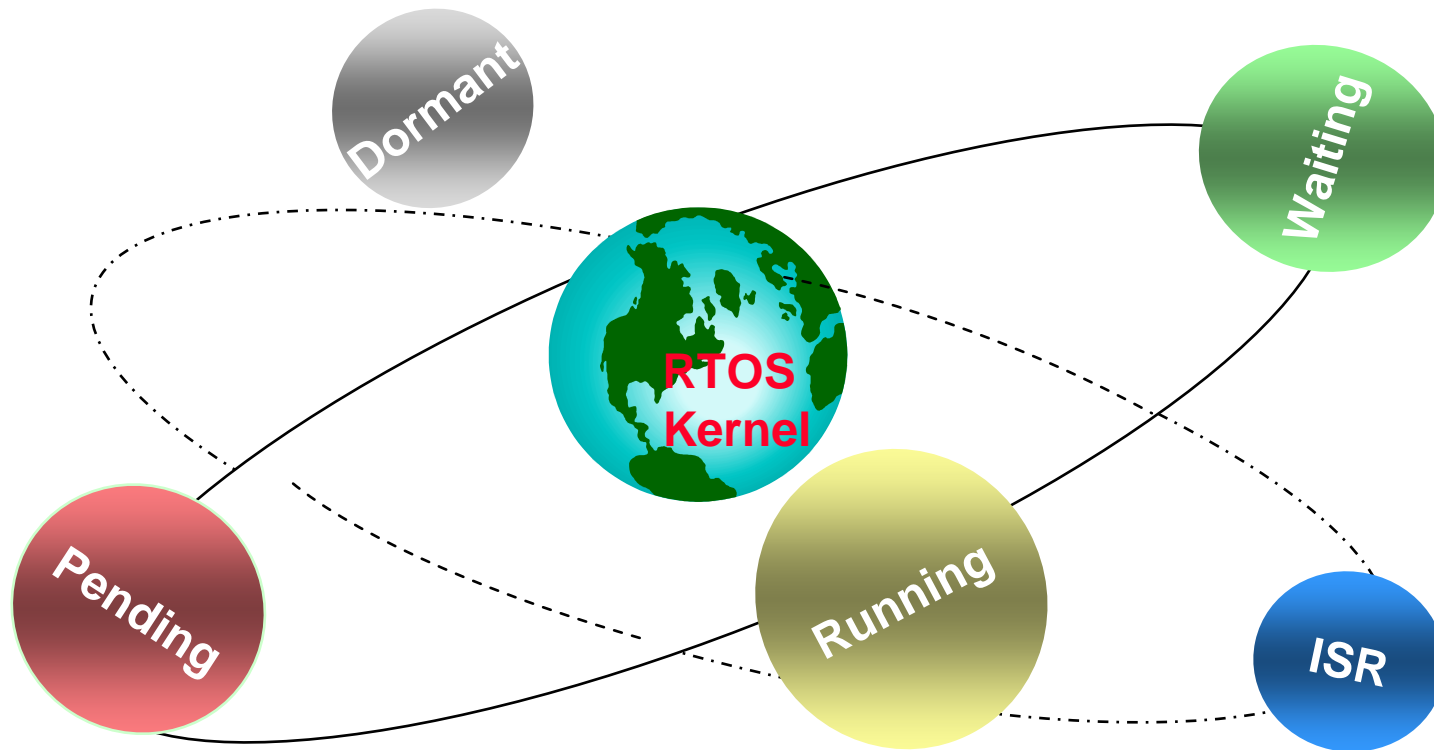
