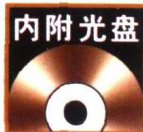


附赠书中所有
实例的源代码光盘



嵌入式 Linux

应用开发详解

刘峥嵘 张智超 许振山 等编著
郝文化 审

本书以 uClinux 为开发平台, 以指导读者方便快速地进行嵌入式 Linux 应用开发为目的, 全面涵盖嵌入式 Linux 应用开发的各个专题, 内容详尽, 条理清晰。

✘ 对如何建立自己的嵌入式 Linux 应用开发平台, 如何规划自己的应用任务进行了详尽的说明。

✘ 囊括嵌入式 Linux 应用开发的各个专题, 包括许多新兴的技术和研究方向, 如 LCD、嵌入式 GUI、USB 以及嵌入式数据库等。

✘ 使用实例的方法对每一个专题都作了全面、深入细致的介绍。



机械工业出版社
China Machine Press

本书立足于嵌入式 Linux 平台,侧重于实际开发应用,是一本结合嵌入式 Linux 技术最新发展潮流和编者多年开发经验,精心编写的嵌入式 Linux 开发用书。全书内容共分三篇:第一篇为基础篇,包括第 1 章和第 2 章,主要介绍嵌入式 Linux 的基本知识和如何配置嵌入式 Linux 开发环境;第二篇为系统篇,包括第 3~7 章,主要介绍了 uClinux 操作系统相关的知识,如任务管理、任务间通信、内存与 I/O 管理以及中断处理机制等;第三篇为应用篇,包括第 8~14 章,主要介绍了如何实现串口通信、键盘扩展、定时器、GUI、LCD 驱动和 USB 驱动、PCI 驱动、网络以及嵌入式数据库等扩展应用。在本书配套光盘中给出了书中所有实例的完整代码,读者可以作为学习和研究之用。全书涵盖了嵌入式 Linux 下应用开发的各个方面,实用性高,针对性强。

本书内容丰富,图文并茂,语言流畅,通俗易懂,可操作性强。本书既可作为各高等院校有关嵌入式 Linux 专业的教学用书,也可供在嵌入式 Linux 平台下进行应用开发的广大工作人员学习和参考。

图书在版编目(CIP)数据

嵌入式 Linux 应用开发详解/刘峥嵘等编著.

-北京:机械工业出版社,2004.6

ISBN 7-111-14765-0

I. 嵌… II. 刘… III. Linux 操作系统 IV. TP316.89

中国版本图书馆 CIP 数据核字(2004)第 060288 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:夏孟瑾 责任编辑:王金航 版式设计:侯哲芬

北京中兴印刷有限公司印刷·新华书店北京发行所发行

2004 年 7 月第 1 版第 1 次印刷

787mm×1092mm 1/16·29 印张·682 千字

0001-5000 册

定价:43.00 元(含 1CD)

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话:(010) 68993821、88379646

封面无防伪标均为盗版

前 言

Linux 之父芬兰人 Linus Torvalds 曾经说过：“我对嵌入市场感兴趣，因为那里会有些别处找不到的特殊用途。”

自从进入后 PC 时代，计算机及其相关技术得到进一步的发展，计算将不再局限于传统的 PC 和服务环境，网络计算和移动计算将很快成为人们日常生活的一部分，并逐渐呈现出普及计算（Pervasive Computing）模式。

作为普及计算的支撑技术，嵌入式实时系统正逐步应用到越来越多的领域，包括智能过程控制、航空航天、交通、飞行控制、通信、多媒体、办公自动化、实时模拟、虚拟现实、医疗电子、军事电子、信息家电等领域。

而目前企业对信息化解决方案简单化、智能化的需求，以及消费者对具有强大网络功能、简单易用、稳定低耗的下一代手持计算设备的期望，将进一步推动一个以智能网络计算部件、瘦服务器和消费电子产品为代表的的全新嵌入式系统及相关产品、解决方案市场的快速发展。嵌入式应用已经成为引人瞩目的热点。

