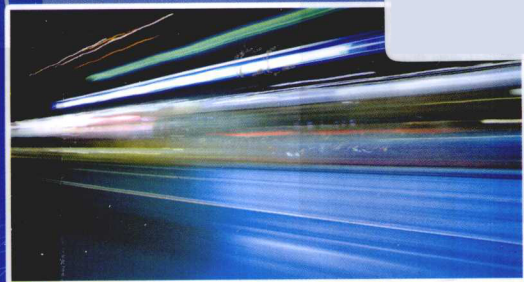




普通高等教育“十二五”规划教材
电子信息科学与工程类专业规划教材

嵌入式操作系统 原理及应用

严海蓉 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

嵌入式操作系统 原理及应用

严海蓉 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书重点讲述嵌入式操作系统原理及应用。全书共分 14 章, 包括嵌入式操作系统基础知识, μ COS-II 在 FPGA 平台上的移植, 嵌入式操作系统的构成元素, 嵌入式操作系统一般的启动方式, 任务管理的各个组成模块及其实现方式, 任务间通信方式及其实现, 内存管理方式及其实现, 中断与异步通信的方式及其实现, Linux 内核及驱动编写, μ Clinux 内核及驱动编写, Android 体系结构及蓝牙驱动编写, Windows CE 内核及驱动编写等内容。本书最后还提供了实验指导, 供读者参考。

本书内容新颖、层次清晰, 可作为高等院校本科、研究生各相关专业的程序设计教材, 也适合嵌入式系统开发人员自学使用。

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

嵌入式操作系统原理及应用 / 严海蓉编著. —北京: 电子工业出版社, 2012.6

ISBN 978-7-121-16953-3

I. ①嵌… II. ①严… III. ①实时操作系统—高等学校—教材 IV. ①TP316.2

中国版本图书馆 CIP 数据核字 (2012) 第 088230 号

责任编辑: 董亚峰 特约编辑: 王 纲

印 刷: 北京市李史山胶印厂
装 订:

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: 787×1 092 1/16 印张: 14.5 字数: 380 千字

印 次: 2012 年 6 月第 1 次印刷

印 数: 4 000 册 定价: 32.00 元

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 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。

前 言

嵌入式系统与计算机系统的起源一样久远，由于嵌入式系统的硬件形形色色，软件也带着各个行业的不同特点，因此一直没有像计算机系统一样被大众普遍认识。尤其是嵌入式操作系统，虽然也风风雨雨发展了很多年，但是由于应用的范围比计算机操作系统小，因此也不被大众所了解。

可是随着计算机系统发展受到了固有统一形态的限制，嵌入式系统在近几个世纪呈现出了蓬勃的发展势头。嵌入式操作系统也随着要求软件的编写更加简单和实现多硬件平台的可跨越而逐渐成熟起来。随着物联网的概念出现，嵌入式系统也将会成为这一轮科技革命的主力军。

在近年来的教学实践中以及在和学生们的探讨中，我们感到必须编写一本专门介绍嵌入式操作系统原理的内容新颖全面的教材，用于指导学生的学习。本书重点讲述嵌入式操作系统原理，尤其是不同于计算机操作系统的一些原理。由于有些嵌入式系统的规模较小，需要自己编写操作系统，因此本书侧重于介绍怎样编写嵌入式操作系统内核。同时也介绍了 μ COS-II在Xilinx FPGA上的移植，让读者了解如何移植一个已有的嵌入式操作系统到另一个硬件平台。最后还以编写设备驱动为主线介绍了各种常用的嵌入式操作系统，包括Linux、 μ Clinux、Andriod、Windows CE等，让读者大致了解嵌入式操作系统的组织。

本书内容可分为以下两大部分。

第1~10章为第一部分，属于嵌入式操作系统原理。其中，第1章介绍嵌入式操作系统基础知识，如嵌入式操作系统的发展历史、分类、编写方式。第2章详细介绍本书嵌入式操作系统实验的运行平台，包括 μ COS-II和Xilinx公司的FPGA平台，以及 μ COS-II在FPGA平台上的移植，让读者体会嵌入式操作系统的具体硬件接口设计。第3章介绍嵌入式操作系统的体系结构和基本概念。第4章详细介绍嵌入式操作系统的构成元素——各种常用数据结构和其在嵌入式操作系统中的使用方式。第5章详细介绍嵌入式操作系统的一般启动方式，包括bootloader的编写方式， μ COS-II、 μ Clinux的启动方式和用到的数据结构。第6章详细介绍任务管理的各个组成模块及其实现方式。第7章详细介绍嵌入式操作系统中资源管理的各种方式及实现方式，并通过对 μ COS-II的改造，讲述如何在已有嵌入式操作系统中实现所需要功能模块做扩展。第8章详细介绍任务间通信方式及其实现。第9章详细介绍内存管理方式及其实现。第10章详细介绍中断与异步通信的方式及其实现。

