

嵌入式操作系统实时性研究与改进

孔 军

(安徽理工大学, 安徽 淮南 232001)

摘 要: 研究了影响嵌入式操作系统实时性的几个因素,以 $\mu\text{C}/\text{OS-II}$ 为例讨论了对嵌入式操作系统实时性影响最大的任务调度,并提出了对嵌入式操作系统内核实时性的改进。

关键词: 实时系统; 嵌入式操作系统; 任务调度

Abstract: This paper studies several factors affecting the real time of embedded operating system. Take $\mu\text{C}/\text{OS-II}$ as the example discussed task scheduling what affected the real-time biggest of embedded operating system, and put forward method to improve the real-time of the embedded operating system.

Key words: Real-Time System; Embedded Operating System; Task Schedule

引言

在嵌入式领域中, 嵌入式操作系统(RTOS)已经得到了非常广泛的应用。应用嵌入式操作系统可以有效的管理各种资源, 简化软硬件设计。对 RTOS 来说最重要的就是实时性了。实时性是指系统能够在限定的时间内完成任务并对外部的异步事件作出及时响应, 即任务完成时间的确定性。在大多数工业控制中, 对实时性的要求非常高。

1 嵌入式操作系统实时性的几个重要指标

影响嵌入式操作系统实时性的因素有很多。下面就讨论几个衡量 RTOS 的最重要的指标。

在讨论这几个指标的时候我们以 $\mu\text{C}/\text{OS-II}$ 为例。

1.1 任务切换时间 (Task Content Switch Time)

可以反映出 RTOS 执行任务的速度。 $\mu\text{C}/\text{OS-II}$ 使用的是抢占式内核, 以保证系统的响应时间。每个任务都被赋予一定的优先级, 最高优先级的任务一旦就绪, 就能得到 CPU 的控制权。当一个运行着的任务通过信号量等机制使一个更高优先级的任务进入了就绪态, $\mu\text{C}/\text{OS-II}$ 会进行任务调度。这时当前任务的 CPU 使用权就要被剥夺, 那个高优先级的任务会立刻得到 CPU 的控制权。

每个任务都有自己的一套 CPU 寄存器和栈空间。任务的切换实际上就是 CPU 寄存器内容的切换。CPU 内部寄存器越多, 额外负荷就越重。在任务切换之前还需要在就绪表中找出优先级最高的任务, 它由任务调度函数 $\text{OSSched}()$ 完成, 是比较花费时间的。因为这个函数有固定长度的语句, 所以它的执行时间是常数, 与应用程序建立了多少个任务没有关系。

所以任务切换时间取决于 CPU 有多少寄存器要出入栈, 以及相关调度函数的执行速度。

1.2 中断响应时间 (Interrupt Response Time)

可以反映出 RTOS 对外界变化的反应速度, 是指从中断发生起到执行中断处理程序的第一条指令所用的时间。它是衡量嵌入式实时操作系统实时性能的最主要、最具有代表性的性能指标。

中断响应时间 = 中断延迟时间 + 保存 CPU 状态的时间 + 该内核的 ISR 进入函数的执行时间

中断延迟时间 = 关中断的最长时间 + 开

始执行中断服务子程序(ISR)的第 1 条指令的时间

关中断的最长时间取决于运行时不同的情况, 其他参数在其系统中都有固定长度代码。中断响应是系统在最坏情况下响应中断的时间。

1.3 任务抢占时间和信号量混洗时间

任务抢占时间是高优先级的任务从正在运行的低优先级任务中获得系统控制权所消耗的时间。信号量混洗时间指从一个任务释放信号量到另一个等待该信号量的任务被激活的时间延迟。

2 嵌入式操作系统的任务调度算法

调度是内核的主要任务之一, 大多数实时内核都是基于优先级调度的, 每个任务根据其重要程度的不同被赋予一定的优先级。基于优先级的调度法是指, CPU 总是让处于就绪态优先级最高的任务先运行。高优先级的任务何时掌握 CPU 的使用权是由内核的类型确定。由于实时内核的强实时性要求, 不可能等优先级低的任务主动放弃 CPU 使用权, 所以实时内核基本都是可剥夺的。

