Broadview
www.broadview.com.cn

嵌入式实时操作系统

μC/OS原理与实践

卢有亮 编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

嵌入式实时操作系统

μC/OS原理与实践

卢有亮 编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书内容包括：实时操作系统基础、任务管理、中断和时间管理、事件管理、消息管理、内存管理、移植等。本书内容翔实，图文并茂，采用逐步深入、反复印证的方法，从数据结构的设计入手，再到代码分析、示例验证的剖析方法，逐层深入讲解，给出在虚拟平台下的移植示例和针对各章内容示例，并给出了基于 NIOS II 的 FPGA 系统上移植的例子。

本书适用于计算机、电子、通信、自动化及相关专业大学本科、研究生，也适用于广大嵌入式开发工程技术人员、电子技术研究人员、操作系统研究人员。

图书在版编目（CIP）数据

嵌入式实时操作系统 μC/OS 原理与实践 / 卢有亮编著. —北京：电子工业出版社，2012.2
ISBN 978-7-121-15441-6

Ⅰ. ①嵌… Ⅱ. ①卢… Ⅲ. ①实时操作系统 Ⅳ. ①TP316.2

中国版本图书馆 CIP 数据核字（2011）第 255180 号

策划编辑：张月萍
责任编辑：付　睿
特约编辑：赵树刚
印　　刷：涿州市京南印刷厂
装　　订：涿州市桃园装订有限公司
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：787×980　1/16　印张：18.25　字数：384 千字
印　　次：2012 年 2 月第 1 次印刷
印　　数：3500 册　　定价：39.00 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 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。

序　言

PREFACE

随着电子技术及信息技术的发展，嵌入式技术已广泛应用于工业、民用、军事、能源、环保等领域，是当前热门也很有发展前途的技术。大到航空航天、汽车电子、电力系统、石油化工系统等，小到智能手机、GPS、MP3 等都离不开嵌入式系统。要高效地开发嵌入式系统，就离不开嵌入式操作系统。一方面，高实时性的操作系统软件是嵌入式软件的基本要求；另一方面，嵌入式软件开发要想走向标准化，就必须使用多任务的操作系统。

我们身边的电子设备，如智能冰箱，考虑到成本和体积，控制电路的核心或者是单片机、DSP 或 ARM，或者是有软核 FPGA 或其他嵌入式系统。在嵌入式的环境下，如果做多任务的系统，就离不开嵌入式操作系统。例如，所谓的智能手机，就是具有嵌入式操作系统的手机。进一步，如果我们的嵌入式系统运用到火箭、飞机等设备，就不能不考虑实时性的要求，也就是任务要在规定的时间内完成，这个时间可以是毫秒级或微秒级，取决于具体的需要，因此必须采用实时性很强的操作系统。

μC/OS 正是这样的操作系统，它是高实时性、多任务的操作系统，是源代码对非商业用途开放的操作系统。μC/OS 适用于嵌入式开发，并已在各领域得到广泛的应用且获得认可。在百度上搜索 μC/OS，得到的信息条数约 125 万条。在 www.micrium.com 网站就可以获得内核源代码及 TCP、GUI 等应用的源代码。本书对最新版本的 μC/OSII-2.91 进行深入剖析，从实时操作系统的原理开始，对重要的数据结构和代码进行解析，采用从原理到数据结构和代码解析，再给出实例以印证和加强的方法，以便于读者掌握。

各专业读者学习方向不同，有些已经学习过操作系统原理，对一般的操作系统原理有一定的认识，有些电子专业的读者并没有深入学习过操作系统原理，另外有些读者对 C 语言掌握的程度并不太高，因此本书采用由浅入深，层层推进的方法，在第 1 章介绍实时操作系统的基本原来，为以后各章的学习打下基础，满足未学习过操作系统只是的读者学习，而学习过操作系统的读者也要掌握本章的内容，因为嵌入式实时操作系统有其个性。第 2 章任务管理部分详尽地讲解了代码方面的内容。从第 3 章介绍中断和时间管理，因为读者已经具备了或提高了 C 语言的能力，对读者已经掌握的内容就不再赘述，但对新的难点仍给出详尽解析。

实践是检验真理的标准，所学的知识必须结合实际。除了基于 NIOS II 的 FPGA 系统上

移植的例子外，前面各章例子都是通过将 μC/OS 移植到 PC 中，在虚拟的平台上运行 μC/OS 来得到印证。所有代码都包含在 VC 的一个工程中，打开工程文件就可以看到结构清晰的源代码，因此非常适合读者学习。