第11~14章为第二部分，属于常用嵌入式操作系统的介绍和应用。其中，第11章介绍Linux内核及驱动编写。第12章介绍 μ Clinux内核及驱动编写。第13章介绍Android体系结构及蓝牙驱动编写。第14章介绍Windows CE内核及驱动编写。

本书内容新颖，立足点高，同时力求重点突出，层次清晰，语言通俗易懂，内容覆盖面广，可作为高等院校本科、研究生各相关专业（如嵌入式系统、物联网、计算机、电子信息、通信）的程序设计教材，也适合于程序开发人员培训、广大嵌入式系统技术爱好者自学使用。

根据我们的教学体会，本书的教学可以安排为32~48学时。如果安排的学时较少，可以根据学生的水平适当删减第二部分的内容。

尽管我们在写作过程中投入了大量的时间和精力，但由于水平有限，错误和不足之处仍在所难免，敬请读者批评指正（任何建议可以发邮件至 yanhairong@bjut.edu.cn）。我们会在适当

的时间对本书进行修订和补充。

本书第 2 章及实验指导由张川编写，第 11 章由陈连升组织内容，第 13 章由丁亚周组织部分内容。Digilent 公司的赵峰经理审阅了本书并提供了宝贵的修改意见，Xilinx 公司的谢凯年经理对本书结构提出了许多宝贵意见，Digilent 公司的赵峰经理和戴硕先生对本书实验部分给予很大的帮助，在此一并表示感谢。同时还要感谢北京工业大学嵌入式系统与软件系的全体师生，本书的最终出版得到了他们的帮助。北京工业大学研究生教材建设基金对本书的编写进行了资助。最后还要感谢我的家人对我的支持，感谢电子工业出版社的董亚峰策划编辑为本书的编写和出版付出的辛勤劳动，感谢张杰、武翠梅等同学的帮助。

本书提供的实验实例全部在目标硬件上调试通过，读者可登录电子工业出版社华信教育资源网 (<http://www.hxedu.com.cn>) 免费下载本书配套实验指导与程序文件。

严海蓉

2012 年 2 月

于北京工业大学

目 录

第 1 章 绪论	1
1.1 嵌入式操作系统的发展历史	1
1.2 嵌入式操作系统的分类	2
1.3 嵌入式操作系统的组成	3
1.4 目前主流嵌入式操作系统概述	4
1.5 嵌入式操作系统的发展趋势	5
1.6 编写嵌入式操作系统的方法	6
习题	7
第 2 章 嵌入式操作系统的运行环境	8
2.1 运行平台介绍	8
2.2 嵌入式操作系统文件构成与接口定义	11
2.3 嵌入式操作系统下载与运行	13
第 3 章 嵌入式操作系统	14
3.1 嵌入式操作系统的定义	14
3.2 嵌入式操作系统的体系结构	14
3.2.1 整体型	14
3.2.2 层次型	15
3.2.3 微内核	16
3.2.4 客户-服务器	16
3.3 嵌入式操作系统组成要素及概念	17
习题	20
第 4 章 嵌入式操作系统常用数据结构	21
4.1 数组	21
4.2 指针	22
4.3 结构体	22
4.4 链表	23
4.5 差分链表	29
4.6 树	30
4.7 位图	31
4.8 文件	32
4.9 内核线程	34

习题	36
第 5 章 嵌入式操作系统初始化	37
5.1 bootloader	37
5.1.1 bootloder 装在哪里	37
5.1.2 bootloder 的启动过程	37
5.1.3 基于 MicroBlaze 软核处理器的 bootloder 设计	47
5.2 嵌入式操作系统初始化数据结构及主要操作	48
5.2.1 μ COS 主要数据结构及操作	48
5.2.2 μ COS 系统初始化	52
5.2.3 μ Clinux 的系统初始化	54
习题	56
第 6 章 任务管理	57
6.1 任务和任务优先级	57
6.2 任务状态	58
6.3 任务控制块链	59
6.4 任务的生成	59
6.5 任务的挂起	63
6.6 任务的唤醒	64
6.7 任务的删除	65
6.8 任务调度	67
6.9 任务编程	69
习题	73
第 7 章 资源管理	74
7.1 资源共享、互斥和任务同步	74
7.2 临界区	74
7.3 信号量	76
7.4 信号量的使用	81
7.5 优先级反转	84
7.6 优先级继承协议	85
7.7 在 μ COS-II 上实现优先级继承协议	86
7.8 优先级天花板协议	88
7.9 在 μ COS-II 上扩展优先级天花板协议	91
习题	94
第 8 章 任务间通信	95
8.1 通信实现的基本数据结构	96