嵌入式实时操作系统是嵌入式系统的“灵魂”。嵌入式实时操作系统的引入，改变了嵌入式系统落后的开发方式，加快了嵌入式系统的开发速度，提高了代码的可重用性和可靠性。

作为开放源代码的操作系统，嵌入式 Linux 有着与生俱来的诸多特性，使得其在满足嵌入式应用对操作系统平台的要求方面具有许多天生的优势。嵌入式 Linux 有着 Linux 的“血统”。它以可自由增减的内核为基础，采用模块化的结构，稳定小巧，易于裁剪和定制，这对于资源有限的嵌入式系统特别重要。嵌入式 Linux 是自由软件，它遵循 GPL 规范，采用开放和协作的开发模式，无偿提供源代码，任何人都可以任意取得、修改和发布源代码。这样，从开发成本来说，获取嵌入式 Linux 几乎是免费的，与其他专用嵌入式操作系统高昂的售价相比，这是它的又一大优势。而且由于它的自由性，极大地激发了世界各地的软件开发人员的积极性和创造热情，各种应用软件层出不穷，它所支持的硬件芯片也日益广泛，这为开发者提供了更多的选择。另外，嵌入式 Linux 所具有的优秀网络性能也使它成为开发具有网络功能的嵌入式产品的正确选择。嵌入式 Linux 的飞速发展也从另一个侧面证明了其自身所具有的强大优势。

本书从实际开发出发，结合实例讲解了如何解决在嵌入式 Linux 开发中所遇到的诸多难题，对于所涉及的嵌入式开发底层细节也进行了详细阐述，这些都是作者从事嵌入式 Linux 应用程序开发的经验与体会。同时，本书还对书中实例代码进行了详细解析，以期读者能更好地把握嵌入式 Linux 应用程序开发中的难点和重点。

全书分为 3 篇，共 14 章，具体内容安排如下：



第一篇 基础篇，包括第 1 章和第 2 章。主要介绍了嵌入式 Linux 应用开发的基础知识。其中，在第 1 章中对嵌入式系统、实时操作系统、嵌入式 Linux 的特点和目前国内外嵌入式 Linux 现状进行了相应介绍；在第 2 章中主要介绍了如何建立嵌入式 uClinux 开发平台，如何实现 uClinux 平台下的应用程序开发与调试，以及各类开发工具的使用，力求使读者能够搭建适合自己需要的嵌入式 Linux 开发平台，学会使用常用工具进行应用程序的开发。

第二篇 系统篇，包括第 3~7 章。主要介绍了与 uClinux 操作系统有关的知识，如任务管理、任务间通信、内存与 I/O 管理以及 uClinux 中断处理机制等。

第三篇 应用篇，包括第 8~14 章。主要对嵌入式 Linux 系统外部扩展应用的各个专题进行了相应的介绍。主要包括串口通信、LED、LCD 显示器、GUI 编程、键盘扩展、USB 驱动、PCI 驱动、网络通信和嵌入式数据库等的应用知识。

另外，在本书的配套光盘中给出了书中所有实例的完整代码，读者可以作为学习和研究之用。

本书是面向嵌入式 Linux 应用开发人员而编写的学习和参考用书。它要求读者具有一定的 C 语言基础和 Linux 操作经验，并且对微机结构、硬件体系有一定的了解。书中代码及其详尽的解析对于正在进行嵌入式 Linux 项目开发的研发人员来说具有较大的参考价值。另外，本书也可作为各高等院校计算机专业的学习用书。

在本书的编写过程中，我们得到了来自各方面的支持和帮助，在此表示感谢。

感谢博嘉公司的王松先生，在他的关心和帮助下，本书才得以顺利编写完成；感谢电子科技大学嵌入式系统教研室的熊光泽、桑楠和雷航老师给予的指导和帮助。同时，本书在编写过程中得到了家人的大力帮助和支持，在这里，我要特别感谢他们。

另外，在本书的编写过程中，还得到了专业的嵌入式 Linux 厂家的帮助。在此，我感谢华恒公司提供了嵌入式 Linux 开发板及相关资料授权。感谢北京飞漫软件技术有限公司提供的优秀的自由软件 MiniGUI 及其资料，该公司总经理兼首席培训师魏永明先生是 MiniGUI 项目的创始人，他对本书的编写给予了大力支持。

特别感谢电子产品世界和驱动开发网论坛的网友们的无私奉献以及对技术的执着追求为我们解决了很多疑难问题。另外，本书的部分资料来自网上，无法一一列出，在此一并致谢。

本书由刘峥嵘、张智超和许振山编著，由郝文化审，另外参与本书编写的还有王安贵、陈郭宜、程小英、谭小丽、卢丽娟、刘育志、吴淬砺、赵明星、贺洪俊、李小平、史利、张燕秋、周林英、黄茂英、李立、李小琼、李修华、田茂敏、苏萍、巫文斌、邹勤、栗德容等，在此一并致谢。

