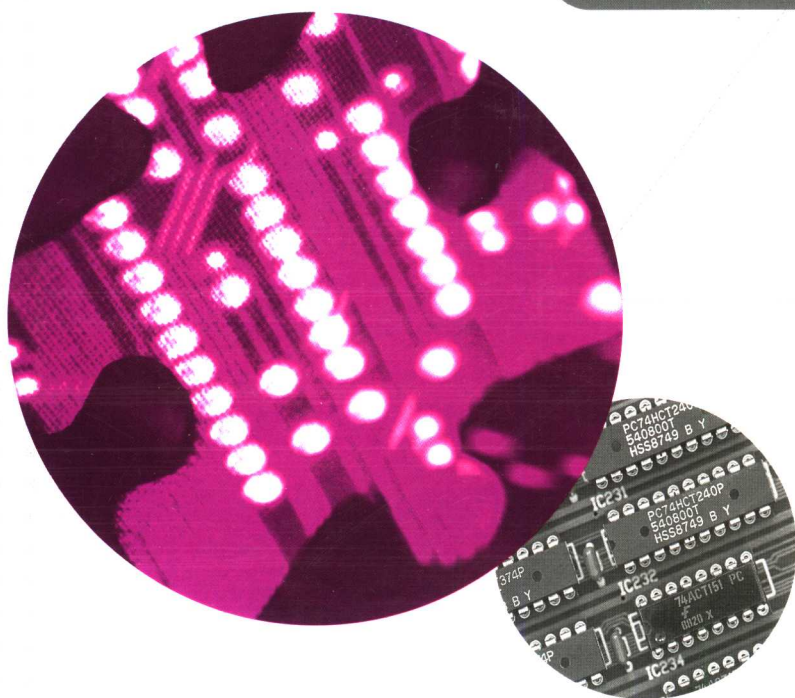


嵌入式

Linux应用系统开发

实例精讲

罗苑棠 杨宗德 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
HTTP://WWW.PHEI.COM.CN

实例典型
即学即用



电子工程应用
精讲系列

嵌入式

Linux应用系统开发

实例精讲

罗苑棠 杨宗德 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是优秀的 Linux 嵌入式应用系统开发实践指导书。全书以典型实例的形式,详细地介绍了 Linux 嵌入式开发环境平台、程序设计基础,以及大量的实际应用案例。全书分为 11 章,前 5 章为基础部分,主要讲述了 Linux 开发平台、程序设计基础、Linux 常用开发工具,以及 Linux 系统开发的一般流程;第 6~11 章为实例部分,重点介绍了几个实际应用的案例,内容包括工业温度监控设备、USB 设备驱动开发、基于 RTL8019 的以太网系统、指纹门禁考勤系统、实时视频采集系统等。这些实例具有代表性、技术领先性、应用广泛性,以及热门性,是作者多年开发经验的归纳与总结。

本书不仅讲述了 Linux 嵌入式应用系统实例开发的设计思想和流程,而且对系统底层做了详细介绍,并对关键的程序代码做了中文注释,读者通过学习便可以快速上手和应用。本书光盘与书结合,附有丰富的实例原图文件和程序源代码,方便读者学习和参考使用。

本书适合计算机、电子信息及相关专业的在校大学生,以及从事 Linux 嵌入式开发的初、中级设计人员。

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 应用系统开发实例精讲 / 罗苑棠, 杨宗德编著. —北京: 电子工业出版社, 2007.3
(电子工程应用精讲系列)
ISBN 978-7-121-03775-7

I. 嵌… II. ①罗… ②杨… 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(2007)第 004075 号

责任编辑: 胡铭姬

印 刷: 北京东光印刷厂

装 订: 三河市皇庄路通装订厂

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: 23.25 字数: 504 千字

印 次: 2007 年 3 月第 1 次印刷

印 数: 5000 册 定价: 45.00 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, 联系电话: (010) 68279077; 邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@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，另一重要原因是电子工程师的门槛相对而言比较高，而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