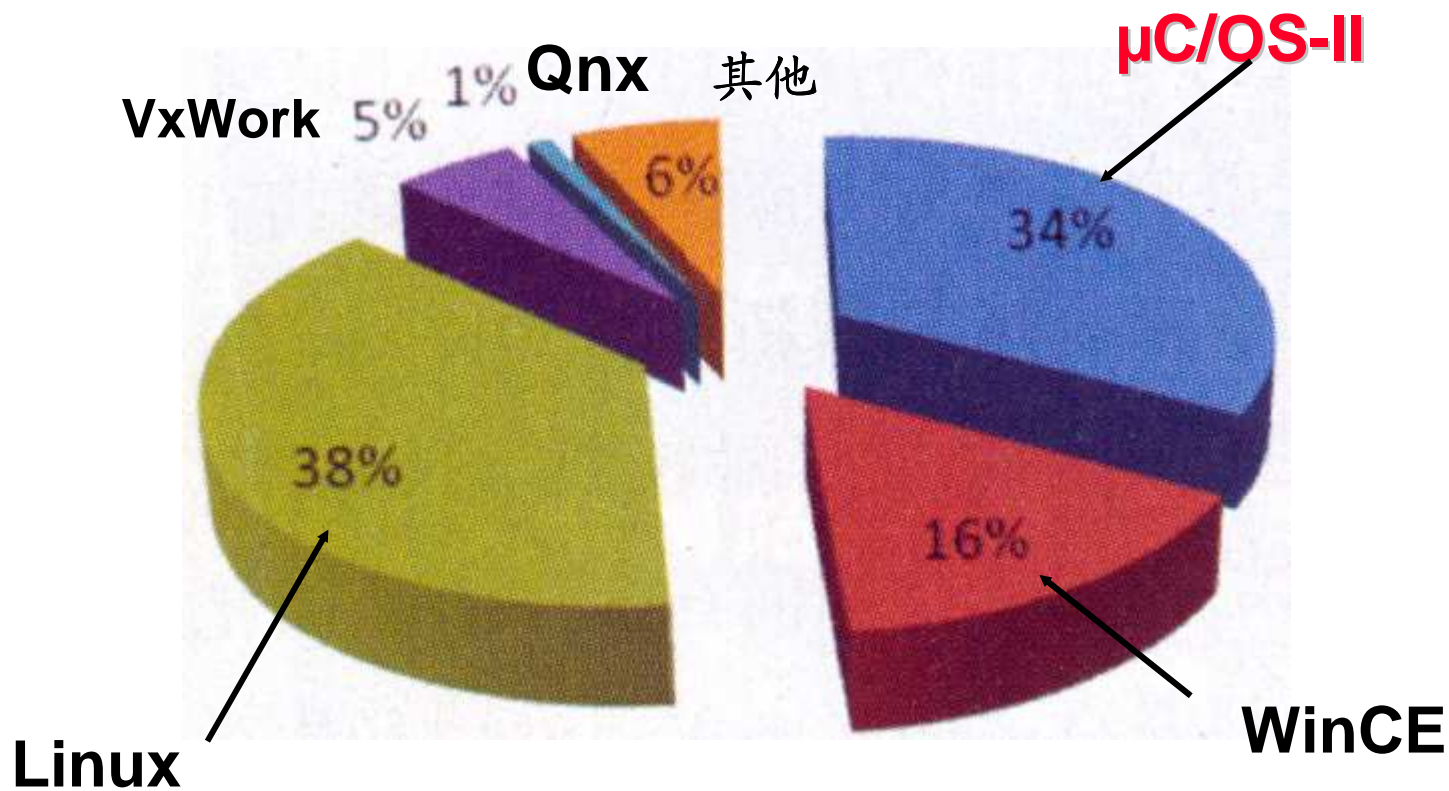


