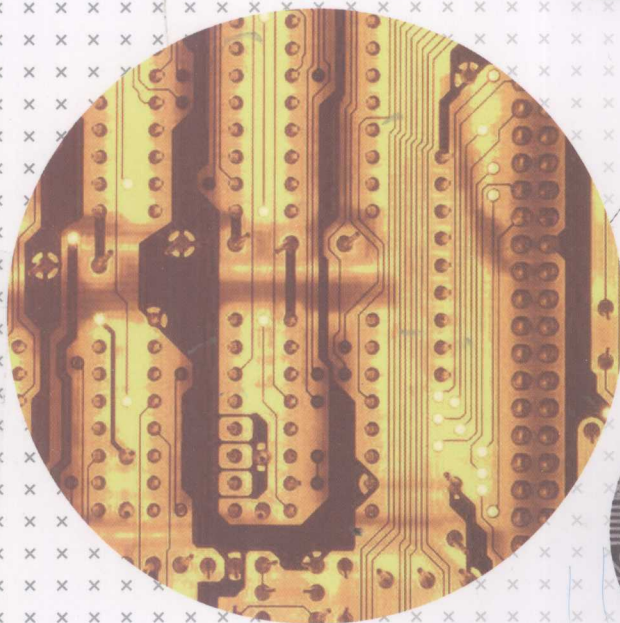


嵌入式Linux

驱动程序和系统开发

实例精讲

罗苑棠 编著



实例丰富
即学即用



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

电子工程应用
精讲系列

嵌入式Linux 驱动程序和系统开发 实例精讲

罗苑棠 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是《嵌入式 Linux 应用系统开发实例精讲》的改版。全书通过大量实例精讲的形式,详细介绍了嵌入式 Linux 驱动程序与系统开发的方法与流程。全书分 3 篇共 19 章,第 1 篇为基础知识篇,介绍了 Linux 的移植、开发环境平台、Linux 程序设计基础及常用开发工具,引导读者技术入门。第 2 篇为 Linux 驱动程序开发与实例篇,结合 6 个实际案例阐述了网卡驱动、声卡驱动、显卡驱动、USB 驱动、闪存 Flash 驱动的开发原理技术和应用。第 3 篇为 Linux 系统开发实例篇,安排了 8 个实际应用系统实例,涵盖工业设备、视频处理、指纹识别、网络传输通信、摄像监控、移动校园系统等嵌入式热门领域,实战和商业价值高,利于读者举一反三,快速掌握 Linux 系统设计的流程,提高实际设计能力。

本书配有光盘 1 张,包含了全书所有实例的硬件原理图和程序源代码,方便读者学习和使用。本书适合计算机、自动化、电子及通信等相关专业的大学生,以及从事 Linux 开发的科研人员使用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

嵌入式 Linux 驱动程序和系统开发实例精讲 / 罗苑棠编著. —北京:电子工业出版社, 2009.1
(电子工程应用精讲系列)
ISBN 978-7-121-07936-8

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

中国版本图书馆 CIP 数据核字(2008)第 188979 号

责任编辑:王鹤扬

印 刷:北京市通州大中印刷厂

装 订:三河市鹏成印业有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:30.75 字数:746 千字

印 次:2009 年 1 月第 1 次印刷

印 数:4000 册 定价:59.00 元(含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

丛书说明

工程技术的电子化、集成化和系统化促进了电子工程技术的发展，同时也促进了电子工程技术在社会各行业中的广泛应用，从近年的人才招聘市场来看，电子工程师的人才需求更是一路走高。

电子工程师如此紧俏，除需求不断走高，人才供不应求外，另一重要原因则是电子工程师的门槛相对而言比较高，这个高门槛则来自于工程师的“经验”和“实践”！

因此，为了满足读者学习和工作需要，解决各种工作中的专业问题，我们紧紧围绕“经验”和“实践”，精心策划组织了此套丛书。

1. 丛书范围

现代电子科学技术的一个特点是多学科交叉，因此，工程师应当了解、掌握两门以上的相关学科，知识既精深又广博是优秀的工程师成长为某领域专家的重要标志。本丛书内容涉及软件开发、研发电子及嵌入式项目开发等，包括单片机、USB 接口、ARM、CPLD/FPGA、DSP 和移动通信系统等。