由于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，以及嵌入式项目开发等，包括单片机、USB 接口、ARM、CPLD/FPGA、DSP、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、自动化控制专业的在校大学生，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。特别是在程序设计思路上和决定项目开发的质量和成功与否的细节上，尽可能用简洁的语言清晰阐述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。在学习过程中，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；若能将自己的工作设计联系起来，则令达到“即学即会，学以致用”的最大化境界。

本套书主要偏重于实用性，具有很强的工程实践指导性。期望读者在学习过程中顺利、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和它的源代码，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，使人们感觉到学习 Linux 的迫切性，也使更多的人参与到 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 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。

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 Linux 嵌入式系统开发的思路、方法、技巧与实际应用案例。全书分为 11 章，第 1~5 章为基础部分，主要讲述了嵌入式基础入门、Linux 环境开发平台、程序设计基础、Linux 常用开发工具，以及 Linux 系统开发的一般流程。通过本部分学习，初级读者将可以具备一定的 Linux 程序设计功底；基础好一点的读者则可以跳过这一部分。从第 6 章到第 11 章为实例部分，重点介绍几个实际的 Linux 系统开发实例，具体包括：工业温度监控设备开发、USB 设备驱动开发、基于 RTL8019 的以太网系统开发、指纹门禁考勤系统开发，以及实时视频采集系统开发。这些实例具有代表性、技术领先性、应用广泛性，以及热门性的特点，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。

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、程序基础和设计流程，然后全部是应用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) 实例安排：本书精选了具有代表性、技术领先性的实例，这些实例全部来自于作者的开发实践，是作者多年开发经验的结晶。案例涉及不同的行业领域和方向，融会了 Linux 嵌入式系统开发的各个方面的技术。

(3) 实例讲解特色：全套书侧重于应用系统开发的流程与软硬件设计思想，并对关键的程序源代码做了中文注释。

(4) 光盘中包含丰富的实例硬件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设计，物超所值。

本书读者对象为：(1) 计算机、电子信息及相关专业的在校大学生；(2) 从事 Linux 嵌入式开发的初、中级设计人员。

本书主要由罗苑棠、杨宗德编写。另外，廖日坤、金镇、李宁宇、黄小惠、廖济林、庞丽梅、邱远彬、黄桂群、刘伟捷、黄乘传、黄小欢、严剑忠、黄小宽、唐清善、邱宝良、周克足、刘斌、李亚捷、李永怀、周卫东等同志在资料收集、整理和技术支持方面也做了大量的工作，在此一并表示感谢！

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。如果有任何意见，请联系 editor@broadview.com.cn。

目 录

第 1 章 嵌入式基础入门.....1	
1.1 嵌入式操作系统简介.....1	
1.1.1 嵌入式系统的基本概念.....1	
1.1.2 嵌入式系统的内核介绍.....2	
1.1.3 嵌入式系统的应用领域.....4	
1.2 Linux 操作系统.....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 其他嵌入式操作系统.....8	
1.3.1 Windows CE.....8	
1.3.2 Symbian OS.....9	
1.3.3 μ C/OS-II.....9	
1.3.4 Palm OS.....11	
1.4 本章总结.....11	
第 2 章 Linux 系统开发环境平台.....12	
2.1 进程/线程管理.....12	
2.1.1 进程/线程的概念.....12	
2.1.2 进程基本操作.....16	
2.1.3 进程通信与同步.....29	
2.1.4 线程基本操作.....38	
2.1.5 简单的多线程编程.....40	
2.2 文件系统结构和类型.....43	
2.2.1 FAT 文件系统.....44	
2.2.2 RAMFS 内核文件系统.....48	
2.2.3 JFFS 与 YAFFS 文件系统.....49	
2.2.4 EXT2/EXT3 文件系统.....53	
2.2.5 /proc 文件系统.....56	
2.2.6 Linux 文件操作函数.....58	
2.3 存储管理.....62	
2.3.1 MTD 内存管理.....62	
2.3.2 Linux 内存管理.....66	
2.4 设备管理.....67	
2.4.1 概述.....67	
2.4.2 字符设备与块设备.....68	
2.4.3 主设备号和次设备号.....71	
2.5 驱动程序.....72	
2.5.1 驱动程序基本概念.....73	
2.5.2 驱动设备运作过程.....73	
2.5.3 常用接口函数介绍.....75	
2.5.4 驱动程序的调试.....82	
2.6 驱动设备编写实例.....84	
2.7 本章总结.....90	
第 3 章 嵌入式 Linux 程序设计基础.....91	
3.1 建立嵌入式 Linux 交叉编译环境.....91	
3.1.1 编译环境概述.....91	
3.1.2 建立交叉编译环境流程.....94	
3.2 工程管理器 make.....101	
3.2.1 make 概述.....101	
3.2.2 Makfile 文件书写规则.....104	
3.3 Linux C/C++程序设计基础.....108	

