



嵌入式系统设计与开发系列丛书

嵌入式操作系统原理与 面向任务程序设计

❖ 张 勇 编著



西安电子科技大学出版社

<http://www.xduph.com>

嵌入式系统设计与开发系列丛书

嵌入式操作系统原理与 面向任务程序设计

——基于 $\mu\text{C}/\text{OS-II}$ v2.86 和 ARM920T

Embedded Operation System Theory & Task Oriented Program Design

——Using $\mu\text{C}/\text{OS-II}$ v2.86 and ARM920T

张 勇 编著

西安电子科技大学出版社

内 容 简 介

本书基于 $\mu\text{C}/\text{OS-II}$ v2.86 和 ARM920T 芯片讲述嵌入式操作系统的工作原理以及面向任务应用程序设计方法, 阐述基于 $\mu\text{C}/\text{OS-II}$ 系统的用户应用程序的工作流程。全书共分八章, 主要内容包括嵌入式实时操作系统概述、嵌入式实时操作系统原理、 $\mu\text{C}/\text{OS-II}$ 内核、 $\mu\text{C}/\text{OS-II}$ 组件、 $\mu\text{C}/\text{OS-II}$ 应用实例、 $\mu\text{C}/\text{OS-II}$ 最小系统、面向任务程序设计(TOP)以及 TOP 设计实例等。本书的特色在于理论讲解透彻、实例丰富且针对性强。

本书是作者近几年来从事嵌入式系统教学与研究的成果结晶, 重点讲述 $\mu\text{C}/\text{OS-II}$ v2.86 原理与应用, 同时给出了裁剪的 $\mu\text{C}/\text{OS-II}$ 最小系统, 对学习嵌入式操作系统设计具有较强的指导作用。本书是作者已出版的《 $\mu\text{C}/\text{OS-II}$ 原理与 ARM 应用程序设计》(西安电子科技大学出版社 2010 年出版)一书的姊妹篇, 侧重于嵌入式操作系统工作原理与设计方法。

本书可作为电子通信、软件工程、自动控制、智能仪器等相关专业高年级本科生或研究生学习嵌入式操作系统的教材, 也可作为从事嵌入式应用和嵌入式操作系统开发的电子工程师、软件工程师以及嵌入式爱好者的参考书。

★本书配有电子教案, 需要者可登录出版社网站, 免费下载。

图书在版编目(CIP)数据

嵌入式操作系统原理与面向任务程序设计: 基于 $\mu\text{C}/\text{OS-II}$ v2.86 和 ARM920T / 张勇编著.

—西安: 西安电子科技大学出版社, 2010.12

(嵌入式系统设计与开发系列丛书)

ISBN 978-7-5606-2490-7

I. ① 嵌… II. ① 张… III. ① 实时操作系统—程序设计 ② 微处理器, ARM—程序设计

IV. ① TP316.2 ② TP332

中国版本图书馆 CIP 数据核字(2010)第 200745 号

策 划 李惠萍

责任编辑 雷鸿俊 李惠萍

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2010 年 12 月第 1 版 2010 年 12 月第 1 次印刷

开 本 787 毫米 $\times$ 1092 毫米 1/16 印 张 19.75

字 数 465 千字

印 数 1~3000 册

定 价 32.00 元

ISBN 978-7-5606-2490-7/TP·1240

XDUP 2782001-1

如有印装问题可调换

本社图书封面为激光防伪覆膜, 谨防盗版。



前 言

➤ 本书的结构与简介

与熟知的通用计算机系统相对的概念是专用集成电路系统。专用集成电路系统的特点在于其应用面向某些方面、存储空间相对较小且具有特定的外设,系统的核心为 ARM 芯片、DSP 芯片或 FPGA 等可编程芯片。随着人们对智能技术提出越来越高的要求,专用集成电路系统的软件设计越来越复杂,特别是基于 ARM 核心的专用集成电路系统,往往需要加载嵌入式操作系统,例如 Windows CE、嵌入式 Linux、VxWorks、eCos、 μ C/OS-II 等,然后在嵌入式操作系统的基础上设计用户应用程序。

嵌入式操作系统与通用 Windows XP(或 Windows 7)系统有较大的区别。一般地,可以认为嵌入式操作系统具有体积小、实时性强、可靠性高、功能可裁剪、系统可移植等特点。www.google.cn 上关于嵌入式操作系统的定义为:“为嵌入式计算机系统设计的操作系统,该操作系统被设计得非常紧凑和高效,舍弃了那些不会被用于专用场合下的非嵌入式计算机操作系统提供的函数,嵌入式操作系统往往是实时操作系统。例如,ATM、CCTV 系统、机顶盒、GPS、MP5 和机器人等设备上常使用嵌入式操作系统。”

本书重点讲述嵌入式操作系统的原理以及基于 ARM 芯片进行面向任务应用程序设计的方法。由于 μ C/OS-II 嵌入式实时系统是一款公开了源代码的中小型嵌入式操作系统,适合于教学和研究,故本书以讲解 μ C/OS-II v2.86 为主线。全书共分八章,第一、二章依次介绍嵌入式实时操作系统的概念和嵌入式实时操作系统的原理;第三至五章介绍 μ C/OS-II 内核、 μ C/OS-II 组件以及 μ C/OS-II 应用实例,全面介绍 μ C/OS-II 的设计原理和应用方法;第六章介绍裁剪了的 μ C/OS-II 最小系统及其应用实例,经裁剪后的 μ C/OS-II 最小系统代码少,仅支持信号量和消息邮箱,结构清晰,易于学习和掌握;第七、八章介绍面向任务程序设计(TOP)和实例,这两章内容基于 μ C/OS-II v2.86 系统进行阐述,进一步明确面向任务程序设计的特点和方法,并给出了具体的实例。