通过本书的学习，读者应能较为全面、深入地掌握嵌入式实时操作系统的基本原理，能够读懂读通 μC/OS，包括内核、任务管理、时间和中断管理、信号量管理、互斥信号量管理、事件标志组管理、消息邮箱管理、消息队列管理、内存管理的源代码，能将 μC/OS 移植到不同平台，并能在移植的过程中知其然又知其所以然，不会出现在没有掌握 μC/OS 而强做移植时出现的困难和错误。

因此，结合笔者嵌入式操作系统的教学及实践经验，本书适合一般层次相关技术人员及相关专业大学生、研究生学习，本书具有以下特点。

1．采用逐步深入，反复印证的方法

首先给出 μC/OS 涉及的基本原理，对不涉及的操作系统原理部分不做论述。原理部分在第 1 章给出，但在后续章节中分块详细论述。先有概念，再加深理解，循序渐进，避免读者读死书。

2．采用从数据结构的设计入手，再到代码分析、示例验证的剖析方法。给出在虚拟平台下的移植示例和针对各章内容的示例

对 μC/OS 的核心部分分章论述，每一部分先给出关键的数据结构，说明相关的宏定义，然后结合原理分析核心代码，并给出示例。这一部分的例子是将 μC/OS 移植到 PC 中，而使用 VC6 作为编译环境，这样读者能够非常方便地看到运行结果。读者在学完一章后，一方面将涉及的原理完全吃透，一方面对关键的代码能读懂读通，在做操作系统的移植的时候才能胸有成竹，并能灵活应用。

3．给出在基于 NIOS II 的 FPGA 系统上移植的例子

嵌入式操作系统最终要移植到嵌入式平台。如今 FPGA 作为可编程逻辑器件的主要产品，在通信、消费电子、汽车电子、工业控制等领域得到迅速应用，不断向 ASIC、ASSP、DSP 和嵌入式产品的传统领域渗透，且通过嵌入处理器核 NIOS 取代 MCU，FPGA 未来发展空间更加广阔。而在基于 NIOS II 的 FPGA 上移植 μC/OS 的例子却不多见，而且如果掌握了在基于 NIOS II 的 FPGA 系统上移植 μC/OS 的技术，就能够很容易地做到在其他嵌入式平台进行移植。

4．表格、图形化的风格

在对重要的数据结构、原理等的论述，以及代码的分析中，尽可能地采用图形化的描述来说明问题，使读者更容易掌握。例如空闲链表，用图形化的描述无疑使读者一目了然。

5．适用面广

本书适用于计算机、电子、通信、自动化及相关专业大学本科、研究生，也适用于广大嵌入式开发工程技术人员、电子技术研究人员、操作系统研究人员。

6．对于没有学习过操作系统原理的读者，完全没有障碍

本书的第 1 章就是原理性的描述，是有针对性而精炼的，不需要再去刻意读一本操作系统原理的教材。

学习操作系统，掌握的不只是操作系统。学习 μC/OS 后，不光掌握了 μC/OS，能够进行移植乃至修改，更重要的是对开发设计的思想有一个质的提升，并极大地提高 C 语言开发水平。本书对计算机专业和非计算机专业读者同样适用，对没有学过操作系统而可能从事嵌入式开发电子类技术人员更有益处。通过本书的学习，以及实践，可以对所学知识融会贯通，应用到实际的项目当中。

作为电子科技大学的一名普通教师，平时除教学外也要从事科研任务，理解操作系统对于电子、通信、自动化及计算机等领域的重要性。笔者认为学好一门嵌入式操作系统对知识层次的提升、工程应用技能的提升，乃至择业都有不小的好处。

本书中第 1 章是操作系统和嵌入式实时操作系统的基本原理，对于对原理掌握已经比较好的有一定基础的读者可以选择跳过或略读。第 2 章是任务管理，是基本内容，需要对数据结构和源代码仔细体会。第 3 章是中断和事件管理，时间管理包括一些函数，中断管理主要是中断的流程。第 4 章、第 5 章是事件和消息部分，包含了各种事件和消息机制。第 6 章是内存管理。第 7 章是移植的流程分析和在虚拟平台及具有软核的 FPGA 下的移植。

学习本课程的先前知识是 C 语言、软件技术基础或数据结构，可以同步学习微机原理。

另外欢迎各位老师将本书选作教材。如果作为教材，建议第 4、5、6 章的内容每章选择 2~3 节讲解。本书在每章后提供了习题，另外笔者也编写了 PPT，适合 32~48 学时对高年级本科生或低年级研究生讲授。

为方便读者学习，在电子工业出版社博文视点公司网站提供了可在 VC 下编译的完整源代码，包括例子程序的代码及相关 PPT，以用于教学或学习目的，另外也可通过邮箱 luyl@uestc.edu.cn 联系笔者，读者还可以到 www.micrium.com 网站下载 μC/OS 操作系统的源