# 嵌入式实时操作系统—— $\mu\text{C}/\text{OS-III}$



# 中国的嵌入式OS



资料来源：《电子产品世界2009年第1期》  
“2008年嵌入式应用调查报告627份调查问卷”

# μC/OS 和 μC/OS-II

Jean J.Labrosse

R & D Publications, Inc

- ◆ 最多可管理**60**个任务
- ◆ 基于优先级的可剥夺型内核(查表算法)
- ◆ **90%**以上是硬件无关的**C**代码
- ◆ 与**CPU**有关的汇编和**C**代码 <**200**行
- ◆ 可在**PC**环境下运行, 源代码在书中所附的软盘或光盘上
- ◆ 可裁剪、可固化, 绝大多数**CPU**相关移植软件可从网上下载
- ◆ 书中有详细解释, 编程风格很专业、漂亮

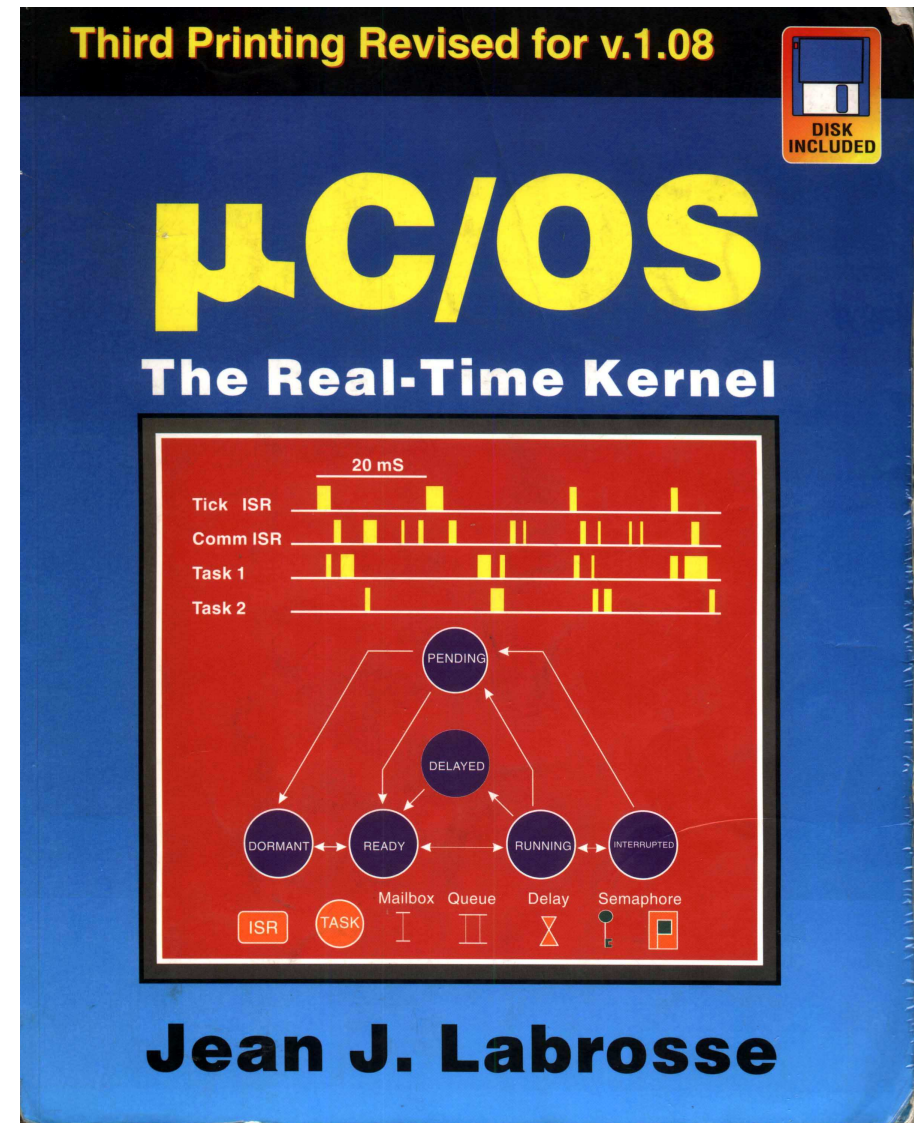
# 第一本书

是很好的教材

261页，含源代码约100页

## Licensing:

*You do not need a license to use this code in your application, if your application is distributed in object format. You should however, indicate in your product literature that you are using  $\mu$ C/OS, the Real Time Kernel. If you distribute  $\mu$ C/OS in source code, you must obtain a license.*