8.2	信号量——资源	101
8.3	消息队列	102
8.4	邮箱	107
8.5	管道	110
8.6	事件集	112
	习题	120
第 9 章	内存管理	121
9.1	内存保护	121
9.2	内存管理机制	122
9.2.1	固定大小存储区管理	122
9.2.2	可变大小存储区管理	123
9.2.3	μ COS 实现内存管理的方式	126
9.2.4	Linux 的内存机制	135
9.2.5	μ Clinux 下虚拟内存机制的屏蔽	137
	习题	137
第 10 章	中断与异步信号	138
10.1	中断	138
10.2	中断向量表	138
10.3	中断处理过程	140
10.3.1	中断检测	141
10.3.2	中断响应	145
10.3.3	中断处理	145
10.3.4	MicroBlaze 中断管理	147
10.3.5	μ COS-II 中断服务程序实现	152
10.4	时钟中断和时钟管理	154
10.5	软中断	158
10.6	异步信号系统	162
10.7	中断性能评价指标	163
	习题	166
第 11 章	Linux	167
11.1	Linux 内核	167
11.2	Linux 文件结构模型	169
11.3	Linux 驱动编写	177
11.4	Linux 驱动的编写实例——字符设备驱动	178
11.4.1	字符设备驱动原理分析	179
11.4.2	字符设备驱动数据结构	180

11.4.3	字符设备驱动的编写步骤	182
11.4.4	驱动程序编译与装载	186
第 12 章	μClinux	188
12.1	μClinux 内核	188
12.2	μClinux 移植	189
12.3	μClinux 驱动编写	189
第 13 章	Android	194
13.1	Android 系统构架	194
13.2	Android SDK 开发环境	196
13.3	Android 平台开发环境搭建	197
13.4	Android 蓝牙驱动编写	199
13.5	Android 平台蓝牙编程	202
第 14 章	Windows CE	204
14.1	内核与驱动	205
14.2	Windows CE 系统驱动简介	206
14.3	Windows CE 驱动程序实例	209
实验指导		213
实验一	移植 μCOS-II 到 FPGA 上	213
实验二	任务的建立与多任务的执行	220
实验三	系统中断机制的建立与使用	220
实验四	蓄水池综合实验	221

第 1 章 绪 论

无论是嵌入式操作系统还是通用操作系统都是夹在硬件平台和应用软件之间的一个重要层次，本身起到承上启下的作用，功能上完成对系统的资源管理。简而言之，它对上提供与硬件无关的接口，管理应用软件；对下提供各类硬件的驱动接口，管理系统硬件。然而嵌入式操作系统与通用操作系统是非常不同的，主要在于它所面对的嵌入式硬件平台与通用操作系统硬件平台不同。首先微处理器比通用操作系统的处理器性能有所降低，体系结构不同；其次，通用操作系统的硬件平台中外围设备都是标准的配置，容易实现统一的设计，而嵌入式外围设备各种各样，对嵌入式操作系统提出的跨平台挑战也更大。另外，嵌入式应用软件往往带有极强的实时性要求、容错处理和某些分布自治的特点，嵌入式操作系统也必须为其提供相应的软件架构支持。

嵌入式操作系统的位置如图 1.1 所示。嵌入式操作系统是在嵌入式硬件平台之上的，可为嵌入式应用软件提供接口、对嵌入式处理器和嵌入式外围设备等硬件资源进行管理的系统软件。其主要作用是对应用软件隐藏复杂的硬件驱动细节，从而简化软件开发。

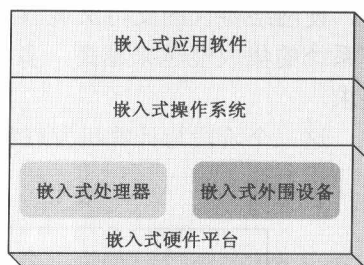


图 1.1 嵌入式操作系统的位置

1.1 嵌入式操作系统的发展历史

嵌入式操作系统的发展与嵌入式系统的发展类似，都是以嵌入式微处理器的发展为主线的。随着芯片的集成度越来越高，微处理器的性能越来越强，嵌入式操作系统也越来越复杂。从最初的嵌入式操作系统和应用程序开发的 10:90 完成功能比例到最近的 90:10 的比例（见图 1.2），可以看出人们期望嵌入式操作系统完成越来越多的功能，期望具体应用的开发工作越来越简单。换句话说，人们期望嵌入式操作系统越来越智能，而应用程序开发越来越傻瓜。