3.3.1	C/C++程序结构	108	4.5	Linux 调试工具	143
3.3.2	C/C++数据类型	111	4.5.1	JTAG 调试工具	143
3.3.3	表达式/语句、函数	112	4.5.2	kgdb 内核调试环境	148
3.4	Linux 汇编程序设计基础	115	4.6	GTK 图形开发工具	154
3.4.1	Linux 汇编语法格式	116	4.7	本章总结	156
3.4.2	汇编程序实例	118	第 5 章	嵌入式系统开发的模式 与流程	157
3.5	SHELL 编程	119	5.1	嵌入式系统的结构	157
3.5.1	SHELL 环境变量及配置 文件	120	5.1.1	嵌入式系统的硬件架构	157
3.5.2	SHELL 编程实例	122	5.1.2	嵌入式系统的软件结构	158
3.6	Perl 编程语言	123	5.2	嵌入式开发的模式及流程	158
3.6.1	Perl 基本程序	123	5.2.1	嵌入式系统开发模式	158
3.6.2	Perl 变量	123	5.2.2	嵌入式系统开发流程	160
3.6.3	文件句柄和文件操作	127	5.3	本章总结	162
3.6.4	循环结构	128	第 6 章	工业温度监控设备开发实例	163
3.6.5	条件结构	129	6.1	应用环境与硬件设计概要	164
3.7	本章总结	130	6.1.1	嵌入式 Linux 在工业控制 领域的应用	164
第 4 章	常用的 Linux 开发工具	131	6.1.2	工控串行通信协议标准	166
4.1	GCC 编译器	131	6.2	相关开发技术——异步串行 通信接口	169
4.1.1	GCC 版本信息	131	6.2.1	异步串行通信标准	169
4.1.2	GCC 目录结构	132	6.2.2	设置串口控制信号	171
4.1.3	GCC 执行过程	132	6.2.3	读入串口控制信号	172
4.1.4	GCC 的基本用法和选项	133	6.2.4	文件 open () 系统调用	173
4.1.5	g++	134	6.3	实例——基于 DS1820 的实时 温度监控系统	174
4.2	gdb 调试器	134	6.3.1	系统基本结构	174
4.2.1	GDB 的基本用法和选项	135	6.3.2	系统工作流程	178
4.2.2	gdb 常用命令	135	6.3.3	系统模块源代码实现	180
4.3	GCC/GDB 程序编译举例	136	6.4	本章总结	188
4.4	Linux 汇编工具	139	第 7 章	USB 设备驱动开发实例	189
4.4.1	汇编器	139	7.1	USB 应用环境与硬件设计 概要	190
4.4.2	链接器	139			
4.4.3	调试器	139			
4.4.4	系统调用	140			
4.4.5	命令行参数	140			
4.4.6	GCC 内联汇编	141			