本书是笔者已出版的《 μ C/OS-II 原理与 ARM 应用程序设计》的姊妹篇,偏重于讲述嵌入式操作系统的工作原理,读者可以在 μ C/OS-II 最小系统的基础上,修改并扩展具有个人特色的功能,并与 μ C/OS-II 系统进行对比。作为从事嵌入式方面教学与科研的大学教师,笔者将会一直跟踪 μ C/OS-II 的升级和发展,并不断充实和修订本书。

➤ 本书的自学方法

本书可供嵌入式工程师自学。在自学过程中,读者需要同步阅读 ARM920T 和 Cortex-M3 相关的资料,特别是 S3C2410 和 STM32F103VB 等芯片的英文手册,从而对这些芯片的片上资源和外设有全面的了解。需要说明的是,书中的一些内容是需要熟记的。

本书已经将 $\mu\text{C}/\text{OS-II}$ 讲解得非常全面,但是仍然建议读者进一步参考 J J Labrosse 先生关于 $\mu\text{C}/\text{OS-II}$ 的原著,以及经常登录 www.micrium.com 网站进行学习。

最后,建议读者有一套 ARM 实验平台,自己动手,在实现本书提供的实例和练习的基础上,能独立地进行实例设计和实现工作,这样将会有有一个巨大的能力提升。

读者在自学过程中遇到的问题,特别是与本书实例和练习相关的问题,可以通过 QQ:493815991 和 E-mail:zhangyong@jxufe.edu.cn 与笔者沟通。

本书内容尽可能自成体系,由于篇幅有限,建议自学时参考笔者已出版的《ARM 原理与 C 程序设计》(西安电子科技大学出版社 2009 年出版)和《 $\mu\text{C}/\text{OS-II}$ 原理与 ARM 应用程序设计》,以及本书中提到的各种英文手册。

► 本书的教学思路

本书根据笔者的讲义修订而成,理论课时为 40 学时,实验课时为 24 学时,开放实验课时为 16 学时。如果本书用作大学本科教材,理论课时宜为 40 学时,建议讲述全部内容。当理论课时低于 30 学时,可选学第一至五章和第七、八章部分内容。实验课时建议为 20 学时。

理论学习过程中,建议第一至五章按顺序讲述,然后讲述第七、八章内容,最后讲述第六章内容。教学过程中可设置 2 学时讨论课,或将学生分组,做学习交流主题报告。可设置 4 个基础性实验和 1 个设计性实验,实验内容应涉及 $\mu\text{C}/\text{OS-II}$ 内核以及实验平台外设驱动等方面的内容,可参考第八章习题。

► 本书的特色

本书具有以下四个方面的特色:

其一,本书很细致地讲解了嵌入式操作系统的基本结构以及 $\mu\text{C}/\text{OS-II}$ v2.86 的原理;

其二,本书基于 $\mu\text{C}/\text{OS-II}$ 讲解了嵌入式操作系统在 ARM920T 内核上的应用,重点讲述了 S3C2410 芯片上加载嵌入式操作系统和应用程序的方法;

其三,本书裁剪了一个 $\mu\text{C}/\text{OS-II}$ 最小系统,通过阅读 $\mu\text{C}/\text{OS-II}$ 最小系统代码更容易领悟嵌入式操作系统的本质;

其四,本书提出了面向任务程序设计方法,结合实例讲述了这种设计方法的优点。

► 本书的配套源码

为了节省篇幅,书中有些程序没有给出完整的源代码,读者可以通过 E-mail:zhangyong@jxufe.edu.cn 向笔者免费索取,或登录西安电子科技大学出版社网站 <http://www.xduph.com> 下载。

► 特别说明

本书写作时,笔者所编写的《 $\mu\text{C}/\text{OS-II}$ 原理与 ARM 应用程序设计》一书还没有再版,该书中自第三章开始的工程文件中的 bsp.c 文件都存在小缺陷,即该文件中函数 OS_CPU_ExceptHndlr 中的“case 14:”下面应添加语句“OSIntEnter();”,在语句“OSTimeTick();”下面应添加语句“OSIntExit()”。不过,在给很多读者朋友发送的程序代码中笔者已修正了该缺陷。特别在此表示歉意!

➤ 致谢

感谢广州天嵌科技、北京博创科技、深圳英蓓特和上海爱亚软件公司提供硬件测试平台和软件开发环境。本书实例基于上述公司研发的平台，笔者相信这些实例也能适用于集成了相同 ARM 核心的其他通用实验平台。

同时，感谢那些阅读了笔者的书并反馈了宝贵意见的读者，这些意见对于笔者编写新书和修订再版已出版的书籍有实质性的帮助。由于笔者水平有限，本书中难免有纰漏之处，敬请同行专家和读者批评指正。