# 第二本书

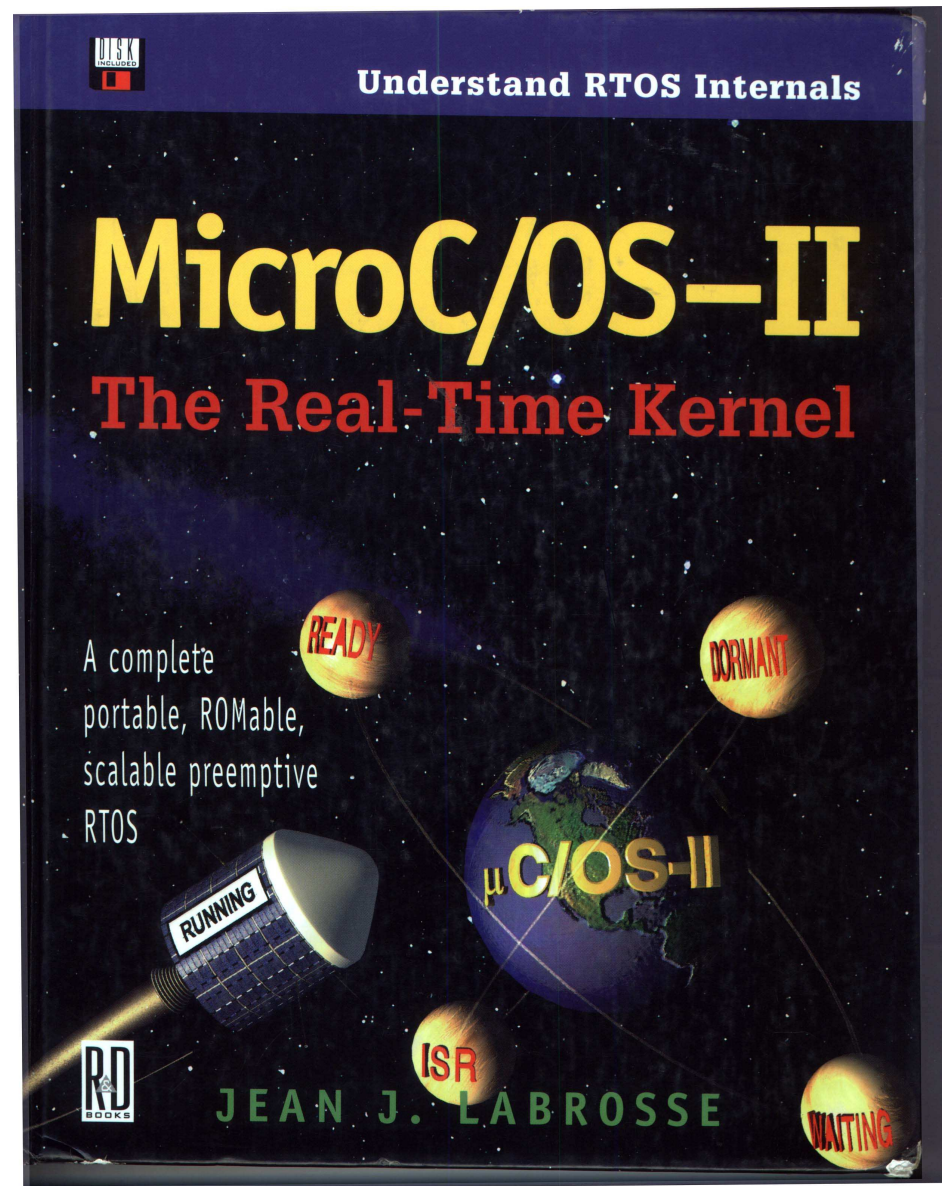
500页

含150页源码

## *Licensing*

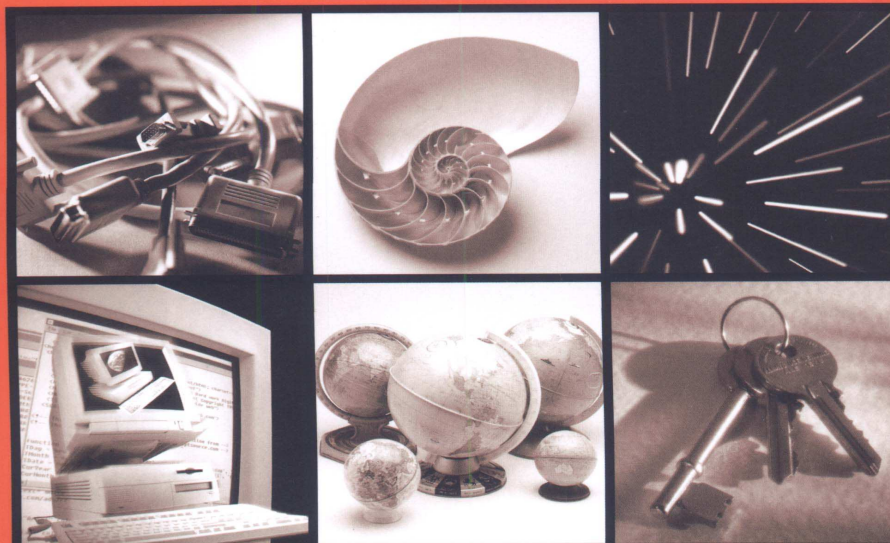
No licensing is required if  $\mu$ C/OS-II is used for education