如果读者愿意参加“嵌入式 Linux 应用开发详解”的学习培训，或是在学习过程中发现问题，或有更好的建议，欢迎来函；我们非常愿意随时同熟悉嵌入式 Linux 技术的高手保持经常的联系，电子邮件地址：bojia@bojia.net，网址：<http://www.bojia.net>。我们将认真负责地对待每一位读者的来函。

编 者

目 录

前言

第一篇 基础篇

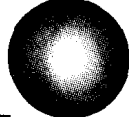
第 1 章 嵌入式 Linux 基础	1
1.1 嵌入式系统	2
1.1.1 嵌入式系统的应用	2
1.1.2 嵌入式系统的特点	3
1.2 实时与实时系统	5
1.2.1 实时与实时系统的基本概念	5
1.2.2 目前应用广泛的嵌入式实时操作系统	7
1.3 嵌入式 Linux	9
1.3.1 从 Linux 到嵌入式 Linux	9
1.3.2 嵌入式 Linux 的特点	11
1.3.3 嵌入式 Linux 发展现状	12
1.4 主流嵌入式芯片简介	14
1.4.1 Motorola 公司嵌入式芯片简介	15
1.4.2 Intel 公司 X86 体系结构嵌入式芯片简介	16
1.4.3 ARM 公司嵌入式芯片简介	17
1.5 小结	18
1.6 思考题	18
第 2 章 开发嵌入式 Linux 应用软件	19
2.1 建立嵌入式开发平台	20
2.1.1 嵌入式开发平台简介	20
2.1.2 uClinux 简介	21
2.1.3 uCsim	22
2.1.4 建立 uClinux 开发平台	23
2.2 嵌入式 Linux 软件开发工具	28
2.2.1 使用 vi 编辑器	28
2.2.2 使用 gcc 编译嵌入式 C 应用程序	31
2.2.3 编写 MakeFile	33



2.2.4	debug 工具 GDB.....	37
2.3	嵌入式 Linux 应用软件开发流程.....	40
2.3.1	对需求进行分析.....	40
2.3.2	任务和模块的划分.....	41
2.3.3	生成代码.....	43
2.3.4	调试代码.....	43
2.3.5	固化运行.....	44
2.4	一个简单的应用程序——Hello World.....	45
2.5	小结.....	51
2.6	思考题.....	51

第二篇 系 统 篇

第 3 章	任务管理.....	53
3.1	任务概述.....	54
3.1.1	标准 Linux 进程.....	54
3.1.2	任务的数据结构表示.....	55
3.1.3	实时任务.....	55
3.1.4	嵌入式 Linux 中的进程.....	55
3.2	任务状态的转变.....	56
3.3	任务调度.....	62
3.3.1	调度目标.....	62
3.3.2	调度方法分类.....	65
3.3.3	经典常用实时调度算法.....	68
3.3.4	多处理器调度算法.....	74
3.3.5	Linux 进程调度.....	74
3.4	常用任务管理 API.....	85
3.5	关于任务的实例.....	90
3.6	小结.....	93
3.7	思考题.....	94
第 4 章	任务的同步与通信.....	95
4.1	任务间同步与互斥.....	96
4.2	任务间的同步.....	97
4.2.1	重要概念.....	97
4.2.2	信号量的概念.....	102
4.3	任务间的通信.....	105
4.4	嵌入式 Linux 中的任务间同步与通信.....	106



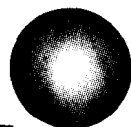
4.4.1	Linux 中的信号	106
4.4.2	Linux 中的管道	117
4.4.3	先进先出文件 FIFO	119
4.4.4	System V IPC 机制	120
4.5	小结	130
4.6	思考题	130
第 5 章	存储器管理	131
5.1	存储器管理概述	132
5.1.1	基本概念	133
5.1.2	内存管理的分类	135
5.1.3	早期连续内存分配	135
5.1.4	基于段、页的存储管理	137
5.1.5	虚拟存储器管理	142
5.2	Linux 存储器管理	146
5.2.1	物理内存管理	147
5.2.2	虚拟内存管理	150
5.3	小结	165
5.4	思考题	166
第 6 章	中断处理	167
6.1	中断概述	168
6.1.1	中断源	168
6.1.2	中断类型号、中断向量表和中断描述符表	169
6.1.3	中断服务程序及其入口地址	169
6.1.4	中断优先级和中断嵌套	170
6.2	中断机制	171
6.2.1	中断响应过程	171
6.2.2	中断服务程序设计	173
6.3	使用中断驱动串口	173
6.3.1	PC 机串口的基本概念	173
6.3.2	PC 机串口驱动程序的实现	178
6.4	小结	190
6.5	思考题	191
第 7 章	嵌入式 Linux 下串口通信	192
7.1	串行 I/O 的基本概念	193
7.1.1	同步通信与异步通信	193
7.1.2	串口传输速率与流控	194