7.2	相关开发技术——USB 设备驱动程序设计	191	9.3.1	系统基本结构	252
7.2.1	USB 系统组成	192	9.3.2	系统工作流程	254
7.2.2	USB Host 总线驱动	193	9.3.3	系统模块源代码实现	255
7.2.3	USB Device 总线驱动	194	9.4	本章总结	265
7.3	实例——USB 设备驱动设计	196	第 10 章	实时视频采集系统开发	266
7.3.1	系统基本结构	197	10.1	应用环境与硬件设计概要	267
7.3.2	系统工作流程	203	10.2	相关开发技术	268
7.3.3	系统模块源代码实现	204	10.2.1	视频图像压缩技术	268
7.4	本章总结	216	10.2.2	视频采集驱动	270
第 8 章	基于 RTL8019 的以太网应用系统	217	10.2.3	视频驱动加载运行	273
8.1	以太网应用技术概述	217	10.3	实例——基于 MV86S02 实时视频采集系统设计	273
8.2	相关开发技术	218	10.3.1	系统基本结构	273
8.2.1	基于 RTL8019 的以太网帧传输原理	218	10.3.2	系统工作流程	276
8.2.2	RTL8019 的初始化	219	10.3.3	系统模块源代码实现	280
8.2.3	RTL8019 驱动程序的框架	220	10.3.4	视频数据比较及分析	296
8.2.4	数据结构和函数	222	10.4	本章总结	298
8.2.5	RTL8109 驱动程序的加载	224	第 11 章	无线网络数据传输系统开发实例	299
8.3	实例——基于 RTL8019 的以太网应用系统设计	225	11.1	无线网络传输系统简介	300
8.3.1	系统基本结构	225	11.2	相关开发技术	301
8.3.2	系统工作流程	228	11.2.1	无线网络接入技术	301
8.3.3	系统模块源代码实现	230	11.2.2	基于 PCMCIA 的无线网卡接口	302
8.3.4	系统调试	238	11.2.3	PCMCIA 驱动程序	303
8.4	本章总结	239	11.3	实例——基于 PCMCIA 的无线网络嵌入式前端系统设计	305
第 9 章	指纹识别门禁应用系统	240	11.3.1	系统基本结构	305
9.1	应用环境与硬件设计概要	241	11.3.2	系统工作流程	307
9.2	相关开发技术	244	11.3.3	系统模块源代码实现	309
9.2.1	指纹识别原理	244	11.3.4	系统调试	317
9.2.2	设备驱动编写框架	248	11.4	本章总结	317
9.2.3	指纹芯片驱动	250	附录 A	Linux 操作系统的移植	318
9.3	实例——基于 ARM Linux 的指纹识别门禁系统	251	习题及答案		346

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，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做一概述。

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 μ C/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：“以应用为中心、以计算机技术为基础、软件硬件可裁剪，功能、可靠性、成本、体积、功耗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，作者在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 的占用时间一般在 2%~5% 之间。

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 COS-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。

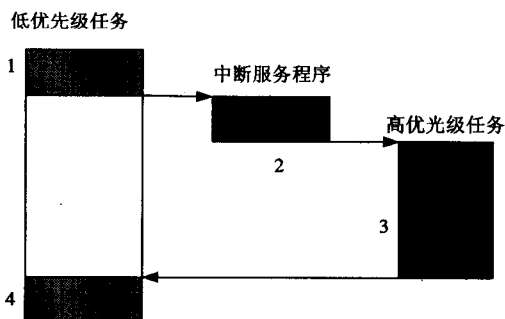


图 1-1 可剥夺型内核任务管理示意图

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，且是可知的。 μ COS-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 T2, 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、通用技术等领域的應用。

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

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的硬件环境中去;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。