You must obtain an Object Code Distribution License to embed  $\mu$ C/OS-II in commercial product that is sold with the intent to make profit



# 第二本书 的中译本


(2001年)



## μC/OS-II

—源码公开的实时嵌入式操作系统

JEAN J. LABROSSE 著  
邵贝贝 译

 本书附赠光盘一张

**CMP books!**



中国电力出版社  
www.infopower.com.cn

## 第三本书

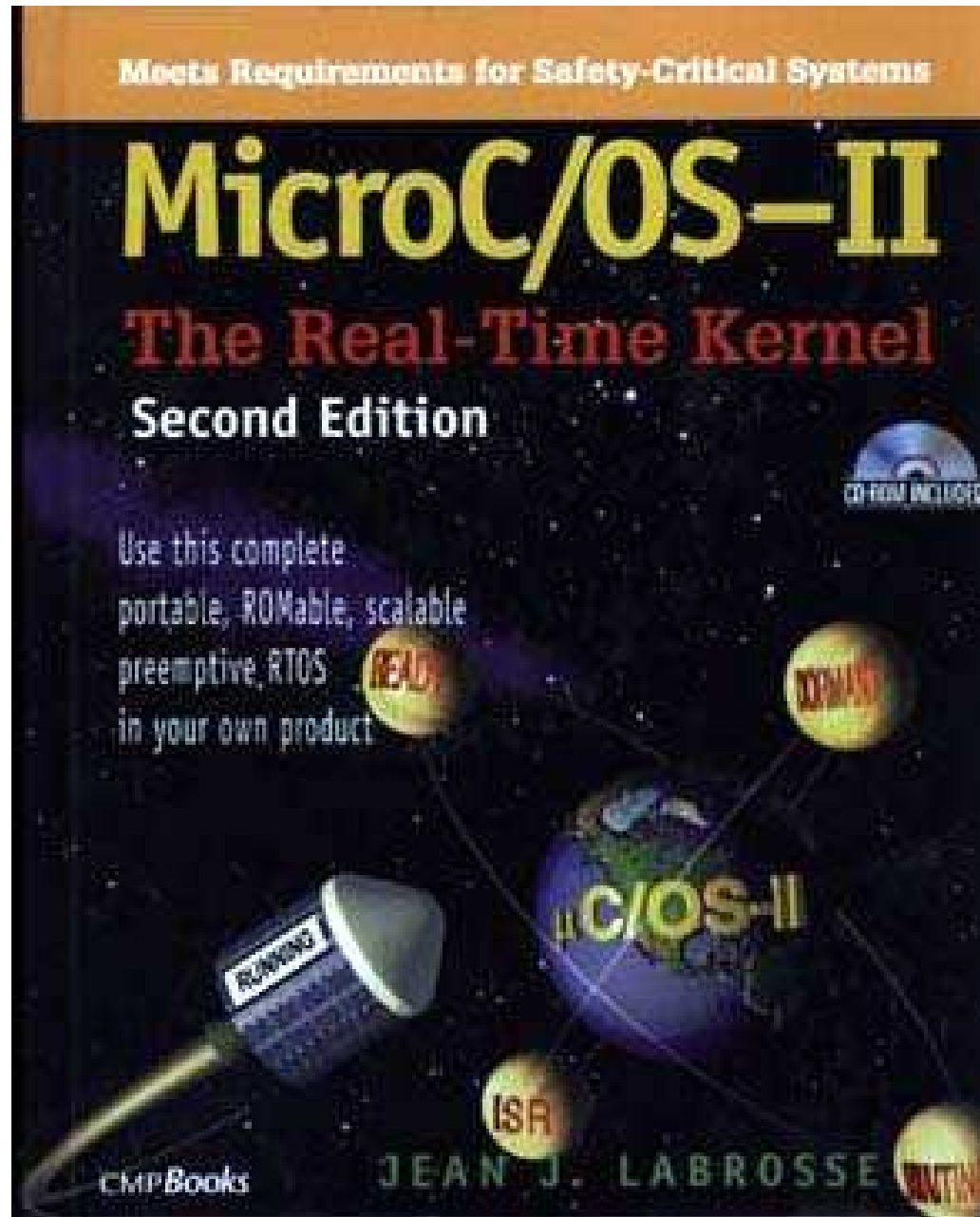
The 2<sup>nd</sup> edit of  
**μC/OS-II**