7.1.3	差错控制.....	196
7.1.4	DTE 和 DCE 通信过程.....	197
7.1.5	RS-232C 串口规范简介.....	199
7.2	编写串口通信程序.....	202
7.2.1	嵌入式 Linux 驱动程序简介.....	202
7.2.2	串口访问函数.....	208
7.2.3	设置串口属性.....	210
7.3	嵌入式 Linux 串口通信实例.....	212
7.4	小结.....	221
7.5	思考题.....	221

第三篇 应 用 篇

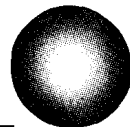
第 8 章	键盘开发和时钟管理.....	222
8.1	最简单的键盘——按键开关.....	223
8.1.1	按键开关电路.....	223
8.1.2	去除按键抖动.....	224
8.1.3	把按键接入嵌入式系统.....	225
8.2	在嵌入式系统中扩展键盘.....	225
8.2.1	矩阵键盘.....	226
8.2.2	用 Intel 8279 扩展键盘.....	228
8.3	嵌入式 Linux 时钟管理.....	239
8.3.1	时间日期管理.....	239
8.3.2	用户任务中的定时器.....	244
8.3.3	内核中的时钟管理.....	246
8.4	小结.....	249
8.5	思考题.....	249
第 9 章	图形界面应用程序开发.....	250
9.1	嵌入式 GUI 特点及种类.....	251
9.2	MiniGUI 简介.....	252
9.2.1	MiniGUI 是什么.....	252
9.2.2	MiniGUI 特点及优势.....	252
9.2.3	MiniGUI 的安装与配置.....	253
9.3	MiniGUI 程序框架及示例.....	255
9.3.1	主函数 MiniGUIMain().....	255
9.3.2	消息处理函数.....	256
9.3.3	第一个 MiniGUI 程序.....	256



9.4	MiniGUI 中的窗口与消息	258
9.4.1	窗口的建立与销毁	258
9.4.2	消息与消息循环	259
9.4.3	几个重要的消息	261
9.5	键盘与鼠标	262
9.5.1	键盘消息与字符消息	263
9.5.2	鼠标消息	263
9.6	绘图工具与图形设备接口	264
9.6.1	设备描述表	265
9.6.2	画点与画线	267
9.6.3	封闭曲线及区域填充	268
9.6.4	字体与文字输出	269
9.7	MiniGUI 中的常用控件	271
9.7.1	静态控件与按钮控件	273
9.7.2	列表框	276
9.7.3	编辑控件	279
9.7.4	工具栏控件	280
9.7.5	控件子类化	282
9.7.6	自定义控件	284
9.8	对话框	285
9.8.1	创建模式对话框	286
9.8.2	创建非模式对话框	288
9.8.3	带属性页的对话框	288
9.9	菜单的使用	293
9.9.1	创建菜单	293
9.9.2	处理菜单消息	296
9.9.3	更改菜单项状态	297
9.10	小结	298
9.11	思考题	298
第 10 章	USB 设备驱动程序开发	299
10.1	USB 体系结构	300
10.1.1	USB 系统的描述	300
10.1.2	电气特性	301
10.1.3	电源分配与管理	302
10.2	USB 通信协议	302
10.2.1	USB 数据流模型	302
10.2.2	USB 数据单元	303