2. 读者对象

本套书面向各领域的初、中级用户，具体为高校计算机、电子信息、通信工程、自动化控制专业在校大学生，以及从事电子开发和应用行业的科研人员。

3. 内容组织形式

本套书紧紧围绕“经验”和“实践”，首先介绍一些相关的基础知识，然后根据不同

的模块或应用领域，分篇安排应用程序实例的精讲。基础知识用来为一些初级读者打下一定的知识功底；基础好一点的读者则可以跳过这一部分，直接进入实例的学习。

4. 实例特色

在应用实例的安排上，着重突出“应用”和“实用”两个基本原则，安排具有代表性、技术领先性，以及应用广泛的典型实例，让读者学习借鉴。这些实例是从作者多年程序开发项目中挑选出来的，也是经验的归纳与总结。

在应用实例的讲解上，既介绍了设计原理、基本步骤和流程，也穿插了一些经验、技巧与注意事项。特别在程序设计思路上，在决定项目开发的质量和成功与否的细节上，尽可能地用简洁的语言来清晰阐述大众易于理解的概念和思想；同时，程序代码部分做了很详细的中文注释，有利于读者举一反三，快速应用和提高。

5. 光盘内容

本套书的光盘中包含了丰富的实例原图文件和程序源代码，读者稍加修改便可应用于自己的工作中或者完成自己的课题（毕业设计），物超所值。读者使用之前，最好先将光盘内容全部复制到电脑硬盘中，以便于以后可以直接调用，而不需要反复使用光盘，提高操作速度和学习效率。

6. 学习指南

对于有一定基础的读者，建议直接从实例部分入手，边看边上机练习，这样印象会比较深，效果更好。基础差一点的读者请先详细学习书中基础部分的理论知识，然后再进行应用实例的学习。在学习中，尽量做到反复理解和演练，以达到融会贯通、举一反三的功效；特别希望尽量和自己的工作设计联系起来，以达到“即学即会，学以致用”的最大化境界。

本套丛书主要偏重于实用性，具有很强的工程实践指导性。期望读者在学习中顺利、如意！

前言

Linux 是一套免费使用和自由传播的类 UNIX 操作系统，这个系统是由世界各地成千上万的程序员设计和实现的。它以高效性和灵活性著称，并且能够在 PC 上实现全部的 UNIX 特性，具有多任务、多用户的能力。Linux 现在受到了广大计算机爱好者的喜爱，原因主要有两个：一是 Linux 属于自由软件，用户不用支付任何费用就可以获得它及其源代码，并且可以根据自己的需要进行必要的修改；另一个原因是它具有 UNIX 的全部功能。

随着 Linux 在我国政府、金融、电信、消费电子等行业的广泛应用，企业对 Linux 人才的需求也开始持续升温。目前 IT 业内许多著名大企业都有急剧扩招 Linux 人才的倾向。巨大的人才需求将使更多的人参与到 Linux 学习的行列中来。

Linux 应用领域比较多，比较常用的有服务器配置与应用、驱动设备开发、嵌入式系统开发等。目前市场上虽存在一些 Linux 驱动程序与嵌入式系统设计图书，但大多以介绍基础理论为主，缺乏商业应用案例的实践指导。本书就是为了弥补这种不足而精心组织编写的。

本书内容

全书以理论为辅、实践为主，重点以典型实例的形式，详细介绍嵌入式 Linux 驱动程序与系统开发的思路、方法与实际应用案例。全书分 3 篇共 19 章，具体内容如下：

第 1~4 章为基础知识篇，主要讲述了嵌入式基础入门、Linux 环境开发平台、C 程序设计基础、Linux 常用开发工具。通过本部分学习，初级读者可以具备一定的 Linux 程序设计功底；基础好一点的读者则可以跳过这一部分。

第 5~10 章为 Linux 驱动程序开发与实例篇，结合 6 个实际案例阐述了网卡驱动、声卡驱动、显卡驱动、USB 驱动、闪存 Flash 驱动的开发原理技术和应用。

第 11~19 篇为 Linux 系统开发实例篇，安排了 8 个实际应用系统实例，涵盖工业设备、视频处理、指纹识别、网络传输通信、摄像监控、移动校园系统等领域，这些实例具有代表性、技术领先性、应用广泛性及热门性的特点，全部调试通过并进入商品化，是作者多

年开发经验的归纳与总结。

本书特色

与同类型书相比, 本书主要具备以下一些特色。

(1) 整体讲解思路: 首先简要讲述了 Linux 嵌入式系统开发的环境平台、程序基础和常用开发工具, 然后是驱动程序与系统开发典型实例的介绍, 并穿插了一些经验、技巧与注意事项, 符合读者循序渐进的学习过程。