代码及查找相关资料。

在编写的过程中，得到了电子科技大学，尤其是能源科学与工程学院、自动化工程学院的大力支持，另外也得到电子工业出版社的大力支持，在这里一并表示感谢。

计算机技术发展迅速，个人的力量和水平有限，在本书的编写中难免有不妥之处，欢迎广大读者批评指正。

卢有亮

2011 年于电子科技大学

目 录 Contents

第 1 章　实时操作系统基础

嵌入式实时操作系统是计算机技术和电子技术交叉的技术，无论是电子专业或计算机专业，还是其他专业的读者，只要有一定的计算机和嵌入式系统的基础，通过本书的学习都可以掌握 μC/OS-II 这一流行的嵌入式实时操作系统。

在工作中，嵌入式系统的开发者不得不与包含了大量程序代码和调度策略的实时操作系统打交道。因此，本章逐步给出实时操作系统的概念，帮助读者以最快的方式学习这一部分。如果对这些概念都很熟悉，可以跳过。如果在阅读的过程中感觉这一部分太抽象，也没有问题，可以快速读过。学习是一个循序渐进的过程，通过第 1 章的学习为后面章节的学习打一个基础。在学习过后面章节的内容后，第 1 章的内容也就融会贯通了。

实时操作系统一般都用于嵌入式的开发平台，如 ARM、DSP、基于软核的 FPGA，甚至是单片机。但是本着学习的目的，先在普通 PC 的 Windows XP 环境下对操作系统进行代码移植，通过 VC++ 对整个操作系统工程进行编译，编译后的可执行代码能在 Windows 平台下仿真运行，并可编译成可调试的代码以便单步调试和跟踪，这对于研究操作系统的代码、编写和验证例子程序、加深学习效果有极大的帮助！因此在本书的前几章，所给出的例子都是在这个环境下完成的，并试验通过。通过网站还可以下载到所有的源代码。

我们研究的 μC/OS-II 是嵌入式实时操作系统，是为嵌入式环境提供的实时操作系统，因此我们在以后的部分省去嵌入式，直接用实时操作系统来代替，不影响其意义。下面进入本章的第一节，操作系统的概述部分。

1.1　操作系统概述

本节的内容是关于操作系统的一些最基本的概念。

1.1.1 什么是操作系统

操作系统 OS（Operating System）是裸机上的第一层软件。操作系统是计算机系统中最重要的系统软件，是硬件的第一层封装与抽象，在计算机系统中占据着重要的地位，所有其他的系统软件与应用软件都依赖于操作系统的支持与服务。除提供编程接口外，操作系统还承担着任务管理、事件管理和消息通信、CPU 管理、内存管理、I/O 管理等核心功能。

如图 1.1 所示，在有操作系统的系统中，应用程序（Application）并不直接和硬件打交道，而是通过操作系统来和硬件打交道。操作系统直接运行在硬件平台之上，是裸机上的第一层软件，提供给用户编程接口。应用程序编程接口（API）可以解释为是一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件的以访问一组例程的能力，而又无须访问源码或理解内部工作机制的细节。

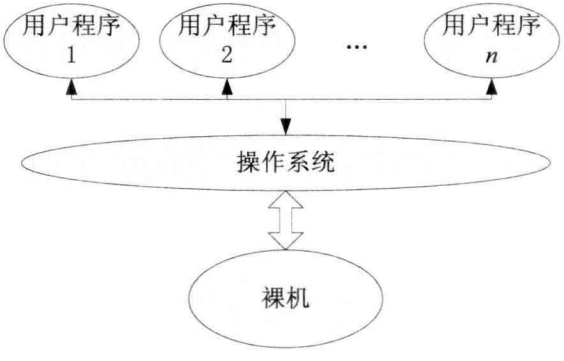

图 1.1　操作系统的位置

因此，从开发人员的角度讲，它提供了应用程序的接口 API，我们通过调用该 API 来管理任务，进行消息通信及访问硬件，如打印机、显示器、硬盘等。访问硬件不仅包括写操作，即给硬件发送指令、发送数据等，也包含读操作，例如，读取打印机是否空闲，如空闲才能发送进一步的指令等。

如程序 1.1 所示为最简单的 C 语言代码通过操作系统访问硬件。

程序 1.1　屏幕输出 Hello World!的 C 语言程序

```
int main(int argc, char* argv[])
{
    printf("Hello World!\n");                                    (1)
    return 0;
}
```