10.2.3	USB 设备请求	306
10.2.4	USB 设备枚举	308
10.2.5	小结	309
10.3	USB 设备驱动程序设计	309
10.3.1	USB 设备驱动程序分类	309
10.3.2	主机端设备驱动程序分析	310
10.3.3	设备端 USB 驱动程序分析	316
10.4	小结	328
10.5	思考题	328
第 11 章	用 LED 和 LCD 作系统输出	329
11.1	在嵌入式 Linux 系统中扩展 LED 输出	330
11.1.1	LED 显示输出的原理和结构	330
11.1.2	LED 显示方式	331
11.1.3	在嵌入式 Linux 系统中使用 LED 显示器	334
11.2	LCD 显示器的使用	338
11.2.1	LCD 简介	338
11.2.2	在嵌入式 Linux 中驱动 LCD	342
11.3	在嵌入式 Linux 中使用 LCD	346
11.3.1	EZ328 对 LCD 的支持	346
11.3.2	uClinux 对 LCD 显示器的支持	347
11.3.3	图形 API 使用实例	352
11.4	小结	357
11.5	思考题	358
第 12 章	在嵌入式 Linux 系统中扩展 PCI 设备	359
12.1	PCI 总线规范	360
12.1.1	PCI 总线规范简介	360
12.1.2	PCI 配置空间	365
12.2	嵌入式 Linux 对 PCI 设备的支持	371
12.2.1	扫描 PCI 设备	371
12.2.2	为 PCI 设备分配资源	376
12.2.3	对 PCI 配置空间的访问	380
12.3	编写嵌入式 Linux 下 PCI 驱动程序	381
12.3.1	编写 PCI 驱动程序	381
12.3.2	嵌入式 Linux 下 PCI 驱动实例——NE2000 网卡驱动程序	387
12.4	小结	393
12.5	思考题	393



第 13 章 嵌入式 Linux 网络编程	394
13.1 嵌入式 Linux 网络体系结构	395
13.1.1 TCP/IP 网络简介	395
13.1.2 嵌入式 Linux 中 TCP/IP 网络结构	398
13.2 嵌入式 Linux 环境下的 socket 编程	399
13.2.1 套接字接口	400
13.2.2 socket 编程基础	403
13.2.3 socket 通信用 API 函数	404
13.2.4 数据流和数据报通信	409
13.2.5 socket 编程高级特性	417
13.3 网络编程实例——使用 socket 编写代理服务器	421
13.3.1 功能说明	421
13.3.2 代码	422
13.3.3 代码分析	428
13.4 小结	433
13.5 思考题	433
第 14 章 嵌入式数据库	434
14.1 嵌入式系统中的数据库	435
14.1.1 嵌入式数据库特点	435
14.1.2 嵌入式数据库现状与发展	435
14.2 mSQL 简介	435
14.3 在 Linux 上安装和配置 mSQL	436
14.3.1 mSQL 的安装	436
14.3.2 mSQL 系统配置	437
14.4 mSQL 工具程序	439
14.5 mSQL 的 C API 函数	442
14.6 mSQL 嵌入式数据库应用实例分析	446
14.7 小结	448
14.8 思考题	448
参考文献	449

第一篇 基础篇

第 1 章 嵌入式 Linux 基础

知识点:

- 嵌入式系统的应用
- 嵌入式系统的特点
- 实时与实时系统
- 嵌入式 Linux 的特点与现状
- 各种主流嵌入式微处理芯片

本章导读:

本章将主要介绍嵌入式 Linux 的基本知识。通过本章的学习,读者可以对嵌入式系统的应用场合、特点及对实时与实时操作系统的基本特点有所了解,为学习后续章节的内容打下良好的基础。



1.1 嵌入式系统

1.1.1 嵌入式系统的应用

当前，人类进入信息大爆炸的时代，各类信息极度丰富，数字信息技术和网络技术高度发达，只有借助各种计算机，才能对各类信息进行处理。同时，这些计算机不再局限于以前的 PC，而是包括形态各异、性能千差万别的各类嵌入式系统——从基于群集的超级计算机到嵌入在冰箱中的微控制器。

后 PC 时代的到来，使得人们开始越来越多地接触到一个新的概念——嵌入式产品。嵌入式产品遍布于人们的日常生活，从手机、PDA 到家中的空调、冰箱，从小汽车到波音飞机，甚至武器库中的巡航导弹。数字时代的标志不再是一台一台的 PC，而是形态各异的嵌入式系统。

嵌入式系统的概念的提出已经有相当长的时间，其历史几乎和计算机的历史一样长。但在以前，它主要用于军事领域和工业控制领域，所以很少被常人关注和了解。直到最近，随着数字技术的发展和新的体积更小的控制芯片和功能更强的操作系统的出现，它才被广泛应用于人们的日常生活中。

现在，嵌入式产品已经在很多领域得到广泛的使用，如国防、工业控制、通信、办公自动化和消费电子领域等。

1. 工业过程控制

工业过程控制即是对工业生产过程中的生产流程加以控制。这种控制是建立在对被控对象和环境不断进行监控的基础上的。在控制过程中，嵌入式的计算机处于中心位置，它通过分布在工业生产中的各个传感器收集信息，并对这些信息进行加工处理和判断，然后向执行器件发出控制指令。整个控制流程如图 1-1 所示。