(2) 包括 GUI、QT 图形工具、驱动程序开发及系统实例的介绍, 使本书在内容上更加完美、全面。

(3) 本书 6 个驱动程序实例和 8 个应用系统实例, 全部典型实用, 涉及 Linux 开发的诸多热门与核心技术, 工程实战价值高。

(4) 本书不但提供了程序设计的详细思路与流程, 而且对实例的程序代码做了详细注释, 利于读者理解和巩固知识点, 学会举一反三。

(5) 光盘包含了丰富的实例硬件电路图文件和程序源代码, 读者稍加修改, 便可应用于自己的工作中或者完成自己的课题设计, 物超所值。

- **光盘的内容说明** 该光盘为实例素材文件, 按照章节序号来组织, 每章包括电路图、程序代码两部分内容。其中, “电路图”文件夹中的内容为各章的电路图, 多用 Protel 软件制作。

- **光盘的使用说明** 光盘中的程序需要采用 C 语言的编译软件打开阅读, 也可以使用 “UltraEdit” 等软件打开阅读或者编辑。

- **系统要求** 该光盘运行只需一般的 PC 就可以。系统配置推荐为 256MB 以上内存, 1280×1024 分辨率, 32MB 以上显存。

本书读者对象为计算机、电子信息及相关专业的在校大学生, 还有从事 Linux 嵌入式开发的初、中级设计人员。

本书主要由罗苑棠编写。另外参加编写的人员还有唐清善、邱宝良、周克足、刘斌、李亚捷、李永怀、李宁宇、刘伟捷、黄小欢、严剑忠、黄小宽、李彦超、付军鹏、张广安、贾素龙、王艳波、金平、徐春林、谢正义、郑贞平、张小红等。他们在资料收集、整理和技术支持方面做了大量的工作, 在此一并向他们表示感谢!

由于时间仓促, 再加之作者的水平有限, 书中难免存在一些不足之处, 欢迎广大读者批评和指正, 联系方式 jsj@phei.com.cn。

编 者

2008 年 11 月

目 录

第 1 篇 Linux 基础知识

第 1 章 嵌入式基础入门	2
1.1 嵌入式操作系统简介	2
1.1.1 嵌入式系统的基本概念	2
1.1.2 嵌入式系统的内核介绍	3
1.1.3 嵌入式系统的应用领域	4
1.2 Linux 操作系统概述	5
1.2.1 嵌入式 Linux 发展现状	5
1.2.2 Linux 相关的常用术语	6
1.3 Linux 操作系统的移植	8
1.3.1 BootLoader 技术详解	8
1.3.2 Linux 内核基本结构	17
1.3.3 移植 Linux 操作系统	28
1.4 本章总结	32
第 2 章 Linux 系统开发环境平台	33
2.1 进程/线程管理	33
2.1.1 进程/线程的概念	33
2.1.2 进程基本操作	37
2.1.3 进程通信与同步	49
2.1.4 线程基本操作	57
2.1.5 简单的多线程编程	59
2.2 文件系统结构和类型	62
2.2.1 FAT 文件系统	62
2.2.2 RAMFS 内核文件系统	66
2.2.3 JFFS 与 YAFFS 文件系统	68
2.2.4 EXT2/EXT3 文件系统	71

2.2.5 /proc 文件系统	74
2.2.6 Linux 文件操作函数	75
2.3 存储管理	79
2.3.1 MTD 内存管理	79
2.3.2 Linux 内存管理	83
2.4 设备管理	84
2.4.1 概述	84
2.4.2 字符设备与块设备	84
2.4.3 主设备号和次设备号	87
2.5 本章总结	88
第 3 章 嵌入式 Linux 程序设计基础	89
3.1 建立嵌入式 Linux 交叉编译环境	89
3.1.1 编译环境概述	89
3.1.2 建立交叉编译环境流程	92
3.2 工程管理器 make	97
3.2.1 make 概述	97
3.2.2 Makfile 文件书写规则	101
3.3 Linux C/C++ 程序设计	104
3.3.1 C/C++ 程序结构	104
3.3.2 C/C++ 数据类型	107
3.3.3 表达式/语句、函数	108
3.3.4 C/C++ 设计注意事项	111
3.4 Linux 汇编程序设计	117
3.4.1 Linux 汇编语法格式	118
3.4.2 汇编程序实例	119
3.5 Linux Shell 语言编程	120