而 AT&T 的 UNIX 系统实验室,同时也在不断改进他们的商用 UNIX 版本,直到他们吸收了 BSD UNIX 中已有的各种先进特性,并结合其本身的特点,推出了 UNIX System V 版本。从此以后,BSD UNIX 和 UNIX System V 形成了当今 UNIX 的两大主流,目前的 UNIX 版本大部分都是这两个版本的衍生产品:IBM 的 AIX4.0、HP/UX11、SCO 的 UNIXWare 等属于 System V,而 Minix、freeBSD、NetBSD、OpenBSD 等属于 BSD UNIX。

Linux 从一开始,就决定自由扩散 Linux,包括源代码也发布在网上,随即就引起爱好者的注意,他们通过因特网也加入了 Linux 的内核开发工作,一大批高水平程序员的加入,使得 Linux 达到迅猛发展,1993 年底,Linux 1.0 终于诞生。Linux 1.0 已经是一个功能完备的操作系统了,其内核写得紧凑高效,可以充分发挥硬件的性能,在 4MB 内存的 80386 机器上也表现得非常好。

Linux 加入 GNU 并遵循公共版权许可证 (GPL)。由于不排斥商家对自由软件的进一步开发,不排斥在 Linux 上开发商业软件,故而使 Linux 又开始了一次飞跃,出现了很多的 Linux 发行版,如 Slackware、Redhat、TurboLinux、OpenLinux 等 10 多种,而且还在增

加,还有一些公司在 Linux 上开发商业软件或把其他 UNIX 平台的软件移植到 Linux 上来,如今很多 IT 界的大腕,如 IBM、Intel、Oracle、Infomix、Sysbase、Netscape、Novell 等都宣布支持 Linux!商家的加盟弥补了纯自由软件的不足和发展障碍, Linux 得以迅速普及。

Linux 由 UNIX 操作系统的发展而来,它的内核由 Linus Torvalds 及网络上组织松散的黑客队伍一起从零开始编写而成。Linux 的目标是保持和 POSIX 的兼容。Linux 操作系统具有以下特点。

- Linux 具备现代一切功能完整的 UNIX 系统所具备的全部特征,其中包括真正的多任务、虚拟内存、共享库、需求装载、共享的写时复制程序执行、优秀的内存管理,以及 TCP/IP 网络支持等。
- Linux 的发行遵守 GNU 的通用公共许可证 (GPL)。
- 在原代码级上兼容绝大部分的 UNIX 标准 (如 IEEE POSIX, System V, BSD),它遵从 POSIX 规范。读者可以在 <http://www.linuxresources.com/what.html> 和 <http://www.linux.org> 得到更多的信息。

1.2.2 Linux 相关的常用术语

1. POSIX 及其重要地位

POSIX 表示可移植操作系统接口 (Portable Operating System Interface, 缩写为 POSIX)。由电气和电子工程师协会 (Institute of Electrical and Electronics Engineers, IEEE) 开发,主要为了提高 UNIX 环境下应用程序的可移植性。然而, POSIX 并不局限于 UNIX,许多其他的操作系统,例如, DEC OpenVMS 和 Microsoft Windows NT, 都支持 POSIX 标准,尤其是 IEEE STD.1003.1-1990 (1995 年修订) 或 POSIX.1。POSIX.1 提供了源代码级别的 C 语言应用编程接口 (API) 给操作系统的服务程序,例如,读写文件。POSIX.1 已经被国际标准化组织 (International Standards Organization, ISO) 所接受,被命名为 ISO/IEC9945-1:1990 标准。现在 POSIX 已经发展成为一个非常庞大的标准族,某些部分正处在开发过程中。

2. GNU 和 Linux 的关系

GNU 是 GNU Is Not UNIX 的递归缩写,是自由软件基金会的一个项目,该项目的目标是开发一个自由的 UNIX 版本,这一 UNIX 版本称为 HURD。尽管 HURD 尚未完成,但 GNU 项目已经开发了许多高质量的编程工具,包括 emacs 编辑器、著名的 GNU C 和 C++ 编译器 (gcc 和 g++),这些编译器可以在任何计算机系统上运行。所有的 GNU 软件和派生工作均适用 GNU 通用公共许可证,即 GPL。GPL 允许软件作者拥有软件版权,但同时授予其他任何人以合法复制、发行和修改软件的权利。