最后，感谢西安电子科技大学出版社李惠萍编辑和雷鸿俊编辑为本书出版所做的细致工作。

➤ 免责声明

本书内容仅用于教学研究，书中引用的相关内容的知识产权归相关公司所有，笔者保留其余内容的所有权利。禁止任何单位或个人摘抄和扩充本书内容用于出版发行。

张 勇

庚寅年于江西财经大学枫林园

目 录

第一章 嵌入式实时操作系统概述.....1	
1.1 操作系统的概念.....1	
1.2 嵌入式操作系统.....2	
1.3 嵌入式实时操作系统举例.....3	
1.3.1 Windows CE.....3	
1.3.2 VxWorks.....4	
1.3.3 嵌入式 Linux.....5	
1.3.4 Android 系统.....5	
1.4 $\mu\text{C}/\text{OS}-\text{II}$ 和 $\mu\text{C}/\text{OS}-\text{III}$ 的特点.....6	
1.4.1 $\mu\text{C}/\text{OS}-\text{II}$ 的特点.....6	
1.4.2 $\mu\text{C}/\text{OS}-\text{III}$ 的特点.....7	
1.5 小结.....8	
习题一.....8	
第二章 嵌入式实时操作系统原理.....9	
2.1 进程与线程.....9	
2.1.1 进程.....10	
2.1.2 线程.....11	
2.1.3 任务.....11	
2.2 任务调度与优先级.....13	
2.2.1 任务状态.....13	
2.2.2 任务优先级.....14	
2.2.3 任务切换.....14	
2.2.4 任务调度算法.....15	
2.2.5 中断与实时性.....18	
2.3 资源.....20	
2.3.1 共享资源.....20	
2.3.2 变量.....20	
2.3.3 可重入函数.....22	
2.3.4 死锁.....23	
2.4 内存管理.....24	
2.4.1 堆.....25	
2.4.2 栈.....25	
2.4.3 内存碎片.....25	
2.5 内核与时钟节拍.....26	
2.5.1 不可抢先型内核.....26	
2.5.2 可抢先型内核.....27	
2.5.3 时钟节拍.....28	
2.5.4 空闲任务.....29	
2.6 信号量与互斥信号量.....29	
2.6.1 信号量.....30	
2.6.2 互斥信号量.....31	
2.7 消息邮箱.....32	
2.8 小结.....33	
习题二.....33	
第三章 $\mu\text{C}/\text{OS}-\text{II}$ 内核.....34	
3.1 $\mu\text{C}/\text{OS}-\text{II}$ 初始化.....34	
3.2 空闲任务控制块链表.....39	
3.3 空闲事件控制块链表.....44	
3.4 空闲内存控制块链表.....47	
3.5 任务就绪组和任务就绪表.....50	
3.6 空闲任务.....56	
3.7 统计任务.....59	
3.8 定时器任务.....64	
3.9 空闲事件标志组链表.....70	
3.10 空闲消息队列链表.....74	
3.11 时钟节拍.....76	
3.12 任务状态.....81	
3.13 任务调度与内核函数.....83	
3.13.1 内核管理函数.....86	
3.13.2 延时管理函数.....91	
3.13.3 移植管理函数.....94	
3.14 多任务启动.....95	
3.15 小结.....96	
习题三.....97	

第四章 $\mu\text{C}/\text{OS-II}$ 组件	98
4.1 任务管理.....	98
4.1.1 创建任务.....	99
4.1.2 删除任务.....	110
4.1.3 堆栈检查.....	113
4.2 信号量管理.....	115
4.2.1 信号量使用方法.....	116
4.2.2 信号量创建函数.....	117
4.2.3 信号量请求函数.....	120
4.2.4 信号量释放函数.....	124
4.2.5 信号量赋值函数.....	128
4.3 互斥信号量管理.....	129
4.3.1 互斥信号量使用方法.....	130
4.3.2 互斥信号量创建函数.....	131
4.3.3 互斥信号量请求函数.....	133
4.3.4 互斥信号量释放函数.....	137
4.4 事件标志组管理.....	140
4.4.1 事件标志组使用方法.....	141
4.4.2 事件标志组创建函数.....	142
4.4.3 事件标志组请求函数.....	143
4.4.4 事件标志组释放函数.....	151
4.5 消息邮箱管理.....	155
4.5.1 消息邮箱使用方法.....	157
4.5.2 消息邮箱创建函数.....	157
4.5.3 消息邮箱请求函数.....	158
4.5.4 消息邮箱释放函数.....	161
4.6 消息队列管理.....	162
4.6.1 消息队列使用方法.....	164
4.6.2 消息队列创建函数.....	164
4.6.3 消息队列请求函数.....	166
4.6.4 消息队列释放函数.....	169
4.7 多事件请求管理.....	170
4.7.1 多事件请求函数使用方法.....	171
4.7.2 多事件请求函数工作原理.....	171
4.8 中断管理宏函数.....	180
4.9 定时器管理.....	181
4.9.1 定时器任务.....	182
4.9.2 定时器使用方法.....	186
4.9.3 定时器创建函数.....	186
4.9.4 定时器启动函数.....	189
4.9.5 定时器停止函数.....	191
4.9.6 定时器刷新函数.....	193
4.10 动态内存管理.....	194
4.10.1 动态内存使用方法.....	195
4.10.2 动态内存创建函数.....	195
4.10.3 动态内存请求函数.....	198
4.10.4 动态内存释放函数.....	199
4.11 小结.....	200
习题四.....	200
第五章 $\mu\text{C}/\text{OS-II}$ 应用实例	201
5.1 Borland C++ 5.02 与实例一.....	201
5.1.1 在 Borland C++ 上实现实例一.....	201
5.1.2 实例一程序解释.....	209
5.1.3 $\mu\text{C}/\text{OS-View}$ 与实例一.....	215
5.2 实例二.....	223
5.3 实例三.....	227
5.4 实例四.....	230
5.5 小结.....	233
习题五.....	233
第六章 $\mu\text{C}/\text{OS-II}$ 最小系统	234
6.1 内核裁剪.....	234
6.1.1 配置文件 os_cfg.h.....	235
6.1.2 最小系统头文件 ucos_ii.h.....	237
6.1.3 最小系统文件与执行流程.....	239
6.2 最小系统实例.....	240
6.3 小结.....	243
习题六.....	243
第七章 面向任务程序设计(TOP)	244
7.1 程序设计方法.....	245
7.2 任务与函数.....	247
7.3 任务构造方法.....	251
7.3.1 指示层任务设计.....	251
7.3.2 输入/输出层任务设计.....	254
7.3.3 计算层和输入/输出层任务 联合设计.....	256
7.4 任务优先级与堆栈.....	257
7.5 任务调度与切换.....	259
7.6 任务间同步与通信.....	260