从498页增加到600  
页，将150页源码附在  
光盘上

从12 章增加到 18 章  
增加了250页新内容

**Meet Requirement for  
Safety-Critical System**

**June 2002**



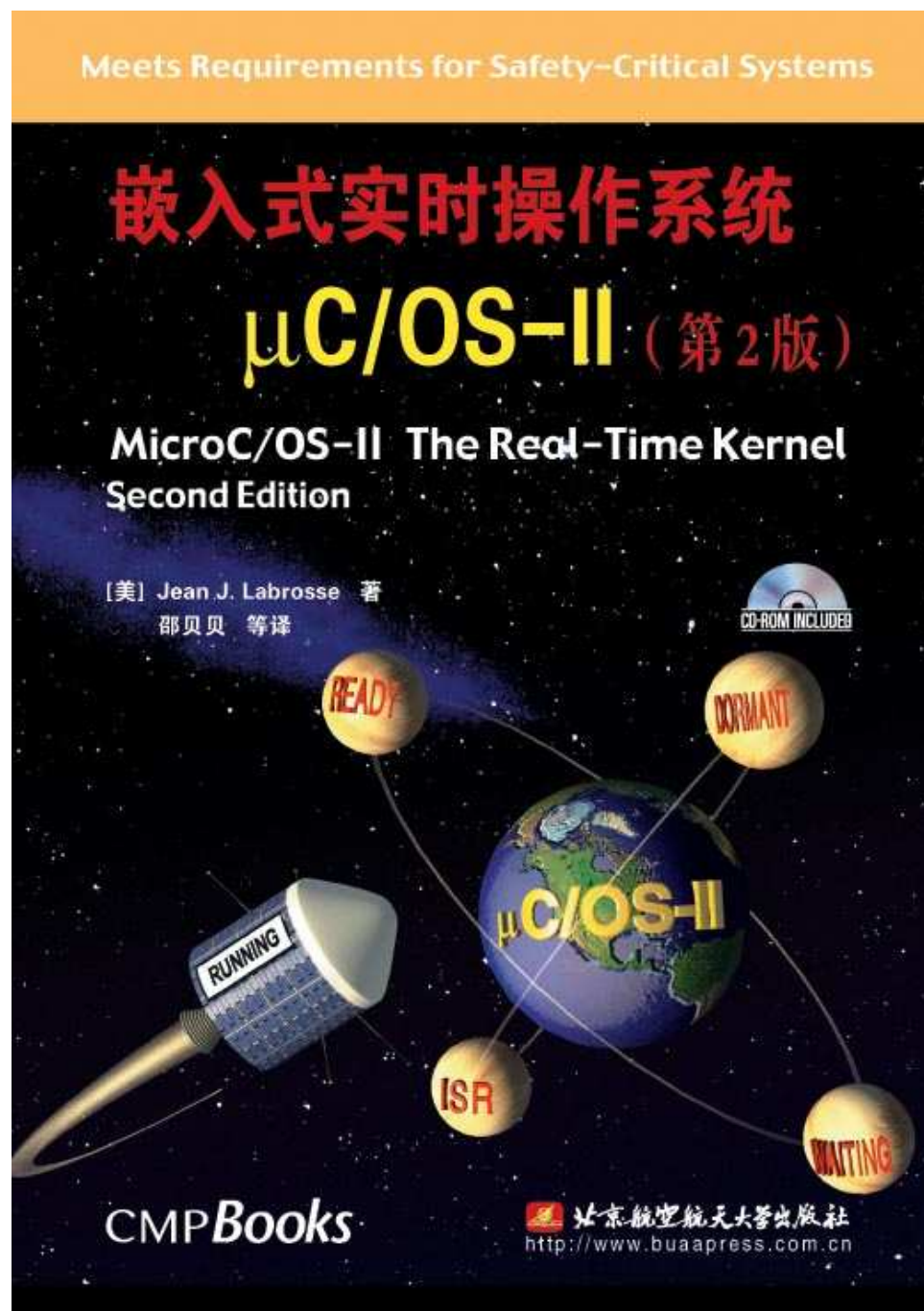
# 第三本书 的中译本

## $\mu$ C/OS-II

V2.52

Published in  
June 2003

Graduated Students  
are encouraged to  
use it in their  
projects, by.....