Linux 的开发使用了许多 GNU 工具。Linux 系统上用于实现 POSIX.2 标准的工具几乎都是 GNU 项目开发的, Linux 内核、GNU 工具,以及其他一些由软件组成的人们常说的 Linux: C 语言编译器和其他开发工具及函数库, X Window 窗口系统,各种应用软件 (包

括字处理软件、图像处理软件等), 其他各种 Internet 软件 (包括 FTP 服务器、WWW 服务器), 关系数据库管理系统等。

3. GPL (General Public License) 公共许可协议

GPL 的文本保存在 Linux 系统的不同目录下的命名为 COPYING 的文件里。例如, 键入 `cd/usr/doc/ghostscript*` 然后再键入 `more COPYING` 可查看 GPL 的内容。GPL 与软件是否免费无关, 它的主要目标是保证软件对所有的用户来说是自由的。GPL 通过如下途径实现这一目标:

- 要求软件以源代码的形式发布, 并规定任何用户能够以源代码的形式将软件复制或发布给别的用户。
- 提醒每个用户, 对于该软件不提供任何形式的担保。
- 如果用户的软件使用了受 GPL 保护的任何软件的一部分, 那么该软件就成为 GPL 软件, 也就是说必须随应用程序一起发布源代码。
- GPL 并不排斥对自由软件进行商业性质的包装和发行, 也不限制在自由软件的基础上打包发行其他非自由软件。
- 遵照 GPL 的软件并不是可以任意传播的, 这些软件通常都有正式的版权, GPL 在发布软件或者复制软件时声明限制条件。但是, 从用户的角度考虑, 这些根本不能算是限制条件, 相反用户只会从中受益, 因为用户可以确保获得源代码。

尽管 Linux 内核也属于 GPL 范畴, 但 GPL 并不适用于通过系统调用而使用内核服务的应用程序, 通常把这种应用程序看作是内核的正常使用。假如准备以二进制的形式发布应用程序 (像大多数商业软件那样), 则必须确保自己的程序未使用 GPL 保护的任何软件。如果软件通过库函数调用而使用了别的软件, 则不必受到这一限制。大多数函数库, 受另一种 GNU 公共许可证, 即 LGPL 的保护, 将在下面介绍。

4. LGPL (Libraray General Public License) 程序库公共许可证

GNU LGPL 的内容全部包括在命名为 COPYING.LIB 的文件中。如果安装了内核的源程序, 在任意一个源程序的目录下都可以找到 COPYING.LIB 文件的一个拷贝。

即使在不公开自己源程序的情况下, LGPL 也允许在自己的应用程序中使用程序库。但是, LGPL 还规定, 用户必须能够获得在应用程序中使用的程序库的源代码, 并且允许用户对这些程序库进行修改。

由于大多数 Linux 程序库, 包括 C 程序库 (libc.a) 都属于 LGPL 范畴。因此, 如果在 Linux 环境下, 使用 GCC 编译器建立自己的应用程序, 程序所链接的多数程序库是受 LGPL 的保护的。如果想以二进制的形式发布自己的应用程序, 则必须注意遵循 LGPL 的有关规定。

遵循 LGPL 的一种方法是, 随应用程序一起发布目标代码, 并可以发布将这些目标程序与受 LGPL 保护的、更新的 Linux 程序库链接起来的 makefile 文件。

遵循 LGPL 的比较好的一种方法是使用动态链接。使用动态链接时, 即使是程序在运行中调用函数库中的函数时, 应用程序本身和函数库也是不同的实体。通过动态链接, 用