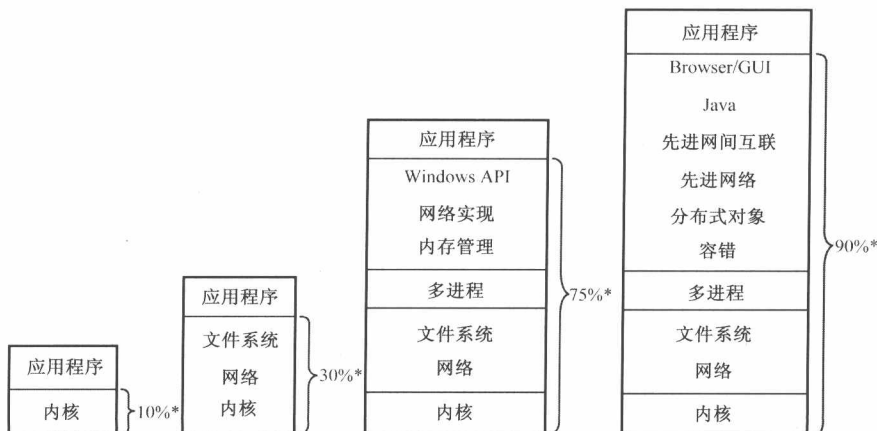


图 1.2 嵌入式操作系统在应用程序开发中所占的比例随时间的变化

嵌入式操作系统的发展经历了以下几个阶段。

第一阶段是无操作系统阶段。最初都是由应用开发人员自己编写所有的程序，没有专门的操作系统可以使用。

第二阶段是简单操作系统阶段。

(1) 一般操作系统阶段。随着微处理器的处理能力越来越强，集成了更多的存储空间，设备需要实现的功能越来越复杂，程序也越来越复杂，为了提高程序编写的效率，把一些常用的功能模块集成起来，就形成了最初的嵌入式操作系统。因此嵌入式操作系统一开始绝不是从通用操作系统划分出来的。

(2) 实时操作系统阶段。因为运行在电子设备上，并不需要用户的交互，所以实时性成为嵌入式操作系统的特殊需求。随着工业自动化的发展，实时嵌入式系统越来越多，使实时嵌入式操作系统成为一个独立的分支而发展出来。

第三阶段是复杂操作系统阶段。从互联网出现开始，各种设备开始添加到网络连接之中，使网络协议的支持又被放到嵌入式操作系统中，形成网络操作系统。SoC 技术的出现，使系统的集成性越来越高，多核出现在一个芯片封装中，从而对嵌入式操作系统也提出了新要求。

这三个阶段与计算机发展和嵌入式系统发展也是紧密结合的，具体见表 1.1。

表 1.1 计算机、嵌入式系统和嵌入式操作系统的发展

	计算机	嵌入式系统	嵌入式操作系统
第一阶段	20 世纪 50 年代，大型机	1960—1970，兴起	无操作系统
第二阶段	20 世纪 70 年代，个人计算机	1971—1989，繁荣	一般操作系统
			实时操作系统
第三阶段	现代，PC 无处不在	1990 至今，纵深发展	网络操作系统，多核操作系统

1.2 嵌入式操作系统的分类

可以在多种原则下对嵌入式操作系统进行分类。

按家族系统可以分成：基于 Windows 平台的，如 Embedded XP, Windows CE, Pocket Windows；风河系统的 VxWorks，遵从 POSIX 和 ANSI C 标准；Linux 家族的 Blue Cat Linux (www.linuxworks.com)，(Embedded) Red Hat Linux (www.redhat.com)，FSM RT-Linux (www.fsmlabs.com)，Monta Vista Linux (www.mvista.com)，TimeSys Linux (www.timesys.com)；LynxOS (www.linuxworks.com)；遵从 POSIX 标准的 QNX (www.qnx.com)；可实时扩展的 Solaris；日本的嵌入式操作系统标准 TRON 系列，如 eCOS。