# 授权许可 License

- ◆ **μC/OS and μC/OS-II source and object code can be used by accredited Colleges and Universities without requiring a license, as long as there is no commercial application involved. In other words, no licensing is required if μC/OS and μC/OS-II is used for educational use.**

只要无牟利商业行为，**μC/OS** 和**μC/OS-II** 的源码和目标码可用于正规大专院校的教学和科研，无需授权。

- ◆ **You need to obtain an 'Object Code Distribution License' to embed μC/OS or μC/OS-II in a product that is sold with the intent to make a profit or if the product is not used for education or 'peaceful' research. Please indicate the processor type(s) you will be using (i.e. 68HC11, 80x86, AVR, etc.).**

若将**μC/OS** 或 **μC/OS-II** 嵌入到意在牟利的产品中出售，而该产品非用于教学及用于非和平类研究，必须得到授权许可，并请指出所用**CPU**类型(如 **68HC11, 80x86, AVR**等)。

# μC/OS-II 中的任务调度算法

## Task Scheduling Algorithm

- ◆ 基于优先级的算法 **Priority based algorithm**
- ◆ **CPU**必须总是运行进入就绪态的任务中优先级最高的那个任务

**The CPU must always execute the highest priority task that is ready to run.**

- ◆ 最大任务数**64**个
- ◆ **0**优先级最高; **63** 优先级最低

## **$\mu\text{C}/\text{OS-III}$ 和 $\mu\text{C}/\text{OS-II}$ 有什么不同? (1)**

- ◆  **$\mu\text{C}/\text{OS-III}$  允许任务使用同一优先级，对优先级相同的任务， $\mu\text{C}/\text{OS-III}$  支持时间片调度法。**

## **μC/OS-III 和 μC/OS-II 有什么不同? (2)**

- ◆ **μC/OS-III 允许用户在程序运行中动态配置实时操作系统内核资源，例如、任务、任务栈、信号量、事件标志组、消息队列、消息数、互斥型信号量、存储块划分和定时器，都可以在程序运行中变更。这样，用户可以避免在程序编译过程中出现资源不够分配的问题。**

## **μC/OS-III 和 μC/OS-II 有什么不同? (3)**

- ◆ **μC/OS-III**中，最多任务数为**64**个（**V2.82**以后是**256**个）
- ◆ **μC/OS-III**中，用户可以有任意多的任务、任意多的信号量、互斥型信号量、事件标志、消息队列、定时器和任意分配的存储块容量（仅受限于用户**CPU**可以使用的**RAM**量）。

# $\mu\text{C}/\text{OS-III}$ 和 $\mu\text{C}/\text{OS-II}$ 有什么不同? (4)

- ◆  $\mu\text{C}/\text{OS-III}$  增加了很多新功能, 如:
  - 可嵌套的互斥型信号量
  - 可嵌套的任务挂起
  - 向无信号量请求的任务发送信号量
  - 向无消息队列请求的任务发消息
  - 任务被内核的多个元素挂起
  - 为其他任务的状态或“出错代码”服务的任务
  - 内在性能测试
  - 防死锁机制
  - 用户可定义的接口函数等

## μC/OS-III 和 μC/OS-II 有什么不同? (5)

- ◆ 除每个任务的最长关中断时间，内在性能测试允许用户测得系统的最长关中断时间。
- ◆ 用户在测得每个任务的最长禁止调度时间基础上，可测得系统禁止任务调度的最长时间。
- ◆ 每次发出的信息都带有时间戳，用户也容易得到任务级的响应时间。
- ◆ 性能测试还包括任务切换次数计数器和每个任务的CPU使用率等。

## **μC/OS-III 和 μC/OS-II 有什么不同? (6)**

- ◆ **μC/OS-III** 设计成能方便地按照**CPU**架构优化。特别是其多数数据类型可按照**CPU**能适应的最佳位数宽度修改。
- ◆ 选择和确定优先级的算法可以用汇编语言写，以发挥一些有特殊指令的**CPU**的优势，如：
  - 置位和复位指令
  - 计数器计到零(**CLZ**)
  - 找出第一个不为零位(**FF1**)指令等