3.5.1	Shell 环境变量及配置	
	文件	121
3.5.2	Shell 编程实例	123
3.6	Linux Perl 语言编程	124
3.6.1	Perl 基本程序	124
3.6.2	Perl 变量	125
3.6.3	文件句柄和文件操作	128
3.6.4	循环结构	129
3.6.5	条件结构	130
3.7	本章总结	131
第 4 章	Linux 常用开发工具	132
4.1	GCC 编译器	132
4.1.1	GCC 版本信息	132
4.1.2	GCC 目录结构	132
4.1.3	GCC 执行过程	133
4.1.4	GCC 的基本用法和选项	134
4.1.5	g++	134
4.2	gdb 调试器	135
4.2.1	基本用法和选项	135
4.2.2	gdb 常用命令	135
4.3	Linux 汇编工具	136
4.3.1	汇编器	136
4.3.2	链接器	136
4.3.3	调试器	137
4.3.4	系统调用	137
4.3.5	命令行参数	137
4.3.6	GCC 内联汇编	138
4.4	Linux 调试工具	139
4.4.1	JTAG 调试工具	139
4.4.2	kgdb 内核调试环境	144
4.5	Linux 图形开发工具	149
4.5.1	GUI 图形界面开发	149
4.5.2	GTK 图形开发工具	157
4.5.3	QT 图形开发工具	161
4.6	本章总结	167

第 2 篇 Linux 驱动程序 开发与实例

第 5 章	Linux 设备驱动基础	170
5.1	驱动程序基本概念	170
5.1.1	驱动程序与应用程序的区别	170
5.1.2	内核版本与编译器的版本 依赖	171
5.2	设备驱动模块概述	171
5.2.1	模块的基本概念	171
5.2.2	模块的初始化和退出	172
5.2.3	Linux 内核模块加载	174
5.3	Linux 设备驱动结构分析	176
5.3.1	内核和用户接口	176
5.3.2	inode 节点	177
5.3.3	File 结构	178
5.4	常用接口函数介绍	181
5.5	驱动程序的调试	187
5.6	本章总结	189
第 6 章	网卡驱动程序开发	190
6.1	网卡概述	190
6.2	RTL8193 网卡驱动	190
6.2.1	网卡驱动的初始化	191
6.2.2	网卡数据收发	197
6.3	典型实例——Ralink 无线网卡 驱动开发	198
6.3.1	Ralink 无线网卡	198
6.3.2	802.11 无线通信协议的选用	199
6.3.3	设备驱动关键数据结构	200
6.3.4	rt2500 无线网卡驱动分析	202
6.3.5	rt2500 程序源代码	207
6.4	本章总结	215
第 7 章	显卡驱动程序开发	216
7.1	显卡驱动概述	216
7.1.1	Linux framebuffer	216
7.1.2	帧缓冲设备数据结构	220

7.2	典型实例——显卡 Framebuffer 驱动实现	225
7.2.1	Framebuffer 驱动框架程序	225
7.2.2	NVIDIA 显卡设备驱动文件	231
7.3	本章总结	233
第 8 章	声卡驱动程序开发	234
8.1	声卡驱动概述	234
8.2	OSS 声卡驱动	234
8.3	ALSA 声卡驱动	235
8.4	典型实例——AC97 声卡驱动实现	237
8.4.1	AC97 驱动分析	237
8.4.2	Realtek 声卡驱动配置	241
8.5	本章总结	243
第 9 章	USB 驱动程序开发	244
9.1	USB 设备驱动概述	244
9.2	USB 驱动设备示例	245
9.2.1	Linux 驱动程序概述	245
9.2.2	驱动程序分析	246
9.3	典型实例——单片机的主从通信实例	253
9.3.1	主从通信介绍	253
9.3.2	USB 设备驱动程序	254
9.3.3	主机程序源代码	260
9.4	本章总结	261
第 10 章	闪存 Flash 驱动程序开发	262
10.1	Flash 闪存基础	262
10.2	Flash MTD 技术	264
10.3	典型实例 1——NAND Flash 驱动实例	265
10.3.1	NAND Flash 驱动设备	265
10.3.2	NAND Flash 驱动源代码	266
10.4	典型实例 2——NOR Flash 驱动实例	270
10.4.1	芯片驱动与 MTD 原始设备	270