根据嵌入式操作系统应用的不同领域，嵌入式操作系统也分为面向信息家电的嵌入式操作系统，如 Linux 和自主产权 Hopen OS；面向智能手机的嵌入式操作系统，如苹果的 iOS 和 Google 的 Android，以及 Microsoft 的 Windows Phone 7.0 等；面向汽车电子的嵌入式操作系统；面向工业控制的嵌入式操作系统等。这些不同的领域都有或将有自己的操作系统标准，如汽车电子的操作系统标准是 OSEK。

从实时性的角度来看，嵌入式操作系统可分为实时嵌入式操作系统：具有强实时特点，如 VxWorks、QNX、Nuclear、OSE、DeltaOS、 μ COS 及各种 ITRON OS 等；非实时嵌入式操作系统：一般只具有弱实时特点，如 Windows CE、版本众多的嵌入式 Linux、PalmOS 等。

从嵌入式系统的商业模式来分类，可以分为商用型和开源型。商用型的功能稳定、可靠，有完善的技术支持和售后服务，但是费用较高，包括开发费用和版税。开源型的一般是开放源码，只收服务费，没有版税，如 Embedded Linux, RTEMS, eCOS。

1.3 嵌入式操作系统的组成

目前的嵌入式操作系统一般由内核、嵌入式网络支持、嵌入式文件支持等组成。

内核是嵌入式操作系统的基础，也是必备的部分。它提供任务管理，内存管理，通信、同步和互斥机制，中断管理，时间管理及任务扩展等功能。内核还提供特定的应用编程接口，但目前没有统一的标准。

其中，任务管理是内核的核心部分，具有任务调度、创建任务、删除任务、挂起任务、解挂任务、设置任务优先级等功能。通用计算机的操作系统追求的是最大的吞吐率，为了达到最佳整体性能，其调度原则是公平，采用 Round-Robin 或可变优先级调度算法，调度时机主要以时间片为主驱动。而嵌入式操作系统多采用基于静态优先级的可抢占的调度，任务优先级是在运行前通过某种策略静态分配好的，一旦有优先级更高的任务就绪，就马上进行调度。

嵌入式操作系统的内存管理比较简单。通常不采用虚拟存储管理，而采用静态内存分配和动态内存分配（固定大小内存分配和可变大小内存分配）相结合的管理方式。有些内核利用 MMU 机制提供内存保护功能。一般不采用被通用操作系统广泛使用的虚拟内存的技术。

通信、同步和互斥机制提供任务间、任务与中断处理程序间的通信、同步和互斥功能。一般包括信号量、消息、事件、管道、异步信号和共享内存等功能。与通用操作系统不同的是，嵌入式操作系统需要解决在这些机制的使用中出现的优先级反转问题。

中断管理一般具有以下功能：安装中断服务程序；中断发生时，对中断现场进行保存，并且转到相应的服务程序上执行；中断退出前，对中断现场进行恢复；中断栈切换；中断退出时的任务调度。通过中断会为系统提供一些时间管理的功能，包括日历、延迟操作等。

由于目前的嵌入式系统大多数都需要网络支持，因此嵌入式操作系统中也往往容纳进了各种网络支持的模块，以实现对应应用编程隐藏复杂的网络协议细节，保证正确收发数据。例如，一般会采用静态分配技术，在网络初始化时就静态分配通信缓冲区，设置了专门的发送和接收缓冲（其大小一般小于或等于物理网络上的 MTU 值），从而确保了每次发送或接收时处理的数据不会超过 MTU 值，也就避免了数据处理任务的阻塞等待。

嵌入式文件系统相比之下较为简单，主要具有文件的存储、检索、更新等功能，一般不提供保护和加密等安全机制。它以系统调用和命令方式提供对文件的各种操作，主要包括：设置和修改对文件和目录的存取权限；提供建立、修改、改变、删除目录等服务；提供创建、打开、读、写、关闭、撤销文件等服务。

1.4 目前主流嵌入式操作系统概述

国际上用于信息电器的嵌入式操作系统约有 40 种。近年来，市场上非常流行的嵌入式操作系统（EOS）产品，包括：3Com 公司下属子公司的 Palm OS，全球占有份额达 50%；Microsoft 公司的 Windows CE，份额不过 29%。在美国市场，Palm OS 更以 80% 的占有率远超 Windows CE。开放源代码的 Linux 很适于做信息家电的开发。

Linux 家族由于其丰富的资源、网络的支持、遵从 POSIX 标准而被应用到嵌入式系统之中。为了应对嵌入式设备的不同微处理器结构、内存受限和实时性要求，Linux 家族为此也做了相应的改变。首先，利用尽可能的模块化来提高系统的可裁剪性和硬件的可扩展性。其次，采用一些新方法提高系统的实时性能，主要方法有限制实时任务和非实时任务的交互，如 LynxOS/Blue Cat Linux，RTLinux/RTAI；采用新核或者资源核的方式集成实时和非实时任务，如 TimeSys Linux, Monta Vista Linux，TimeSys Linux。