操作系统的实时性主要体现在: 当优先级高的任务要求工作时, 操作系统要以最快的时间将此任务调度到 CPU 执行。这里所花费的时间主要包括两部分: 查找最高优先级任务和任务上下文切换。其中, 任务上下文切换时间是和处理器相关的, 操作系统无法控制。我们来分析下 $\mu\text{C}/\text{OS-II}$ 如何查找最高优先级任务的。

因为任务较少, $\mu\text{C}/\text{OS-II}$ 采用单一优先级, 这为算法的实现提供了很大的方便在 $\mu\text{C}/\text{OS-II}$ 中, 优先级可以作为任务的标识 (当然要在任务存在的情况下, 是通过一个指针数组实现的) 采用。调度算法主要基于分级查询。考虑到任务数目 <64 , 可以用 6bit 来表示, 分为高 3 位和低三位。 $\mu\text{C}/\text{OS-II}$ 将优先级进行分组, 按高三位进行分组, 可得 8 个 (最多) 优先级数组 (000~111); 每个优先级的在数组中的位置由其低三位表示。在源码中, 高三位用带 Y 后缀的变量表示, 而低三位用带 X 后缀的变量表示。这样建立了 1 个变量 OSRdyGrp (INT8U, 8bit, 每个 bit 代表一组) 和 1 个数组 $\text{OSRdyTbl}[8]$ (INT8U, 每组 8bit, 每个 bit 代表一个优先级)。这样形成了的二级查询, 先选组, 再选组内偏移。

3 $\mu\text{C}/\text{OS-II}$ 的任务调度的不足

我们知道, 实时性分为弱实时性, 一般实时性和强实时性 3 种, $\mu\text{C}/\text{OS-II}$ 非常适合强实时

性, 只要高优先级的任务不放弃 CPU, 低优先级的任务就不会执行, 只有等高优先级的任务执行完才能执行。如果是弱实时性或者一般实时性要求的的话, 就不大适合了, 所以合理的改进 RTOS 的任务调度, 使 $\mu\text{C}/\text{OS-II}$ 这样的 RTOS 更适合弱实时性和强实时性也是嵌入式实时操作系统的一个研究方向。

4 对 $\mu\text{C}/\text{OS-II}$ 的内核的改进方法

现在对嵌入式操作系统的改进的方法有很多, 大致可以归为两类。

4.1 对非 RTOS 进行扩展, 使其适用于实时任务, 比如对于对 Linux 的增加一个实时内核, 只要在硬件中断和原来的 linux 内核之内加入一个实时内核, 专门用来处理实时进程。

4.2 对强实时内核的任务调度进行修改, 比如对 $\mu\text{C}/\text{OS-II}$ 增加时间片轮转的方法, 将任务分类, 增加非实时任务, 分为实时任务和非实时任务, 实时任务仍然是抢占的, 而非实时任务则是等待自己的时间片来得到执行。

5 结束语

随着嵌入式系统的不断发展, 嵌入式操作系统已经得到了越来越广泛的应用。商用嵌入式操作系统内核一般价格昂贵, 这限制了嵌入式操作系统的应用。所以选择开源的免费的内核是一个很好的办法, 对这些内核进行改进, 让分时系统具备实时性, 让强实时内核也可以应用到弱实时和一般实时性场合。这些改进必将促进嵌入式操作系统在嵌入式领域的广泛应用。

参考文献

- [1] 沈胜庆. 嵌入式操作系统的内核研究[J]. 微机计算机信息, 2006, (05).
- [2] 石世光, 陈云洽, 叶奇明. 嵌入式操作系统 $\mu\text{C}/\text{OS}$ 的运行机制 [J]. 计算机技术与发展, 2006, (08).
- [3] 嵌入式实时操作系统的内核抢占机制研究[J]. 计算机工程, 2005, 24.
- [4] [美] Jean J. Labrosse 著, 绍贝贝等译. 嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ [M]. 北京: 北京航空航天大学出版社, 2006.

作者简介: 孔军(1980~), 男, 汉族, 硕士研究生, 主要研究方向为嵌入式系统。

责任编辑: 程鹏