10.4.2	NOR Flash 驱动分析	270
10.4.3	NOR Flash 驱动源代码	274
10.5	本章总结	276

第 3 篇 Linux 系统 开发实例

第 11 章	嵌入式系统开发的模式与流程	278
11.1	嵌入式系统的结构	278
11.1.1	嵌入式系统的硬件架构	278
11.1.2	嵌入式系统的软件结构	278
11.2	嵌入式开发的模式及流程	279
11.2.1	嵌入式系统开发模式	279
11.2.2	嵌入式系统开发流程	280
11.3	本章总结	282
第 12 章	工业温度监控设备开发实例	283
12.1	应用环境与硬件设计概要	283
12.1.1	嵌入式 Linux 在工业控制领域的应用	283
12.1.2	工控串行通信协议标准	286
12.2	相关开发技术——异步串行通信接口	288
12.2.1	异步串行通信标准	288
12.2.2	设置串口控制信号	290
12.2.3	读入串口控制信号	291
12.2.4	文件 Open() 系统调用	292
12.3	实例——基于 DS1820 的实时温度监控系统	292
12.3.1	系统基本结构	293
12.3.2	系统工作流程	296
12.3.3	系统模块源代码实现	298
12.4	本章总结	306
第 13 章	实时视频采集系统开发实例	307
13.1	应用环境与硬件设计概要	307

13.2	相关开发技术	308	15.3.1	系统基本结构	368
13.2.1	视频图像压缩技术	308	15.3.2	系统工作流程	371
13.2.2	视频采集驱动	310	15.3.3	系统模块源代码实现	372
13.2.3	视频驱动加载运行	313	15.3.4	系统调试	380
13.3	实例——基于 MV86S02 实时 视频采集系统设计	313	15.4	本章总结	381
13.3.1	系统基本结构	313			
13.3.2	系统工作流程	316	第 16 章 无线网络数据传输系统		
13.3.3	系统模块源代码实现	319	开发实例	382	
13.3.4	视频数据比较及分析	335	16.1	无线网络传输系统简介	382
13.4	本章总结	336	16.2	相关开发技术	383
			16.2.1	无线网络接入技术	383
			16.2.2	基于 PCMCIA 的无线 网卡接口	385
			16.2.3	PCMCIA 驱动程序	386
第 14 章 指纹识别门禁系统开发			16.3	实例——基于 PCMCIA 的 无线网络嵌入式前端系统 设计	387
实例	337		16.3.1	系统基本结构	387
14.1	应用环境与硬件设计概要	338	16.3.2	系统工作流程	389
14.2	相关开发技术	340	16.3.3	系统模块源代码实现	391
14.2.1	指纹识别原理	340	16.3.4	系统调试	398
14.2.2	设备驱动编写框架	344	16.4	本章总结	398
14.2.3	指纹芯片驱动	346			
14.3	实例——基于 ARM Linux 的 指纹识别门禁系统	347	第 17 章 基于 PDIUSB12 的数据 传输系统实例	399	
14.3.1	系统基本结构	347	17.1	USB 应用环境与硬件设计 概要	400
14.3.2	系统工作流程	349	17.2	相关开发技术——USB 系统 与总线驱动	401
14.3.3	系统模块源代码实现	350	17.2.1	USB 系统组成	401
14.4	本章总结	360	17.2.2	USB Host 总线驱动	402
			17.2.3	USB Device 总线驱动	403
第 15 章 基于 RTL8019 的以太网 应用系统开发实例	361		17.3	实例——基于 PDIUSB12 的 数据传输设计	406
15.1	以太网应用技术概述	361	17.3.1	系统基本结构	406
15.2	相关开发技术	362	17.3.2	系统工作流程	412
15.2.1	基于 RTL8019 的以太网 帧传输原理	362	17.3.3	系统模块源代码实现	412
15.2.2	RTL8019 的初始化	363	17.4	本章总结	424
15.2.3	RTL8019 驱动程序的框架	364			
15.2.4	数据结构和函数	365			
15.2.5	RTL8109 驱动程序的加载	368			
15.3	实例——基于 RTL8019 的 以太网应用系统设计	368			

第 18 章 家庭安全监控系统设计

实例	425
18.1 应用环境与硬件设计概要	425
18.1.1 系统功能和组成	425
18.1.2 系统模块功能描述	426
18.2 系统硬件结构	430
18.2.1 Linux 客户端系统硬件 结构	430
18.2.2 传感器系统硬件结构	433
18.3 系统软件结构	435
18.3.1 Linux 客户端系统软件 结构	435
18.3.2 传感器系统软件结构	438
18.4 Linux 客户端系统设计实现	440
18.4.1 系统数据结构设计	440
18.4.2 通信模块设计说明	441
18.4.3 显示模块设计说明	442
18.4.4 用户管理模块设计说明	443
18.4.5 系统设置模块设计说明	445
18.4.6 客户端主要代码与注释	445
18.5 系统主要模块设计实现	447
18.5.1 红外监控模块设计说明	447
18.5.2 报警模块 (warning)	448