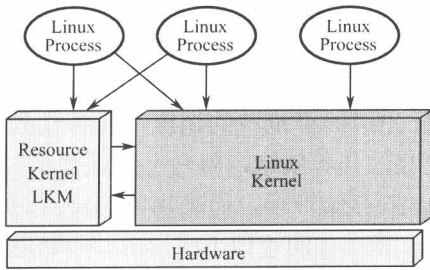


图 1.3 Linux 资源核的方式

如图 1.3 所示是 Linux 资源核的方式。在资源核里，所有任务都运行在虚拟资源上，利用对资源的控制来进行任务的调度，从而达到实时的方式。

Palm 是 3Com 公司的产品，其操作系统为 Palm OS。Palm OS 是一种 32 位的嵌入式操作系统。Palm OS 是一套专门为掌上电脑开发的 OS。在编写程序时，Palm OS 充分考虑了掌上电脑内存相对较小的情况，因此它只占有非常小的内存。由于基于 Palm OS 编写的应用程序占用的空间也非常小（通常只有

几十 KB），因此从个人管理、游戏到行业解决方案，Palm OS 无所不包。

Windows CE 是微软开发的一个开放的、可升级的 32 位嵌入式操作系统，是基于掌上电脑的电子设备操作系统，具有模块化、结构化和基于 Win32 应用程序接口以及与处理器无关等特点。Windows CE 中的 C 代表袖珍（Compact）、消费（Consumer）、通信能力（Connectivity）和伙伴（Companion），E 代表电子产品（Electronics）。与 Windows 95/98、Windows NT 不同的是，Windows CE 是所有源代码全部由微软自行开发的嵌入式新型操作系统，其操作界面虽来源于 Windows 95/98，但 Windows CE 是基于 Win32 API 重新开发的、新型的信息设备平台。它不仅继承了传统的 Windows 图形界面，而且继承了 Windows 95/98 上的编程工具（如 Visual Basic、Visual C++ 等），使绝大多数的应用软件只需简单修改和移植就可以在 Windows CE 平台上继续使用。Windows CE 的设计可以满足多种设备的需要，这些设备包括了工业控制器、通信集线器及销售终端之类的企业设备，还有照相机、电话和家用娱乐器材之类的消费产品。

Android 是手机专用的嵌入式操作系统，基于 Linux 内核的操作系统开发，由 Google 公司在 2007 年 11 月 5 日公布。早期由原名为“Android”的公司开发，Google 公司在 2005 年收购“Android.Inc”后，继续对 Android 系统进行开发运营。它采用了软件堆层（Software Stack，又名以软件叠层）的架构，主要分为三部分：底层以 Linux 内核工作为基础，由 C 语言开发，只提供基本功能；中间层包括函数库（Library）和虚拟机（Virtual Machine），由 C++ 开发；最

上层是各种应用软件，包括通话程序、短信程序等，应用软件则由各公司自行开发，以 Java 作为编写程序的一部分。它不存在任何以往阻碍移动产业创新的专有权障碍，号称是首个为移动终端打造的真正开放和完整的移动软件。

μ COS-II 由于其实时性不错、内核小、并行运行的特点，很受单片机及低端 ARM 用户的喜爱。包含内核的应用程序编译后可以到几 KB 的级别，非常适合内存空间受限、价格低的电子产品。同时，由于代码短小，结构性强，因此它非常适合初学者，被很多学校当成教学的素材。

eCOS 是由 Redhat 推出的小型即时操作系统（Real-Time Operating System），最低编译核心可小至 10KB 的级别，适用于 bootloader 增强及微小型系统。此系统和嵌入式 Linux 系统的差异是它将操作系统做成静态链接的方式，让应用程序通过链接（Linker）产生出具有操作系统的特性的应用程式。eCOS 的全称为 embedded Configuration Operating System，eCOS 是开放源码、免税费的实时操作系统，这套操作系统是针对嵌入式系统及应用而设计的，因此是以单一进程搭配多个线程的方式来执行的，提供了较多的元件和包供用户选择使用。

VxWorks 是美国 Wind River System 公司（即风河公司，简称 WRS 公司）推出的一个实时操作系统。Tornado 是 WRS 公司推出的一套实时操作系统开发环境，类似于 Microsoft Visual C，但是提供了更丰富的调试、仿真环境和工具，具有可裁剪的微内核结构、高效的任务管理、灵活的任务间通信、支持 TCP/IP 网络等优点。但是其价格昂贵，通常需要花费 10 万元人民币以上才可以搭建起一个可用的开发环境，每个应用还要另收版税。同时，也不提供源代码。