7.7 任务挂起与恢复.....	261	8.4.3 数码管显示实例.....	286
7.8 小结.....	262	8.5 实例四.....	290
习题七.....	263	8.5.1 模/数变换驱动	290
第八章 TOP 设计实例	264	8.5.2 模/数变换实例	291
8.1 硬件平台.....	264	8.5.3 中值滤波实例.....	295
8.2 工程框架与实例一.....	267	8.6 小结.....	299
8.2.1 工程框架.....	267	习题八.....	299
8.2.2 LED 灯闪烁与实例一.....	273	附录 μC/OS-III 和 Cortex-M3	
8.3 实例二.....	276	简要说明.....	300
8.3.1 串口驱动.....	276	附录 1 μ C/OS-III 文件组织结构	300
8.3.2 串口通信实例.....	278	附录 2 Cortex-M3 处理器和 EMSTM32V100	
8.4 实例三.....	281	实验平台.....	301
8.4.1 数码管驱动.....	281	附录 3 μ C/OS-III 实例说明	303
8.4.2 数码管秒表实例.....	282	后记	305

第一章 嵌入式实时操作系统概述



本章从宏观角度介绍与嵌入式操作系统相关的概念,并对现有的应用广泛的嵌入式操作系统作简要说明,重点在于分析 $\mu\text{C}/\text{OS-II}$ 和 $\mu\text{C}/\text{OS-III}$ 嵌入式操作系统的特色。通过本章的阅读,读者特别是非计算机专业的读者,将对嵌入式操作系统有一个全面的认识。

1.1 操作系统的概念

操作系统在维基百科全书中的定义为:“操作系统是一类特殊的计算机程序,它管理着计算机硬件系统,且为用户应用程序提供服务。”这一概念容易被大多数人所接受,也基本上概括了操作系统的本质。而深入理解操作系统的概念,必须结合计算机操作系统的发展史。

众所周知,美国微软公司是全球最大的操作系统软件供应商。在单机操作系统软件方面,微软公司先后推出的有代表性的计算机操作系统有 MS-DOS、Windows 3.1、Windows 95、Windows 2000、Windows XP、Windows Vista 和 Windows 7 等,相信其还会不断推出新版本的视窗操作系统。

MS-DOS 是 Microsoft Disk Operating System 的缩写,即磁盘操作系统。在 20 世纪 90 年代以前的 IBM-PC 兼容机上普遍安装有 DOS 系统,该系统由一个 Boot 引导程序和三个文件模块组成,即输入/输出模块(IO.SYS)、文件管理模块(MSDOS.SYS)以及命令解释模块(COMMAND.COM)。用过 386 电脑的读者会知道,此时的 DOS 操作系统的作用为管理计算机的硬件资源(如处理器、存储器以及键盘和显示器等外设),人机交互界面为命令行模式。由于 DOS 提供的服务(或称函数)功能有限,当时人们流行设计汇编语言程序,直接对计算机硬件进行操作。更有意思的是,人们常常通过查看 COMMAND.COM 程序的字节大小来决定计算机系统是否感染病毒。

从 Windows 3.1 开始,微软公司成功地推出了视窗操作系统,此时的 Windows 3.1 是安装在 DOS 之上的,与 DOS 的重大区别在于 Windows 3.1 是多任务的图形用户界面操作系统。由于借助 DOS 管理硬件,从严格意义上讲,Windows 3.1 不直接作用于硬件资源,而偏重于内存管理、进程管理和图形界面等,而这些特性也隶属于操作系统的范畴。

目前,Windows 95、Windows 98、Windows Me 和 Windows 2000 的用户已经很少了,大部分 PC 机用户都在使用 Windows XP 和 Windows Vista 甚至 Windows 7 操作系统。Windows 95 至 Windows 7 均为分层的操作系统,包括与硬件相关的驱动层和与硬件无关的内核层,这些操作系统在文件系统管理、图形界面管理、进程和线程调度、内存管理以及

网络管理等方面具有一些共同点。借助 Windows 操作系统提供的各种服务,用户能直观高效地使用计算机资源。例如,用户想打印一份通知,可首先在 Office Word 应用软件中编辑通知(Word 是 Windows 系统之上的应用软件),编辑排版好后,点击 Word 软件中的打印菜单项,Word 软件调用 Windows 系统的打印功能(包括打印机驱动程序模块等),即可将通知打印出来。可见,用户直接接触和使用的软件为应用软件,而 Windows 系统软件则为“幕后英雄”。

在发布单机版 Windows 操作系统的同时,微软也针对服务器计算机发布了 Windows NT 等操作系统。

除了 Windows 视窗操作系统外,还有 Linux(或 UNIX)操作系统以及它们的变种系统,这些系统大多分为内核层和应用层,内核层具有进程管理、存储管理、设备管理和文件系统等功能,应用层直接为用户应用程序提供服务。