代码中（1）调用 printf 实现屏幕打印，不需要关心打印的细节。但是如果离开了操作系统，完成屏幕打印是比较困难和复杂的事情，首先要掌握硬件，然后编写底层程序驱动硬件。

所以，本例说明操作系统提供了应用程序的接口，有了操作系统，就不用为这些底层的编程烦恼了。

换一个角度，从系统的使用人员，即最终用户的角度来看，操作系统是一台虚拟机器，在其上运行和操作用户软件。例如，一个财务人员做 Excel 财务报表，就是在一台虚拟机器上运行 Excel 软件。财务人员专注的目标是 Excel 中的报表，其他的底层细节与他完全无关。

1.1.2　操作系统基本功能

操作系统包含如图 1.2 所示的基本功能，分为 5 个主要组成部分。

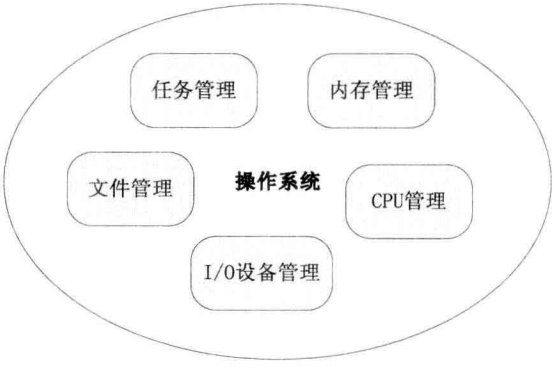

图 1.2　操作系统的功能组成

1．任务管理

任务是程序的一次执行。任务可以分为系统任务和用户任务。系统任务是操作系统本身的任务，如操作系统的主程序、时钟中断服务程序、后面要讲到的空闲任务和统计任务等。用户任务是用户应用程序的运行，如用户设计的计算器软件的一次执行或 Word 软件的运行、本书中给出的一些用户任务。这些任务都需要任务管理部分来管理。

2．CPU 管理

CPU 管理的含义在于多任务 OS 对 CPU 的分配，也就是分配对 CPU 的所有权，即哪个软件正在运行，占有 CPU。可以把它归入任务管理。

3．内存管理

内存是任务的生存空间。内存管理用于给任务分配内存空间，相应的，在任务结束后释放内存空间。

4．文件管理

文件管理是实现对文件的统一管理，是对文件存储器的存储空间进行组织、分配和回收，负责文件的存储、检索、共享和保护。从用户角度来看，文件管理主要是实现“按名取存”，

文件系统的用户只要知道所需文件的文件名，就可存取文件中的信息，而无须知道这些文件究竟存放在什么地方。

5. I/O 设备管理

I/O 设备即管理系统中的各种硬件设备，如打印机、显示器、硬盘等。很明显，用户应用程序应该调用 I/O 设备管理模块提供的 API 来对设备进行操作，而不是直接读/写硬件。

1.2 实时操作系统概述

1.2.1 什么是实时操作系统

实时操作系统（Real Time Operating System，RTOS）是指当外界事件或数据产生时，能够接收并以足够快的速度予以处理，其处理的结果又能在规定的时间内来控制生产过程或对处理系统做出快速响应，并控制所有实时任务协调、一致运行的操作系统。

所谓足够快，就是要使任务能在最晚启动时间之前启动，能在最晚结束时间之前完成。

因而，提供及时响应和高可靠性是其主要特点。实时操作系统有硬实时和软实时之分，硬实时要求在规定的时间内必须完成操作，这是在操作系统设计时保证的；软实时则只要按照任务的优先级，尽可能快地完成操作即可。

实时系统与非实时系统的本质区别就在于实时系统中的任务有时间限制。实时操作系统可以用于不需要实时特性的场合，反之则不行。

1.2.2 实时操作系统的基本特征

实时操作系统具有以下基本特征。

1. 实时操作系统首先是多任务操作系统

实时操作系统最基本的要求就是它是一个多任务的操作系统，即在多任务的基础上，因为实时性的要求，编写出的实时操作系统。所谓多任务，是指允许系统中多个任务同时运行，而 CPU 只有一个，在某一个时刻，只有一个任务占有 CPU。因此，多任务操作系统的核心任务之一就是任务调度，为任务分配 CPU 时间。

2. 多级中断机制

一个实时系统通常需要处理多种外部信息或事件，如串行通信、网络通信或者事件报警，例如温度超高。但处理的紧迫程度有轻重缓急之分，很明显，温度超高的报警事件是最急切的，必须立即做出响应，而通信可以延后处理，并不会使整个系统出现问题。因此，建立多级中断嵌套处理机制，以确保对紧迫程度较高的实时事件进行及时响应和处理是实时操作系统必须具备的功能。