表 1.2 是几种主流实时嵌入式操作系统的性能比较。

表 1.2 几种主流实时嵌入式操作系统的性能比较

OS	VxWorks	μ COS-II	RT-Linux	QNX6
供应商	Wind River	Micrium	FSMLabs	Quantum
硬件平台	MC68000	80486/33MHz	80486/60MHz	80486/33MHz
任务切换	3.8 μ s	<9 μ s	不详	12.57 μ s
中断响应	<3 μ s	<7.5 μ s	25 μ s	7.54 μ s

从这几种嵌入式实时操作系统的比较可以看出，VxWorks 的实时性能最好， μ COS-II 的表现也相当不错，作为 BlackBerry 手机的新操作系统的 QNX6 表现也很好，而 RT-Linux 因为沿袭了 Linux 的设计理念，所以实时性能表现不佳。

1.5 嵌入式操作系统的发展趋势

随着信息化和各种网络平台的发展，嵌入式操作系统也越来越多地呈现出以下趋势。

(1) 网络化和云平台支持。随着网络化、云计算的发展，未来嵌入式操作系统需要支持云平台，也就是说不仅管理本地资源，还要能够分配和管理好网络、云端资源。

(2) 多核支持。随着 SoC 技术的继续发展，多核芯片逐步出现，目前操作系统如何支持多核化成为一个研究热点。首先，如何把顺序的程序分配给多核来处理成为嵌入式操作系统的首要任务；其次，SoC 多核带来的低能耗问题也将是操作系统考虑的重要问题，在某个核没有任务时，系统将可处于深度睡眠状态；再次，还需要提供给不同核一个抽象层，继承对一些原有

操作系统的支持；最后，当系统变得庞大时，还必须维护系统运行的高可靠性。

1.6 编写嵌入式操作系统的方法

要学习嵌入式操作系统原理，则必以编写为目的，立足于如何编写嵌入式操作系统将收到事半功倍的效果。编写嵌入式操作系统分成以下几步。

第一步必须明确目标。例如，C 语言当初的目的就是编写系统程序，很清楚什么是要的，什么是不要的，因此成功了。而 PL 语言，当时为了提供给 FORTRAN、COBOL、ALGOL 使用者一个标准的通用语言，因为目标不够明确，所以失败了。

编写一个操作系统要考虑以下几部分：

- 定义一个硬件抽象（Define Abstractions），把硬件和软件隔离开。为此必须了解所需要涵盖的不同微处理器的结构、存储器特征和支持 I/O 的具体种类。
- 提供基本操作（Provide Primitive Operations）。基本操作的主要功能是维持系统的数据结构，并保持一致。既要下面的硬件抽象负责，也要对上层的调用负责。可以说操作系统的核心就是下面的硬件抽象以及对基本操作的调用。
- 保证隔离（Ensure Isolation）。因为可以有多个用户同时登录或使用操作系统，或者多个任务同时运行，所以操作系统要保证它们在内存和上下文环境上的隔离。但是又要能让它们之间互用一些资源，因此完成时比较困难。
- 管理硬件（Manage Hardware）。当然操作系统是必须管理底层硬件的，如各种芯片和设备，尤其是要允许用户通过操作系统去管理一些常用的硬件设备，如 LCD 显示屏、串行接口等。

第二步需要了解编写操作系统的难点。

- 为了支持更多的功能，操作系统程序就越变越大，可是没有用户愿意花很长时间安装一个非常庞大的操作系统。
- 操作系统必须解决并行性，而资源的竞争、死锁等一系列问题都必须考虑。
- 操作系统必须要面对一些恶意用户，如连接网络时想远程破坏系统等。
- 许多设备希望能和别人分享一些资源。
- 操作系统的生命周期相对非常长，如 UNIX 等还一直在使用。
- 操作系统的设计往往面临一些新事物的挑战，如 Windows、UNIX 最初都没有设计 Email 支持。
- 操作系统必须面对各种硬件，即使对硬件不兼容也不能瘫痪。
- 必须兼容以前的系统。

第三步要了解编写嵌入式操作系统的基本原则。

原则 1：简单。

原则 2：完整。

原则 1 和原则 2 就是中国古人说的“增一分则肥，减一分则瘦”的道理。例如，MINIX 操作系统最初只设计了三个系统调用：send、recieve 和 sendrec。send 就是发送一个消息。recieve 就是接收信息。sendrec 是一个优化操作，为了方便在一个内核周期内完成发送并请求一个应答。Amoeba 操作系统只有一个系统调用 perform remote procedure call。从简单入手，然后再