18.5.3 触发监控模块	449
18.5.4 管理模块	450
18.5.5 主要代码与注释	453

18.6 本章总结	459
-----------------	-----

第 19 章 移动校园系统设计实例

19.1 应用环境与硬件设计概要	460
19.1.1 系统功能和组成	460
19.1.2 系统模块功能和软件图	460
19.2 系统硬件结构	462
19.3 系统软件结构	463
19.3.1 软件整体结构	463
19.3.2 软件模块结构	464
19.3.3 接口设计	467
19.3.4 运行过程设计	468
19.3.5 系统数据结构设计	469
19.3.6 搭建开发环境	470
19.4 系统模块程序代码	472
19.4.1 主函数	472
19.4.2 Syllabus 课表模块	472
19.4.3 BBS 论坛模块	474
19.4.4 Map 地图模块	476
19.4.5 Message 系统消息模块	478
19.5 本章总结	478

第 1 篇

Linux 基础知识

- ◆ 第 1 章 嵌入式基础入门
- ◆ 第 2 章 Linux 系统开发环境平台
- ◆ 第 3 章 嵌入式 Linux 程序设计基础
- ◆ 第 4 章 Linux 常用开发工具

第 1 章

嵌入式基础入门

随着微电子技术的飞速发展及后 PC 时代的到来,嵌入式芯片被广泛运用到消费、汽车、电子、微控制、无线通信、数码产品、网络设备、安全系统等领域。越来越多的公司、研究单位、大专院校,以及个人开始进行嵌入式系统的研究与应用,嵌入式系统设计将是未来相当长一段时间内电子领域研究的热点。下面首先对嵌入式操作系统进行概述。

1.1 嵌入式操作系统简介

随着嵌入式操作系统及嵌入式处理器技术的发展,嵌入式操作系统已经被广泛应用到大量以嵌入式处理器为硬件基础的系统中,常见的嵌入式操作系统有:Linux、Windows CE、Symbian、Palm 和 $\mu\text{C}/\text{OS-II}$ 等。

这些操作系统都各有自己强劲的优势,Linux 以其开源的经济优势被广泛应用到很多嵌入式系统中,得到了中小型企业青睐;Windows CE 有着全球最大的操作系统厂商 Microsoft 强大的技术后盾,得到了越来越多的市场份额;Symbian 操作系统是全球最大的手机研发制造商 NOKIA 的手机操作系统,被广泛应用于高端智能手机上。在将来相当长的一段时间内,将存在几个操作系统并存发展、齐头并进的情况,但是,经过一段时间的角逐,常用的嵌入式设备所采用的操作系统将会集中到其中的 2~3 种。

1.1.1 嵌入式系统的基本概念

业界有多种不同的关于嵌入式系统 (Embedded System) 的定义,被大多数人所接受的是根据嵌入式系统的特点下的定义:它是“以应用为中心、以计算机技术为基础、软件硬件可裁剪,对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。”该定义强调软硬件可裁剪、专用计算机系统的特点,这也是嵌入式系统与通用计算机平台最为显著的差别。

由于嵌入式的应用太广泛,因此,这里仅给出未来发展空间最为看好的嵌入式系统特点,即嵌入式系统是一类在硬件上采用专用(相对于通用的 X86 来说)的高性能处理器(通常为 32 位),在软件上以一个多任务的操作系统为基础的专用系统。一方面,它与通用的计算机平台有本质的区别(软硬件可裁剪);另一方面,又与以前的单片机有着本质的区别,因为单片机几乎无法使用移植操作系统,而 32 位嵌入式处理器设备能够很便捷地移植操作系统。

实时嵌入式系统也称为实时系统,它反映了嵌入式系统对时间响应要求较高的特点,

即如果逻辑和时序出现偏差将会引起严重后果。常见的实时系统有两种类型,即软实时系统和硬实时系统,它们各自任务要求如下。

- 软实时系统。系统的宗旨是使各个任务运行得越快越好,但并不要求限定某一任务必须在多长时间内完成。
- 硬实时系统。各任务不仅要执行无误,而且要做到准时,例如:火星车。

在实际应用中,大多数实时系统是以上二者的结合。常见的实时操作系统分为以下3类。

- 具有强实时特点的操作系统。系统响应时间在毫秒或者微秒级(如数控机床);
- 一般实时特点的操作系统。系统响应时间在毫秒到几秒的数量级上(如电子点菜机);
- 弱实时特点的操作系统。系统响应时间约数十秒以至更长时间(如MP3系统)。