对传统操作系统的概念进一步革新的系统是分布式操作系统和网络操作系统。分布式操作系统强调多个计算机系统相结合,形成一个统一的操作系统。在分布式操作系统中,每个计算机系统是分布式操作系统的一个角色,应用软件是建立在分布式操作系统之上的,应用软件可以通过分布式操作系统访问每一个成员计算机系统的资源。相对于集中式操作系统而言,分布式操作系统在资源管理、系统结构和进程通信方面有重大区别。网络操作系统(WebOS)是建立在 Web 浏览器之上的操作系统,借助于网络操作系统可以使计算机资源网络化和计算网络化,基于 WebOS 的应用程序可称为网络应用程序。维基百科全书中关于网络操作系统的定义为:“一般地,网络操作系统指借助于 Web 浏览器而不依赖于本地操作系统的软件平台。”估计未来几年内可能会出现“云操作系统”,但是,无论哪种样式的操作系统,最终目的都在于方便用户实现各种功能的应用程序。

1.2 嵌入式操作系统

微软的 Windows XP 等多用户多任务操作系统在桌面计算机领域取得了巨大的成功,在智能设备和 PDA(Personal Digital Assistant)方面微软公司的 Windows CE 操作系统同样成就辉煌。相对于 Windows XP 而言,Windows CE 称为嵌入式操作系统,流行的 Windows Mobile 智能手机就是基于 Windows CE 嵌入式操作系统的。要深刻理解嵌入式操作系统的概念,需要首先明确嵌入式系统的概念。

根据维基百科全书的解释,嵌入式系统(Embedded System)是一种完全嵌入到设备内部、为特定应用而设计的专用计算机系统。嵌入式系统的内核往往是单片机、DSP、ARM 或 FPGA 等数字可编程芯片,然而,随着数字化芯片的技术水平不断提高,嵌入式系统的功能越来越强大,以至于嵌入式系统的概念变得模糊。例如,现在流行的一些 PDA 在多媒体处理、数据存储和网络通信等方面的功能比原来的通用计算机功能还要强大,但是 PDA 仍隶属于嵌入式系统。

除了 PDA 之外,常见的家电、医疗器械、运动装备、自动柜员机(ATM)、计算机网络设备、便携式 MP5 等都是嵌入式系统,它们的共同点在于:设计目的专一,实现的功能专一,硬件核心为可编程芯片,软件功能相对固定(称为固件)。例如,便携式心电图测试仪主要

用于观测病人心脏的长时间波动情况，其软件主要完成心电数据的采集、处理、显示和存储功能。

根据维基百科全书的解释，嵌入式操作系统是嵌入式系统的操作系统，通常被设计得非常紧凑有效，抛弃了运行在它们之上的特定应用程序所不需要的各种功能。互动百科全书则指出，嵌入式操作系统负责嵌入式系统的全部软、硬件资源的分配和调度工作，控制协调并发活动，且能通过装卸某些软件模块来达到系统所要求的功能。

嵌入式操作系统往往也是实时操作系统，常见的嵌入式操作系统有 Windows CE、嵌入式 Linux、VxWorks、 μ C/OS-II、eCos、QNX、Android、Symbian 等。

Labrosse 指出，实时系统是对逻辑和时序要求非常严格的系统，如果逻辑和时序出现偏差，将会引起严重后果。即实时系统是必须能在确定的时间内执行特定功能，并能对外部的异步事件做出响应的计算机系统，实时系统对响应时间有严格要求。实时多任务操作系统是指具有多任务调度和资源管理功能的实时系统，即所谓的嵌入式操作系统，它往往具有以下特点：

- (1) 实时性，即在确定的时间内执行特定功能和对中断做出响应。
- (2) 体积小，一般为几 KB 到几百 KB。
- (3) 可裁剪，即嵌入式操作系统采用模块化设计，可根据需要选择特定的功能模块。
- (4) 健壮性，即具有很好的运行稳定性。
- (5) 可移植性，即可以运行于多种嵌入式系统平台上。
- (6) 可固化性，即嵌入式操作系统可固化在嵌入式系统的 Flash 芯片内。
- (7) 提供设备驱动和应用程序接口，即用户可以借助嵌入式操作系统使用和管理系统资源。
- (8) 提供图形用户界面和网络功能。有些嵌入式操作系统提供了友好的图形用户界面(GUI)和网络支持。

1.3 嵌入式实时操作系统举例

在美国有几百个嵌入式实时操作系统，引进到我国的嵌入式操作系统有 Windows CE、VxWorks、嵌入式 Linux、Android 和 μ C/OS-II 等，这些系统在嵌入式应用领域占有绝对优势，下面集中介绍前四个嵌入式操作系统的特点，在第 1.4 节将介绍 μ C/OS-II 的特点。

1.3.1 Windows CE

Windows CE 中的 C 代表袖珍(Compact)、消费(Consumer)、互连(Connectivity)和伴侣(Companion)，而 E 代表电子产品(Electronics)。Windows CE 是一个可抢先式、多任务、多线程并具有强大通信能力的 32 位嵌入式操作系统，是微软为移动应用、信息设备、消费电子和各种嵌入式应用而设计的实时系统，其目标在于实现移动办公、便携娱乐和智能通信。

Windows CE 是模块化的操作系统，主要包括四个模块，即内核(Kernel)、文件子系统、图形窗口事件子系统(GWES)和通信模块。其中，内核负责进程和线程调度、中断处理、虚拟内存管理等；文件子系统管理文件操作、注册表和数据库等；图形窗口事件子系统包括