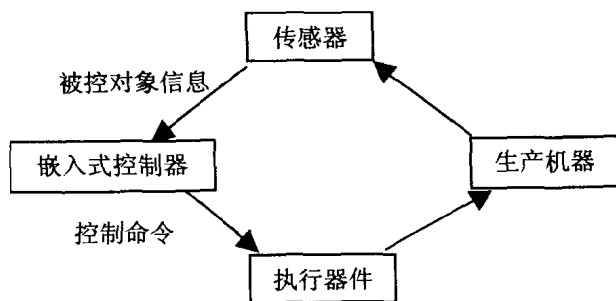


图 1-1 工业控制中的嵌入式应用模型

2. 军事电子设备和现代武器

这是早期嵌入式系统的重要应用领域，军事领域从来就是许多高新技术的发源地。由

于内装嵌入式计算机的设备反应速度快，自动化程度高，所以威力巨大，自然很得军方青睐。从爱国者导弹的制导系统到战斗机的瞄准器，从 M1A2 的火控系统到单兵系统的通信器，都可觅得嵌入式系统的踪迹。

3. 网络通信设备

众多网络设备都是嵌入式系统使用的典型例子，如路由器、交换机、Web 服务器、网络接入设备等。另外，在后 PC 时代将会产生比 PC 时代多成百上千倍的瘦服务器和超级嵌入式瘦服务器，这些瘦服务器将为人们提供需要的各种信息，并通过 Internet 自动、实时、方便、简单地提供给需要这些信息的对象。设计和制造嵌入式瘦服务器、嵌入式网关和嵌入式因特网路由器已成为嵌入式系统的一大应用方向，这些设备为企业信息化提供了廉价的解决方案。

4. 消费电子产品

后 PC 时代的消费电子产品应具有强大的网络和多媒体处理能力。易用的界面和丰富的应用功能。这些特性的实现，都依赖于嵌入式系统提供的强大的数字处理能力和简洁实用的特性。

作为移动计算设备的 PDA 和手机已出现融合趋势，未来必然是二者合一，提供给用户随时随地访问 Internet 的能力，同时还具有其他信息服务功能，如文字处理、邮件管理、个人事务管理和多媒体信息服务等。而且简单易用，价格低廉，维护简便。

作为消费电子产品的嵌入式系统的另一大应用是信息家电。信息电器是指所有能提供信息服务或通过网络系统交互信息的消费类电子产品。如前几年打得火热的“维纳斯”与“女蜗”之战就是信息家电中的机顶盒之争。如果在家电中的冰箱、空调、监控器等设备中嵌入计算机并提供网络访问能力，用户就可以通过网络随时随地地了解家中的情况，并控制家中的相应电器。

总之，随着信息技术的发展，人类进入一个全新的数字时代，数字化产品空前繁荣，嵌入式系统被应用于空前广泛的领域。在以后相当长一段时间内，嵌入式技术将在消费电子领域进一步飞速发展，嵌入式产品将与人们的生活结合得越来越紧密。

1.1.2 嵌入式系统的特点

什么叫嵌入式系统？嵌入式系统就是以应用为中心，以计算机技术为基础，软硬件可裁减，适合应用系统对功能、可靠性、成本、体积和功耗要求的专用的计算机系统。

在嵌入式系统中，计算机系统一般作为智能控制部件嵌入到整个应用系统中，是整个系统的控制中心，主要用于对系统的信息处理部件和用户交互界面加以控制。在这种情况下，用户并不知道（或者不需要知道）嵌入的计算机的存在，系统控制软件一般被固化在嵌入式计算机中，嵌入式计算机一般不需要（或不能）被用户重新编程，通过特殊的输入、输出设备与系统进行交互。

任何嵌入式系统都包括硬件和软件两个方面。硬件包括微处理器、存储器、I/O 端口和



图形控制器等。软件包括操作系统软件和应用软件，应用软件控制着嵌入式系统的运作和行为，而操作系统则为应用程序提供必要的底层支持，它一般是通过提供应用编程接口（API）来实现的。但在嵌入式系统开发中它们的区别又不是绝对明显的，有时嵌入式系统的开发者可能要做操作系统和应用编程两方面的工作。

