



■ 题目：实时多任务内核的研究与应用（1999 年）

■ 作者：王若鹏

■ 摘要：

目前国内的大部分嵌入式中、小型系统的开发设计者都采用的是传统的单线程的开发方式，这样的开发方式使开发、调试和维护过程显得冗长和复杂，系统化程度比较低，所以不是一个很正规化的系统开发方案。本论文的作者以两年的对实时多任务内核的分析和应用经验为基础，通过解剖实时内核  $\mu C/OS$  的实现方法，为广大的设计工程师提供一种设计中、小型实时系统的有效的解决方案。所以这里的重点是如何以已有的内核为工具，应用到自己的目标系统中，而不是如何自行设计一个实时内核。

实时多任务内核是实时操作系统的一个派生产物。它没有操作系统的内存管理、文件系统等庞大内容，但它所具有的是实时操作系统的精华，即多任务实时调度和任务的定时和同步操作。其二进制代码的大小一般不会超过 10K Byte，是纯粹为嵌入式应用而设计，具有很短的任务切换时间和实时响应速度。

本文首先讨论了实时内核的一般性问题，例如多任务的概念，基于优先级的抢占式调度方式，实时系统的核心—参考时钟的意义，各任务间的相互通讯和同步操作（信号灯、消息和消息队列），以及实时响应速度等内容。随后，从任务，中断处理，通讯/同步三个方面介绍了一个典型的实时多任务内核  $\mu C/OS$  的实现方法。“ $\mu C/OS$ ” 是 Micro-Controller Operating System 的缩写，它是由美国工程师 Jean J. Labrosse 试图为 Motorola 八位微控制器 MC68HC11 开发一个微小但实用的内核时诞生的。 $\mu C/OS$  绝大部分代码是用 C 语言写成的，与 CPU 类型有关的代码仅占尽可能小的一部分，所以它很快就被移植到各种 8 位、16 位或 32 位的处理器上。目前，它已经在世界控制领域取得了一席之地，被广泛应用于各种中小型嵌入式实时控制系统中。 $\mu C/OS$  虽然仅是一个微内核，但它具有可同大型商业实时操作系统相媲美的优越性能，非常适合于在我国推广、使用。

第五章以一个应用  $\mu C/OS$  开发的实例来说明如何有效的将这个内核应用到一个目标系统中。此实例是作者对合肥国家同步辐射实验室的环境监测仪 DDL（Digital Data Logger）系统的改造方案。论文详细介绍了 DDL 的改造目的和要求，以及硬件和软件设计方案，尤其着重描述了应该根据什么原则来将整体的设计目标划分为一个个相对独立的任务模块，如何给每个任务赋予一定的优先级，这些都是实时应用中最为重要的内容。希望能通过这个实例提供将  $\mu C/OS$  置入目标系统中的一个比较具体的方法。

论文最后总结了应用实时多任务内核进行开发的几个突出优点，即提高系统的实时响应速度，有利于软件设计的模块化，提高系统的整体性能，以及提高抗干扰能力。

附录中列出了开发实例环境监测仪 DDL 的硬件总原理图。