下面列出部分实时操作系统所具有的特点。

(1) 高效的任务管理。实时操作系统支持多任务、优先级管理和任务调度,其中任务调度是基于优先级的抢占式调度,并采用时间片轮转调度的算法。

(2) 快速灵活的任务间通信。实时操作系统的通信机制采用消息队列和管道等技术,有效地保障快速灵活的任务间通信。

(3) 高度的可裁剪性。实时操作系统的系统功能可针对需求对软件进行裁剪、调整。

(4) 便捷地实现动态链接与部件增量加载。

(5) 快速有效地实现中断和异常事件处理。

(6) 动态内存管理。

1.1.2 嵌入式系统的内核介绍

(1) 内核(Kernel):多任务系统中,内核负责管理各个任务,或者说为每个任务分配CPU时间,并且负责任务之间的通信。内核提供的基本服务是任务切换。使用实时内核可以大大简化应用系统的设计的原因在于,实时内核允许将应用分成若干个任务,由实时内核来管理它们。内核本身也增加了应用程序的额外负荷,代码空间增加ROM的用量,内核本身的数据结构增加了RAM的用量。但更主要的是,每个任务要有自己的栈空间,这一块消耗起内存来是相当厉害的。内核本身对CPU的占用时间的比例一般是2%~5%。

单片机一般不能运行实时内核,因为单片机的RAM很有限。实时内核通过提供必不可少的系统服务,如信号量管理、邮箱、消息队列、延时等,使得CPU的利用更为有效。一旦用户用实时内核做过系统设计,将绝不再想返回到前后台系统。

(2) 调度(Scheduler):这是内核的主要职责之一,决定该轮到哪个任务运行了。多数实时内核是基于优先级调度的。每个任务根据其重要程度的不同被赋予不同的优先级。基于优先级的调度是指CPU总是让处在就绪态的优先级最高的任务先运行。然而,究竟何时让高优先级任务掌握CPU的使用权,就要看用的是什么类型的内核,是不可剥夺型的还是可剥夺型内核。

(3) 可剥夺型内核:当系统响应时间很重要时,要使用可剥夺型内核。因此, μ C/OS-II及绝大多数商业上销售的实时内核都是可剥夺型内核。最高优先级的任务一旦就绪,总能得到CPU的控制权。当一个运行着的任务使一个比它优先级高的任务进入了就绪态,当前任务的CPU使用权就被剥夺了,或者说被挂起了,那个高优先级的任务立刻得到了

CPU 的控制权。如果是中断服务子程序使一个高优先级的任务进入就绪态，中断完成时，中断了的任务被挂起，优先级高的那个任务开始运行，如图 1-1 所示。

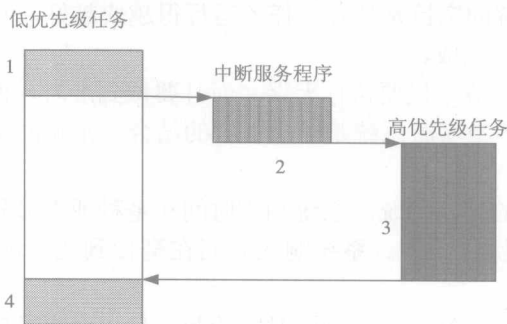


图 1-1 可剥夺型内核任务管理示意图

使用可剥夺型内核，最高优先级的任务什么时候可以执行，可以得到 CPU 的控制权是可知的。使用可剥夺型内核使得任务级响应时间得以最优化。

使用可剥夺型内核时，应用程序不应直接使用不可重入型函数。调用不可重入型函数时，要满足互斥条件，这一点可以用互斥型信号量来实现。当调用不可重入型函数时，低优先级的任务 CPU 的使用权被高优先级任务剥夺，不可重入型函数中的数据将有可能被破坏。由此可见，可剥夺型内核总是让就绪态的高优先级的任务先运行，中断服务程序可以抢占 CPU，到中断服务完成时，内核让此时优先级最高的任务运行（不一定是那个被中断了的任务）。这样，任务级系统响应时间得到了最优化，并且是可知的。 μ C/OS-II 属于可剥夺型内核。

(4) 死锁 (Deadly Embrace): 死锁也称做抱死，指两个任务无限期地互相等待对方控制着的资源。设任务 T1 正独享资源 R1，任务 T2 在独享资源 R2，而此时 T1 又要独享 R2，T2 也要独享 R1，于是哪个任务都没法继续执行了，发生了死锁。最简单的防止发生死锁的方法是让每个任务都具备以下特征。

- 先得到全部需要的资源再做下一步的工作；
- 用同样的顺序去申请多个资源；
- 释放资源时使用相反的顺序。

内核大多允许用户在申请信号量时定义等待超时，以此化解死锁。当等待时间超过了某一确定值，信号量还是无效状态，就会返回某种形式的出现超时错误的代码，这个出错代码告知该任务，不是得到了资源使用权，而是系统错误。死锁一般发生在大型多任务系统中，在嵌入式系统中不易出现。