图形界面、图形设备驱动和图形显示 API 函数等；通信模块负责设备与 PC 间的互连和网络通信等。目前 Windows CE 的最高版本为 6.0。微软公司旗下的 Windows Mobile 智能手机操作系统均基于 Windows CE 设计，而且普遍使用的 Windows Mobile 5/6/6.1 和 Windows Phone 6.5 均基于 Windows CE 5，最新推出的 Windows Phone 7 智能手机操作系统则基于 Windows CE 6.0(自 2009 年底，微软将 Windows Mobile 更名为 Windows Phone，早期还使用过 Pocket PC 和 Smart Phone 等名称)。

Windows CE 支持四种处理器架构，即 x86、MIPS、ARM 和 SH4，同时支持多媒体设备、图形设备、存储设备、打印设备和网络设备等多种外设。除了在智能手机方面得到广泛的应用之外，Windows CE 也被应用于机器人、工业控制、导航仪、PDA、示波器等设备上。

相对于其他嵌入式实时操作系统而言，Windows CE 的优点在于：

(1) 具有美观的图形用户界面，而且该界面与桌面 Windows 系统一脉相承，使得操作直观简单。

(2) 开发基于 Windows CE 的应用程序相对简单，因为 Windows CE 的 API 函数集是桌面 Windows 系统 API 函数的子集，熟悉桌面 Windows 程序设计的程序员可以很快地掌握 Windows CE 应用程序设计方法，所以，Windows CE 应用程序的开发成本较低。

(3) Windows CE 的文件管理功能非常强大，支持桌面 Windows 系统下的 FAT、FAT32 等。

(4) Windows CE 的可移植性较好。

(5) Windows CE 下的设备驱动程序开发相对容易。

(6) Windows CE 的电源管理功能较好，主要体现在 Windows Phone 上。

(7) Windows CE 的进程管理和中断处理机制较好。

(8) Windows CE 支持桌面 Windows 系统的众多文件格式，例如 Word 和 Excel 等，这种兼容性方便了桌面 Windows 用户在 Windows CE 设备上处理文档和数据。

Windows CE 凭借上述突出的优点，在便携设备、信息家电、工业监控等领域得到了广泛的应用。

1.3.2 VxWorks

VxWorks 是一款真正意义上的嵌入式实时操作系统(RTOS)，是由专注于嵌入式和移动软件技术的美国风河(WindRiver)公司设计的，而且该公司在嵌入式 Linux 方面的研究成果也很丰富。VxWorks 系统可以用于多核处理器系统，具有极高的可靠性和安全性，风河多媒体库支持图形用户接口(GUI)开发。此外，VxWorks 在设备互连和网络通信方面也具有一定的优势。

VxWorks 有以下特点：

(1) 可靠性极高。VxWorks 通过了 DO-178B、ARINC 653 和 IEC 61508 等平台严格的安全性验证，因而它主要应用于军事、航空、航天等对安全性和实时性要求极高的场合。稳定性和可靠性高是 VxWorks 最受欢迎的特点。

(2) 实时性好。实时性是指能够在限定时间内执行完规定功能并对外部异步事件做出响应的能力。VxWorks 系统实时性极好，系统本身开销很小，进程调度、进程间通信、中断处理等系统程序精炼有效，造成的任务切换延时很短，提供了优先级抢先式和时间片轮换

方式多任务调度,使硬件系统发挥最好的实时性。例如,美国的 F-16 战斗机、B-2 隐形轰炸机和爱国者导弹,甚至 1997 年的火星探测器上也使用了 VxWorks 系统。

(3) 可裁剪性好。VxWorks 内核只有 8 KB 大小,其他的系统模块可根据需要定制,使得 VxWorks 系统具有灵活的可裁剪性能,既可用于极小型单片系统,也可用于大规模网络系统。VxWorks 的存储脚本(Memory Footprint)可以指定系统运行内存空间大小(这里的存储脚本可理解为基于 VxWorks 的应用程序可执行代码)。

(4) 开发环境友好。基于图形化集成开发环境 WindRiver Workbench(目前版本为 3.1),可开展基于 VxWorks 和 WindRiver Linux 系统应用的工程开发。WindRiver Workbench 是一个完备的设计、调试、仿真和工程集成解决方案。

1.3.3 嵌入式 Linux

嵌入式 Linux 是嵌入式系统领域最重要的实时操作系统,是几乎所有涉足嵌入式领域的人士必学的嵌入式操作系统。

嵌入式 Linux 是对流行的 Linux 操作系统进行裁剪和修改,使之能应用于嵌入式计算机系统的一种操作系统,其实时性、稳定性和安全性均较好,在通信电子、工业控制、信息家电、仪器仪表方面应用广泛。

嵌入式 Linux 具有以下特点:

(1) 嵌入式 Linux 是完全开源的,因此它广泛用于高校教学。研究嵌入式 Linux 代码的专家、学者远比其他嵌入式操作系统都多,而且 Internet 上的资源丰富,也有大量的图书、资料,使得学习 Linux 系统的代价最小。

(2) 嵌入式 Linux 是免费的,不涉及任何版权和专利,这一点被商界所看重。因此,大部分嵌入式产品在研发初期都使用过嵌入式 Linux 版本。嵌入式 Linux 被很多团体和组织二次开发后,形成具有独立知识产权的嵌入式操作系统,所以,嵌入式 Linux 变种系统非常多,如 WindRiver Linux 和 μ CLinux 等。