因为嵌入式系统是面向应用、产品和用户的，所以不可能不研究应用特性以开发出一个如 PC 般通用的嵌入式系统。在嵌入式系统中，具体的应用将决定对硬件和软件的需求，如芯片、存储器、I/O 扩展和操作系统、应用程序编制等。和通用计算机不同，嵌入式系统的硬件和软件都必须高效率地设计，量体裁衣，去除冗余，尽量以最小的系统、最低的成本去实现目标功能，这样的产品才具有竞争力。它通常都具有低功耗、体积小、集成度高等特点，能够把通用 CPU 中许多由板卡完成的任务集成在芯片内部，从而有利于嵌入式系统设计趋于小型化，移动能力大大增强，跟网络的结合也越来越紧密。

下面具体介绍嵌入式系统的软硬件特性。

1. 硬件特性

嵌入式系统总是面向特定应用的，与通用 PC 的硬件相比，它的硬件系统具有以下特性：

- 体积小，集成效率高。嵌入式系统总是去除冗余，力争用最小的系统完成目标功能，特别在一些手持设备中更是这样。
 - 面向特定应用的特性。具体嵌入式系统只能适合某一特定应用，针对另一应用就需要重新设计硬件系统。
 - 低功耗，电磁兼容性好，能在恶劣环境下工作，即使死机也要求能够快速重启。
- 总之，嵌入式系统硬件在价格、功能、体积、重量、能耗等方面都有严格的限制。

2. 软件特性

软件是一个应用系统的灵魂，对于嵌入式系统的软件部分，它具有以下特点：

- 嵌入式软件的研发与硬件紧密相关。由于嵌入式软件的开发是针对具体硬件平台进行的，它往往牵涉硬件驱动方面的一些软硬结合部分，这就要求开发人员必须具备相关的硬件知识。
- 软件代码要求高效率和高可靠性（小）。由于嵌入式系统中软件运行空间有限，内存空间非常宝贵，在软件的编程过程中必须时刻考虑软件的运行效率，同时选用高质量的编译工具。在实时系统中，处理器必须严格处理异步发生的各种任务，这对程序的算法设计提出了更高的要求。另外，嵌入式软件系统还应该有异常处理、快速复位等特点。
- 软件一般固化在 FLASH 或 ROM 中。为了提高执行速度和系统的可靠性，同时缩短系统复位时间，一般在嵌入式软件调试好后，会下载固化到目标板中的 FLASH 或 ROM 中。目标板启动时，再运行其中的代码，而不是像 PC 那样从硬盘存储器中读取程序。

1.2 实时与实时系统

1.2.1 实时与实时系统的基本概念

在嵌入式领域中一个非常重要的概念是实时，实时系统是指在确定的时间内完成规定功能，并能对外部异步事件作出正确响应的计算机系统。设计实时系统时，不仅要考虑算法逻辑设计的正确性，而且要考虑执行这些算法的时间相关性。

实时系统的核心是必须在确定的时间内执行完一项预先定义的操作，否则将引起性能下降甚至系统崩溃等严重后果。比如在一个工业控制系统中，传感器每秒钟收集 5 组数据，而处理器必须根据每组数据发出相应的控制指令，这就要求处理器处理每组数据的时间不能大于 1/5 秒，否则在上一组数据还没有处理完，相应的控制命令不能及时发送出去，新的一组数据又输入进来，这样必将导致系统崩溃。

需要说明的是，实时系统并不是说系统的响应和处理速度非常快；而一个高速系统也不一定是实时系统，就如一个使用 UNIX 操作系统的大型服务器，不一定是实时系统一样。实时系统对处理速度的要求是千差万别的，必须视具体应用而定。高性能的粒子甄别系统的甄别处理可能要求在微秒甚至纳秒内作出判断，而上一个例子只要求秒级的处理速度。

实时系统中实时性的实现需要硬软件的配合来完成。首先必须保证硬件的处理速度满足实时要求，同时相对软件而言，实时性体现在组成软件系统的各个任务的执行时限。在实时系统中，每个任务的运行都是有时限（deadline）的，即任务执行的最长时间。

实时系统按时限对系统的影响程度不同，分为软实时系统（soft real-time system）和硬实时系统（hard real-time system）。前者对偶尔超过时限的操作是可以容忍的，并且具有相应的处理机制，不会因为个别存在的超时限操作而引起严重的后果，尽管这可能造成系统性能的下降；而后者是不能容忍的，在硬实时系统中，超过时限的后果是无法预测的。

在实时系统中，用响应时间（response time）、生存时间（survival time）和吞吐量（throughput）等 3 个指标来衡量系统的实时性。下面对这 3 个指标进行相应的说明。