1.1.3 嵌入式系统的应用领域

嵌入式系统的应用很广泛，可以这样说，除了通用的计算机系统应用外，其他所有的智能电子设备都属于嵌入式系统。以下简要列出嵌入式系统在数字家电、个人数据处理、通用技术等领域的应用。

嵌入式系统在数字家电领域的应用。当前家庭电子设备中包含了越来越多的嵌入式处理器产品，以下电子设备都属于嵌入式系统：IP 电话、PDA、无线音频系统、在线游戏、

无线接入设备、微波炉、电冰箱、洗衣机、电视、收音机、CD 播放器、个人电脑、遥控开关。一般来说,每个家庭拥有至少多于 20 种以上的电子设备。据有关调查表明,预计 2010 年后每个城市家庭都将基本实现电子化,那时每个人所使用的全部设备中包含的 MCU 数量将会超过 100 个!

基于嵌入式的多用途 PDA (Personal Digital Assistant) 解决方案。随着各行各业对信息化要求的日益提高,行业用户对 PDA 的需求量越来越大,同时对 PDA 的功能要求也越来越高,而且不同的行业用户所需的功能要求也不同,将呈现多元化和个性化趋势。采用嵌入式系统的智能化多功能 PDA 终端平台可以提供个性化定制业务,可定制与所在行业相对应的专用功能,并可适应不同环境下的功能要求。基于嵌入式的多用途 PDA 可以广泛运用于军警用设备、信息查询、服务行业、石油、地质、电力、水利、GPRS 应用(上网、通话、短信等)、GPS 定位、指纹识别技术、GIS (地理信息系统)应用、IC 卡应用、蓝牙技术、CCD 摄像处理、条形码识别、红外、USB 传输、多媒体、MP3 等领域。

嵌入式系统在通信领域的应用。根据市场调查,基于 ARM 处理器的嵌入式系统在 GSM/UMTS 市场(GSM850、900、1800、1900、GPRS、EDGE、UMTS)中将超过 85% 的市场占有率,并主要为 OEM (Original Equipment Manufacturer) 客户,在 CDMA 系统(IS95A/B、CDMA2000 1X、EV-DO、BREW 等)中将超过 99% 的市场占有率,在 Bluetooth 系统中将超过 75% 的市场占有率。

1.2 Linux 操作系统概述

1.2.1 嵌入式 Linux 发展现状

UNIX 操作系统于 1969 年由 Ken Thompson 在 AT&T 贝尔实验室的一台 DEC PDP-7 计算机上实现。后来 Ken Thompson 和 Dennis Ritchie 使用 C 语言对整个系统进行了再加工和编写,使得 UNIX 能够很容易地移植到其他硬件的计算机上。由于此时 AT&T 还没有把 UNIX 作为它的正式商品,因此研究人员只是在实验室内使用并完善它。也正是由于 UNIX 被作为研究项目,其他科研机构 and 大学的计算机研究人员也希望能得到这个系统,以便进行自己的研究。AT&T 以分发许可证的方法,对 UNIX 仅仅收取很少的费用,UNIX 的源代码就被散发到各个大学,使得科研人员能够根据需要改进系统,或者将其移植到其他硬件环境中去;另一方面,也培养了大量懂得 UNIX、使用 UNIX 和编程的学生,这使得 UNIX 的普及更为广泛。

到了 20 世纪 70 年代末,在 UNIX 发展到了版本 6 之后,AT&T 认识到了 UNIX 的价值,成立了 UNIX 系统实验室 (UNIX System Lab, USL) 来继续发展 UNIX。此时,AT&T 一方面继续发展内部使用的 UNIX 版本 7,一方面由 USL 开发对外正式发行的 UNIX 版本,同时 AT&T 也宣布对 UNIX 产品拥有所有权。几乎同时,加州大学伯克利分校计算机系统研究小组 (CSRG) 使用 UNIX 对操作系统进行研究,他们对 UNIX 的改进相当多,增加了很多当时非常先进的特性,包括更好的内存管理,快速且健壮的文件系统等,大部分原有的源代码都被重新写过,很多其他 UNIX 使用者,包括其他大学和商业机构,都希望能得到 CSRG 改进的 UNIX 系统。也正因为如此,CSRG 中的研究人员把他们的 UNIX 组成一个完整的 UNIX 系统——BSD UNIX (Berkeley Software Distribution) 向外发行。