## **μC/OS-III 和 μC/OS-II 有什么不同? (7)**

- ◆ **μC/OS-III** 有很多数需要自行处理的内部据结构和变量，是通过给调度器上锁的方式保护这些临界段代码的，不使用关中断的方式，内核关中断的时钟周期数几乎为零。这就保证了**μC/OS-III** 能够响应那些最快的中断源。

## **μC/OS-III 和 μC/OS-II 有什么不同? (8)**

- ◆ **μC/OS-III 还支持内核觉察式调试。用户友好地使内置内核觉察调试器检查、显示μC/OS-III 的变量、数据结构，支持μC/Probe 工具在程序运行过程中动态显示修改各种变量。**

# **μC/OS-III 与 VxWorks**

- ◆ 从小内核**μC/OS-II**到**μC/OS-III**，很多性能在接近**VxWorks**
- ◆ **Wind River 1981**成立,在**15**个国家有**1,600**多员工，年收入**\$3.6**亿，主营**VxWorks**，用于网络设备、航天、国防等
- ◆ **VxWorks** 是公认的最好的嵌入式**RTOS**，是完整的、自成体系的软件包
  - 曾经靠**BSP**支持的是黑盒子
  - 近些年来开始公开源码
  - 价格贵、运行成本高

# Intel 收购 Wind River (2009.6.4.)

- ◆ Wind River 打算进一步转向 Linux 用于 ARM 和 PowerPC
- ◆ Intel 为支持多核设计，打算发展软件，如嵌入式应用、Linux
- ◆ 面向可视电话、移动上网设备、车载娱乐/信息系统、航天、国防等

# μC/OS-II 目前的商业模式

- ◆ 10人左右的团队, 维护和发展内核
- ◆ 和第3方合作形成完整的软件包体系
  - GUI
  - FS
  - TCP/IP
  - USB host & device)
  - CAN
  - ModBus
  - FlashLoader
  - **μC/Probe**
  - Building Block
  - .....
- ◆ 和自由软件的结合

# Micrium

Empowering embedded systems

# Micrium 主页

Home

About Micrium

Products

News & Events

Sales

Customer Support

Downloads

Contact Us

Search

Sitemap



Royalty-free  
**Micrium** software  
targets **embedded**  
designs

Let **Micrium's** robust software  
help you **accelerate your next design...**

Micrium's vision is to provide the highest-quality embedded software components in the industry, in the form of engineer-friendly source code with unsurpassed documentation and customer support. Micrium's products consistently deliver on that vision to shorten time-to-market throughout all product development cycles.

In the complex world of short design development cycles and higher-than-ever demand for quality, embedded software is a major investment. Today's design engineers require embedded code that works right away, every time.

Micrium's embedded software components are written with the design engineer in mind. Ease-of-use is a commitment that guides all product development. Unlike other software providers, Micrium provides source code, ensuring full control over the code's implementation.

**μC/OS-II™**

The Real-Time Kernel

**μC/TCP-IP™**

Protocol Stack

**μC/FS™**

Embedded File System

**μC/GUI™**

Embedded Graphical User Interface

**μC/USB Device™**

Universal Serial Bus Device Stack

**μC/USB Host™**

Universal Serial Bus Host Stack

**μC/FL™**

Flash Loader

**μC/Modbus™**

Embedded Modbus Stack

**μC/CAN™**

CAN Protocol Stack

**μC/BuildingBlocks™**

Embedded Software Components

**μC/Probe™**

Real-Time Monitoring

## 思考与体会

- ◆ 从 $\mu\text{C}/\text{OS}$ 发展到 $\mu\text{C}/\text{OS-III}$ ，经历了17年，从1个小内核发展成为1个可以和大的商业RTOS媲美的专门用于嵌入式系统的成熟的RTOS体系
  - 不但开放源码，且出版图书、教用户学习、理解RTOS的商业模式的成功，说明开发源码是趋势
  - 需要有团队维护、发展
  - 需要第3方的支持
  - 需要有与发展相适应的商业模式
  - 是对VxWorks等大公司产品的冲击

# 2009年4月7日 Jean Labrosse 夫妇来清华访问