- 响应时间：是实时系统从识别一个外部事件到作出响应的的时间，在控制应用中它是最重要的指标，如果在硬实时系统中外部事件不能及时响应处理，系统将可能崩溃。
- 生存时间：是数据的有效等待时间，在该时间内数据是有效的，超过生存时间的数据是无效的，如在上面的工业控制的例子中，数据的生存时间为 1/5 秒，超过这段时间缓冲区的数据就将无效。
- 吞吐量：是在一个给定时间内，系统可以处理的事件总数。例如 MODEM 中用每秒钟处理的比特数表示吞吐量。吞吐量可能是平均响应时间的倒数，但一般要小一些，因为在每次响应后还要做处理工作，有一段恢复时间。

与 PC 中所使用的操作系统一样，在实时领域也有实时操作系统。实时操作系统（Real-time Operating System, RTOS）是基于计算机的，是管理计算机软硬件资源并提供



人机命令和编程接口的系统，它能在固定的时间内对一个或多个由外设发出的信号作出适当的反应，同时它还是嵌入式软件开发平台。与普通分时操作系统不同，实时系统强调系统对外部异步事件响应时间的确定性。RTOS 是操作系统的一个重要分支，它与通用的商用操作系统如 UNIX、Windows 有共同的一面，也有不同的一面。对于一般的商用操作系统，它的目的是方便用户管理计算机资源，以最大限度地利用计算机的资源；而 RTOS 追求的是任务切换的实时性、事件响应时间的可确定性及系统的高可靠性。评价一个实时系统的性能要从任务调度功能、内存管理功能、最小内存开销、任务切换时间、最大中断禁止时间等方面来考虑。下面对这些概念进行简单的说明，详细介绍将在后面章节中进行。

1. 任务与任务调度

在 RTOS 中的任务是指一个程序分段，这个分段被操作系统当作一个基本单位来调度，它大致相当于 PC 机中操作系统的进程的概念。任务是一个可运行的程序段，它一般完成一项单一的功能。任务自己认为它独自占有 CPU 资源。在嵌入式软件的设计过程中，要把问题分化为一个一个的任务，每个任务都是应用的某一部分，完成一部分功能，它们被赋予一定优先级，有自己的一套 CPU 寄存器和堆栈空间。在嵌入式软件设计中多任务的存在可以使 CPU 利用达到最大值，并且可以实现程序模块化。

在只有一个 CPU 的嵌入式系统中，多任务运行的实现要靠 RTOS 在许多任务之间切换、调度。任务调度即决定应该运行哪个任务，这是内核的主要职责之一。多数实时操作系统都是基于优先级调度算法的。即每个任务都被赋予一定的优先级，而 CPU 总是执行处于就绪态的优先级最高的任务，但究竟何时让优先级最高的任务掌握 CPU 使用权，要看 RTOS 的内核类型（即是可剥夺抢占式的，还是不可剥夺抢占式的），具体相关知识的介绍，请参见第 3 章关于任务调度的内容。

2. 内存管理

它是针对有内存管理单元（MMU）的处理器设计的一些桌面操作系统，如 Windows、Linux，都引入了虚拟存储器的概念。虚拟内存地址被送到内存管理单元（Memory Management Unit, MMU）。在这里，虚拟地址被映射为物理地址，实际存储器被分割为相同大小的页面，采用分页的方式载入进程。一个程序在运行之前，没有必要全部装入内存，而是仅将那些当前需要运行的部分页面装入内存运行。在许多 RTOS 中，也采用 MMU 进行内存管理，并且它的内存管理模式可以分为实模式和保护模式。

还有许多嵌入式系统是针对没有 MMU 的处理器设计的，在这里不能使用处理器的虚拟内存管理技术，采用的是实存储器管理策略。因而对于内存的访问是直接的，它对地址的访问不需要经过 MMU，而是直接送到地址线上输出，所有程序中访问的地址都是实际的物理地址。而且，大多数的嵌入式操作系统都对内存空间没有保护，各个进程实际上都共享一个运行空间。一个进程在执行前，系统必须为它分配足够的连续地址空间，然后将其全部载入主存储器的连续空间，如嵌入式 Linux 领域的 uClinux 就是这样。

在嵌入式系统的开发中，RTOS 的内存管理能力是选择实时操作系统的重要参考因素，因为嵌入式开发人员也不得不参与系统的内存管理。从编译内核开始，开发人员必须告诉