

图书基本信息

书名：<<嵌入式实时操作系统  $\mu$ C\OS- >>

13位ISBN编号：9787810772907

10位ISBN编号：7810772902

出版时间：2003-5

出版时间：北京航空航天大学出版社

作者：JEAN J.LABROSSE

页数：582

译者：邵贝贝

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## 前言

本书是Jean J.Labrosse先生撰写的Micro C / OS- The Real-Time Kernel一书第2版本的中译本，原文书于2002年出版。

该书第1版本面世于1999年，其中译本名为《C / OS-II——源码公开的实时嵌入式操作系统》，2001年由电力出版社出版。

与第1版本比较，第2版本增添了许多新内容。

原文书第1版本的内容只有12章，而第2版本扩展到18章。

第1版本的正文只有357页，其余的122页是C / OS-II V2.00的源程序清单，源码同时也附在一张3寸软盘上；而第2版本有598页，且去掉了C / OS-II的源程序清单，仅将其附在随带的光盘上。

粗略地说，书中有250页是全新的。

其中如何测试实时内核中的每个函数及如何调试RTOS，对于拟将C / OS-II移植到其他CPU上去的读者会特别有用。

作者对原有的约350页内容也做了修订，在编排风格上也有不少改变，使之更便于学习和查阅。

原文书第1版本给出的是C / OS-II V2.00的全部源代码，第2版本给出的是C / OS-II V2.52的全部源代码。

新版本增加了一些新的功能函数，如mutex，Event-flag及浮点运算。

最重要的是，C / OS-II V2.52通过了美国航空航天管理局（FAA）的安全认证，可以用于飞机、航天器等与人生命攸关的控制系统中。

也就是说，用户可以放心地使用C / OS-II V2.52，放心地使用本书中提供的源代码。

提倡在计算机嵌入式应用中使用实时操作系统RTOS（Real Time Operating System），是因为RTOS将应用分解成多任务，简化了应用系统软件的设计；RTOS使控制系统的实时性得到保证，可以接近理论上能达到的最好水平；良好的多任务设计，有助于提高系统的稳定性与可靠性。

嵌入式RTOS是多年来计算机专家们潜心研究的成果，是他们智慧的结晶。

嵌入式系统应用的范围不胜枚举，在一个现代化的社会里，可以说，只要能想到的领域都有计算机的嵌入式应用。

开发嵌入式应用的工程师，往往只是他相关领域的专家，对计算机的理解，尚不能与计算机专家们相比，这就是国外人士提到的嵌入式应用工程师与计算机专家之间的“缝隙（gap）”。

把计算机专家的研究成果拿给嵌入式应用开发的工程师使用，是填补这一缝隙的好方法。

在我国广泛地提倡学习和使用RTOS还是最近几年的事。

这是由于RTOS的商业软件价格昂贵，著名的商业RTOS需几万美元，而且其中大部分是不提供源码的，即是所谓的黑盒子，不便于学习和使用。

另外早年的单片机存储器资源有限，容不下RTOS本身的开销。

随着半导体技术的飞速发展，片上系统（system on chip）成为嵌入式应用的发展趋势，RTOS的使用也随之越来越普及。

开卷有益。

在产品中用也好，不用也好，对于嵌入式应用工作者，学习和理解RTOS是绝对必要的，故译者更看中C / OS-II用做教材的意义。

## 内容概要

$\mu$ C/OSII是著名的、源码公开的实时内核，是专为嵌入式应用设计的，可用于各类8位、16位和32位单片机或DSP。

从 $\mu$ C/OS算起，该内核已有10余年应用史，在诸多领域得到了广泛应用。

本书是MicroC/OSII The Real Time Kernel一书的第2版本，在第1版本（V2.0）基础上做了重大改进与升级。

通过对 $\mu$ C/OSII源代码的分析与描述，讲述了多任务实时的基本概念、竞争与调度算法、任务间同步与通信、存储与定时的管理以及如何处理优先级反转问题；介绍如何将 $\mu$ C/OSII移植到不同CPU上，如何调试移植代码。