(3) 嵌入式 Linux 与 QT 相结合,使嵌入式 Linux 具有良好的图形人机界面,甚至可以和 Windows CE 相媲美,而且 QT 目前也是开源的。

(4) 嵌入式 Linux 的移植性强,其变种形式几乎可应用于所有主流嵌入式系统中。嵌入式 Linux 对外设驱动能力很强,驱动接口程序设计相对容易,网络上有大量常用设备的驱动代码可供参考借鉴。

(5) 嵌入式 Linux 在内核、文件系统、网络支持等方面均有突出的特点。2007 年 2 月发布的 Linux 2.6.2x 是最新的 Linux 操作系统,具有约 201 万行源代码,内核可支持 32 个 CPU,实时性显著提高(但严格意义上不是实时操作系统),采用了更有效的任务调度器,增加了对多种嵌入式处理器的支持,在多媒体方面也有很大提高。

1.3.4 Android 系统

Google 的 Android 系统是 Windows Phone 的主要竞争对手。

Android 系统基于 Linux 系统,是 Google 在 2005 年并购 Danger 公司后发展他们的 Android 计划的成果。Andy Rubin 是这个计划的负责人,主要针对智能手持设备。Android 的运行库文件只有 250 KB,最基本配置为 32 MB 内存、32 MB 闪存和 200 MHz 处理器。

Windows Phone 对硬件配置的要求是固定的,智能设备生产商必须按 Windows Phone 的要求进行硬件设计,否则不能运行 Windows Phone。相比之下,Android 系统更灵活多样。

作为嵌入式操作系统而言,比较 Android 系统和 Windows Phone 系统的意义不大,因为它们都实现了对硬件资源的抽象和美观的图形用户界面,并且 Android 系统是开源的。但是 Android 系统也可被视为一个应用系统,其集成的一些软件的附加值相当高。例如,Google 地图以及与 Google 地图相关的生活关爱软件能从根本上为人们节省时间并改善人们的生活。此外,多媒体娱乐软件和基于云计算和网络服务的软件也相当出色,这些是 Android 系统的独特优势。

开发 Android 系统应用程序与开发 Windows Phone 应用程序相似,可基于其 SDK 包和集成开发环境实现。就目前来说,相对于 Windows Phone 系统,Android 系统还没有明显的劣势。

1.4 μ C/OS- II 和 μ C/OS-III 的特点

μ C/OS- II 是一个完整、可移植、固化、裁剪的抢先式实时多任务操作系统,最终版本为 2.86,源码完全公开,约有 1 万多行。相对于庞大的嵌入式 Linux 而言,小巧的 μ C/OS- II 更适合教学,因此,目前很多国家的高等院校指定使用 μ C/OS- II 进行嵌入式操作系统教学。2010 年 Micrium 公司推出了 μ C/OS-III,由于其没有公开源码,一定程度上限制了其教学推广,但是 μ C/OS-III 加入了一些新的特性,在商界备受欢迎。

1.4.1 μ C/OS- II 的特点

μ C/OS- II 公开全部源代码,大约有 1.1 万行代码,这些源代码是由 Labrosse 一个人写成的,逻辑性很强,他为全部代码添加了详细的注释,并且这些代码的结构合理,格式清晰,很方便阅读和学习。Labrosse 先后出版了三本书介绍 μ C/OS- II,使得 μ C/OS- II 迅速在全球流行起来。在我国, μ C/OS- II 的地位几乎超越了其他嵌入式操作系统,成为家喻户晓的首选系统。

μ C/OS- II 具有以下特点:

(1) μ C/OS- II 具有优秀的可移植特性。 μ C/OS- II 的绝大部分源代码由 C 语言写成,只有小部分与处理器相关的代码使用汇编语言编写,汇编语言代码量压缩到最低限度。一般可认为支持 CPU 堆栈操作指令的所有微处理器,均可以移植 μ C/OS- II,因此,现在流行的单片机、DSP、ARM、FPGA 等芯片均可移植 μ C/OS- II,这使得 μ C/OS- II 系统的应用领域十分广阔。Micrium 网站上有大量可供参考的移植范例,移植工作可在几小时到一周时间内完成。

(2) μ C/OS- II 系统可固化在嵌入式系统的 Flash 中。由于 μ C/OS- II 是开源代码的,因此往往被添加到用户应用程序工程文件中,被统一编译和连接为可执行目标文件,该目标文件可被固化到 ROM 存储器或 Flash 芯片中。

(3) μ C/OS- II 系统可裁剪。通过 μ C/OS- II 系统的 OS_CFG.H 配置文件可以有选择地使用 μ C/OS- II 系统功能组件, μ C/OS- II 的可裁剪性是靠条件编译实现的。应根据实际的嵌入

式系统的存储空间大小和实现的功能, 选择 $\mu\text{C}/\text{OS-II}$ 系统的裁剪情况。

(4) $\mu\text{C}/\text{OS-II}$ 系统是可抢先型的实时内核, 即 $\mu\text{C}/\text{OS-II}$ 总是执行所有处于就绪状态下优先级最高的任务。 $\mu\text{C}/\text{OS-II}$ v2.86 最多支持 255 个任务, 并且各个任务的优先级不能相同, 即 $\mu\text{C}/\text{OS-II}$ 不支持同优先级任务间的调度。基于 $\mu\text{C}/\text{OS-II}$ 系统的应用程序由多个任务组成, 每个任务具有独立的堆栈空间, 并且允许其堆栈空间大小不同。 $\mu\text{C}/\text{OS-II}$ 系统可对堆栈大小和使用情况作动态检测。