从能够完整表述所有系统的范例来验证这种设计，从而不断完善，反复推敲，最后才能确定方案。

原则 3：效率。

如果一个系统调用不能用有效的代码完成，那么不如不要。所以对设计者的要求是必须知道每条代码的执行效率。例如，UNIX 编程者希望 `lseek` 要比 `read` 调用执行效率高，那么操作系统则必须满足这种效率来设计这两个系统调用。

第四步从接口设计开始。操作系统必须为硬件和用户提供一些处理数据结构和硬件操作的接口。尤其要为一些用户开放权利，让其可以插入自己的驱动。

第五步明确自己要采用的体系结构。体系结构划分详见 3.2 节。

通过以上 5 步，嵌入式操作系统的设计基本就成形了，剩下的就是实现的问题了。

习 题

1. 简述嵌入式操作系统的发展历史。
2. 嵌入式操作系统分为几类？
3. 编写嵌入式操作系统大体分为几步？基本原则是什么？

第 2 章 嵌入式操作系统的运行环境

嵌入式操作系统设计是从接口开始的，所以包括与下层硬件相结合的开发和与上层应用软件相结合的开发两部分。在编写具体的嵌入式操作应用之前，首先要做一定的平台移植工作，也就是根据硬件平台特性，修改嵌入式操作系统的一些配置文件，以及某些硬件平台相关的内部核心文件。通过这个移植工作，我们可以看到嵌入式操作系统设计时是如何定义及设计与硬件相关的接口部分的。下面就以 $\mu\text{COS- II}$ 移植到 FPGA 为例来讲解 $\mu\text{COS- II}$ 的硬件接口是如何设计与定义的。

2.1 运行平台介绍

了解运行硬件平台要从两个方面入手，一是其基本硬件特性和外围接口 I/O 的情况，二是微处理器的相关情况。了解外围接口是为了了解嵌入式操作系统中可能需要编写哪些外围设备驱动程序。了解基本硬件特性如内存大小，将了解嵌入式操作系统中存储管理的相关问题 and 是否适合运行嵌入式操作系统。以此为目，下面将首先介绍硬件平台 Spartan-6 的特性和外围接口。

Spartan-6 系列不仅拥有先进系统集成能力，同时还能实现适用于大批量应用的最低总成本。该系列由 13 个成员组成，可提供的密度从 3840 个逻辑单元到 147 443 个逻辑单元不等。与上一代 Spartan 系列相比，该系列功耗仅为其 50%，且速度更快，连接功能更丰富全面。Spartan-6 系列采用成熟的 45nm 低功耗铜制程技术制造，实现了性价比与功耗的完美平衡，能够提供全新且更高效的双寄存器 6 输入查找表 (LUT) 逻辑和一系列丰富的内置系统级模块，其中包括 18KB (2×9KB) Block RAM、第二代 DSP48A1 Slice、SDRAM 存储器控制器、增强型混合模式时钟管理模块、SelectI/O 技术、功率优化的高速串行收发器模块、PCI Express 兼容端点模块、高级系统级电源管理模式、自动检测配置选项，以及通过 AES 和 Device DNA 保护功能实现的增强型 IP 安全性。这些优异特性以前所未有的易用性为定制 ASIC 产品提供了低成本的可编程替代方案。Spartan-6 FPGA 可为大批量逻辑设计、以消费类为导向的 DSP 设计以及成本敏感型嵌入式应用提供最佳解决方案。

Spartan-6 系列产品的的主要技术特征见表 2.1。

表 2.1 Spartan-6 系列产品的的主要技术特征

元器件	逻辑单元	可配置逻辑模块 (CLB)			DSP48A1 Slice	Block RAM 模块		CMT 数量	存储器控制器模块 (最大)	PCI Express 端点模块数	最大 GTP 收发器数	总 I/O bank 数	最大用户 I/O 数
		Slice	触发器	最大分布式 RAM (KB)		18KB	最大 (KB)						
XC6SLX4	3840	600	4800	75	8	12	216	2	0	0	0	4	132
XC6SLX9	9152	1430	11440	90	16	32	576	2	2	0	0	4	200
XC6SLX16	14579	2278	18224	136	32	32	576	2	2	0	0	4	232
XC6SLX25	24051	3758	30064	229	38	52	936	2	2	0	0	4	266