在所附光盘中，给出已通过FAA安全认证的 $\mu$ C/OSII V2.52的全部源码以及可在PC机上运行的移植范例。

## 作者简介

Jean J.Labros, 1957年生于加拿大蒙特利尔。

自幼酷爱电子学, 着迷于迪斯科舞厅中的灯光和音响系统。

作为业余嗜好, 大学读书期间曾开过一家名为Mephistronique的公司。

大学3年级时, 开始涉足微处理器的精彩世界。

从Intel 8080开始, 然后是MOTOROLA6800, 接着是Zilog的Z-80, 并将它们应用于其业余产品的设计。

## 书籍目录

概述第1章 初识  $\mu$ C/OSII 1.00 安装  $\mu$ C/OSII 1.01 范例1 1.02 范例2 1.03 范例3 1.04 范例4第2章 实时系统概念 2.00 前/后台系统 2.01 代码的临界段 2.02 资源 2.03 共享资源 2.04 多任务 2.05 任务 2.06 任务切换 2.07 内核 2.08 调度 2.09 不可剥夺型内核 2.10 可剥夺型内核 2.11 可重入函数 2.12 时间片轮番调度法 2.13 任务优先级 2.14 静态优先级 2.15 动态优先级 2.16 优先级反转 2.17 任务优先级分配 2.18 互斥条件 2.18.01 关中断和开中断 2.18.02 测试并置位操作 2.18.03 禁止, 然后允许任务切换 2.18.04 信号量 2.19 死锁 2.20 同步 2.21 事件标志 2.22 任务间通信 2.23 消息邮箱 2.24 消息队列 2.25 中断 2.26 中断延迟 2.27 中断响应 2.28 中断恢复时间 2.29 中断延迟、响应及恢复 2.30 中断处理时间 2.31 非屏蔽中断 2.32 时钟节拍 2.33 对存储器的需求 2.34 使用实时内核的优、缺点 2.35 实时系统小结第3章 内核结构第4章 任务管理第5章 时间管理第6章 事件控制块第7章 信号量管理第8章 互斥型信号量管理第9章 事件标志组管理第10章 消息邮箱管理第11章 消息队列管理第12章 内存管理第13章 移植  $\mu$ C/OSII第14章  $\mu$ C/OSII在80x86上的移植第15章  $\mu$ C/OSII在带有硬件浮点运算单元的80x86上的移植第16章  $\mu$ C/OSII参考手册第17章  $\mu$ C/OSII配置手册第18章 PC 中的功能函数附录A C语言编程规约附录B  $\mu$ C/OSII的使用许可证附录C  $\mu$ C/OSII速查手册附录D 工具程序附录E 参考文献附录F 本书所附光盘

## 章节摘录

插图：我试着让那几个很简单的任务运行起来，竟用了2个月时间。

我一天到晚给该厂商打电话求援。

该厂商声称，内核B是用C语言写的；不管怎样，我须用汇编语言给程序中每个对象初始化。

虽然该厂商很耐心，但我实在是够了！

产品的开发也耽误了。

我真是不想再花时间去调试那个廉价的内核了。

后来得知，我是该厂商的第一个客户。

这个内核实际上没有完全测试和调试过。

还是回到正路上来，我决定使用内核A。

5个用户的价格是5000\$，此外，我的每个产品还要付200\$的内核使用费。

在当时，那可是一笔很大的开支，就算是花钱买个心里踏实吧。

用了2天时间，内核转起来了。

项目进展到3个月上，我的一个工程师发现，内核好像有毛病（bug）。

当内核中的这个毛病被确认无疑后，我把我的代码寄给了该厂商。

此时，厂商提供的90天的担保已经过期了。

为了得到答复，我还须支付每年500\$的维护费。

我和销售商争辩了好几个月，我认为，他们应该尽快处理这一处毛病才是，我实际是在为他们做好事；可是，他们不予理睬。

最后，我只好认了，又签了程序维护合同，而该厂商6个月以后才处理了那个毛病。

是的，6个月以后！

我简直气坏了，最要紧的是，我的产品开发耽误了。

用了将近1年的时间，内核A才在我的产品中稳定运行。

然而，我得承认，从那儿以后，内核A再没有什么问题了。

渐渐地，我自然会想到，写个实时内核有那么难吗？

不就是不断地保存、恢复CPU的那些寄存器嘛。

那是我决定试试写我自己的实时内核时的情景，主要是用业余时间、晚上和周末。

也用了大约1年的时间，才使内核转起来，在某些方面还比内核A好。

我并不想开公司去卖这个内核，因为当时市场上已经有大约50个实时内核的产品了，人们会想，怎么又多出一个来了呢？

媒体关注与评论

如果您从未用过RTOS，那么这本书将让您掌握RTOS——开发应用程序的得力工具；如果您打算选购商用RTOS，那么这本书为您揭示RTOS的内在奥秘。

——JackGanssle，顾问，教师，作家

## 编辑推荐

《嵌入式实时操作系统  $\mu$ C/OS-2(第2版)》介绍  $\mu$ C/OS- 的设计与实现,讲解多任务实时系统的基本原理。

读者可以学到内核结构、调度算法、任务切换、时间管理、任务间通信与同步及存储空间的动态分配等知识。

与《嵌入式实时操作系统  $\mu$ C/OS-2(第2版)》的前一版本相比,书中程序的解释全部重新写过,给出了更详尽的说明,以便于理解;系统功能函数的描述从1章扩充到了6章,其中2章用以描述新增加的功能函数:互斥型信号量(mutex)和事件标志组函数;重写了程序移植指南,新增加了移植过程中的调试技巧等内容;编程规约函数以及变量等的速查手册也编排在附录中。

附带的光盘包含了  $\mu$ C/OS- V2.52的全部源代码,以及移植到Intel 80x86处理器上的范例,在实模式模块方式下编译。

只需几分钟,就可以在用户PC机的Windows环境下测试  $\mu$ C/OS- 的一些特性。

$\mu$ C/OS- 是一种可移植、可固化、可裁剪及可剥夺型的多任务实时内核(RTOS),适用于各种微处理器和微控制器。

其性能足可媲美于商用内核,在某些方面甚至更佳。

所有代码用ANSI的C语言编写,故具有良好的可移植性。

实际上  $\mu$ C/OS- 已经被移植到40多种处理器架构中,从8位到64位的各种CPU。

自1992年起,  $\mu$ C/OS- 已在全世界从照相机到航空设备等各个领域得到广泛应用。

这个RTOS的健壮性、可靠性和安全性已经得到了广泛的证明。

对于各类系统,包括一些对安全性要求极为苛刻的系统,它都能够胜任。

#### 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>