(5) $\mu\text{C}/\text{OS-II}$ 系统提供了信号量、互斥信号量、事件标志组、消息邮箱、消息队列等多种服务组件, 提供了用于时间管理和内存管理的函数, 使用这些组件可方便地在任务间进行通信和同步。 $\mu\text{C}/\text{OS-II}$ 系统服务的执行时间是确定的, 即调用和执行 $\mu\text{C}/\text{OS-II}$ 系统函数的时间是确定的, 对于中断延时的时间也几乎是确定的。

(6) $\mu\text{C}/\text{OS-II}$ 系统具有很高的安全性和可靠性。2000 年 7 月, $\mu\text{C}/\text{OS-II}$ 取得了美国联邦航空管理局(FAA)关于 RTCA DO-178B 标准的质量认证, 表明 $\mu\text{C}/\text{OS-II}$ 系统可用于与人的生命攸关的、安全性要求苛刻的嵌入式系统中, 从而大大提升了 $\mu\text{C}/\text{OS-II}$ 系统的知名度。实际上, 目前国内外有大量的商业应用基于 $\mu\text{C}/\text{OS-II}$ 系统。

1.4.2 $\mu\text{C}/\text{OS-III}$ 的特点

$\mu\text{C}/\text{OS-III}$ 是 Micrium 公司最新的嵌入式实时操作系统(RTOS), 基于 $\mu\text{C}/\text{OS-II}$ 而添加了很多新的特性, 主要如下:

(1) $\mu\text{C}/\text{OS-III}$ 支持 ARM7/9、Cortex-Mx、Nios-II、PowerPC、Coldfire、Microblaze、SHx、M16C、M32C 和 Blackfin 等微处理器。 $\mu\text{C}/\text{OS-III}$ 支持无限多个任务, 支持时间片轮转调度, 不同任务的优先级可以相同, 优先级取值不受限制。一般地, 嵌入式系统应用程序只需配置 32~256 个任务即可满足要求。

(2) 由于 $\mu\text{C}/\text{OS-III}$ 的任务个数不受限制, 与任务相关联的信号量、互斥信号量、事件标志组、消息队列、定时器、内存分区等的个数也不受限制, 并且 $\mu\text{C}/\text{OS-III}$ 允许受监视的任务堆栈的大小扩展, 在使用时需要指定堆栈的安全大小, 当堆栈使用的空间超过安全大小时向系统报警, 这样可有效地保护应用程序而不至于因堆栈访问越界而瘫痪。

(3) $\mu\text{C}/\text{OS-III}$ 支持多个任务具有相同的优先级, 当相同优先级的几个任务同时就绪时, $\mu\text{C}/\text{OS-III}$ 为每个任务分配用户指定的 CPU 时间片, 每个任务可定义它自己的时间片。这种时间片轮转调度方式可以有效地解决多任务同一个优先级的的问题。

(4) $\mu\text{C}/\text{OS-III}$ 对中断响应时间是确定的, 进入临界区的转换时间(即关中断时间)几乎为 0 个时钟周期。通过 $\mu\text{C}/\text{OS-III}$ 配置文件可对 $\mu\text{C}/\text{OS-III}$ 系统进行裁剪, 对特定的应用而言, 只保留那些需要的特性和服务。绝大多数 $\mu\text{C}/\text{OS-III}$ 系统服务的时间是确定的常量, 这些系统服务(或称系统函数)运行的时间与应用程序中的任务数量无关。

(5) 没有裁剪的 $\mu\text{C}/\text{OS-III}$ 系统大约 24 KB, 最小配置的 $\mu\text{C}/\text{OS-III}$ 系统只有 6 KB, 内核服务组件有 10 类, 相关的应用程序接口(API)函数有 80 个。

综上所述, $\mu\text{C}/\text{OS-III}$ 可应用于通信设备、数码家电、智能电话、PDA、工业控制、消费娱乐电子、汽车电子以及大多数嵌入式系统中。相对于 $\mu\text{C}/\text{OS-II}$, 其明显的改进在于采用时间片轮转调度方法(Round-Robin Scheduling), 允许相同优先级的多个任务并存, 任务数可为无限个。

1.5 小 结

本章介绍了嵌入式操作系统的概念，重点介绍了当前流行的嵌入式操作系统的特点。随着嵌入式系统包容的范畴越来越广，随之嵌入式操作系统的概念也在不断升华，很难给嵌入式系统以及加载在其上的嵌入式操作系统下一个完整而准确的定义。一些高性能的嵌入式系统，不仅能加载嵌入式操作系统，而且可以加载桌面 Windows 系统，除了其专用功能特别显著外，其附加的多种通用功能也十分强大，以至于与通用计算机的界线越来越模糊。本书后续内容重点基于 ARM 平台和 $\mu\text{C}/\text{OS-II}$ 系统，是公认的典型嵌入式系统和嵌入式操作系统。

习 题 一

1. 简述嵌入式操作系统的概念，并结合实际列举嵌入式操作系统的应用实例。
2. 列举 $\mu\text{C}/\text{OS-II}$ 实时操作系统的特点。
3. 查阅相关的资料，分析 Android 系统与 Linux 系统的关系。
4. 查阅相关的资料，详细分析最新的 $\mu\text{C}/\text{OS-III}$ 比 $\mu\text{C}/\text{OS-II}$ 系统在哪些方面作了重大改进？