

3种嵌入式操作系统内核的关键技术分析^{*}

金亮, 张学杰

(云南大学 计算机科学与工程系, 云南 昆明 650091)

摘要:内核是操作系统的核心组成部分. 其中, 任务管理、进程的同步和通信机制、存储器管理等是内核实现的关键技术. 对 $\mu\text{C}/\text{OS}-\text{II}$, Windows CE 和嵌入式 Linux 3 种嵌入式操作系统的内核关键技术进行了比较研究后, 详细讨论了它们之间的差别. 在对其适应性进行了深入分析的基础上, 给出了嵌入式操作系统的一些选型原则.

关键词:嵌入式操作系统; $\mu\text{C}/\text{OS}-\text{II}$; Windows CE; 嵌入式 Linux

中图分类号: TP 316 **文献标识码:** A **文章编号:** 0258-7971(2006)S2-0001-04

嵌入式系统(embedded system)是以应用为中心, 软硬件可裁减的, 适用于对功能、可靠性、成本、体积、功耗等综合性能严格要求的专用计算机系统. 具有软件代码小、高度自动化、响应速度快等特点, 特别适合于要求实时和多任务的体系^[1].

嵌入式系统一般包括硬件部分和软件部分. 硬件部分包括微处理器、存储器、外部设备和 I/O 端口等. 软件部分中极为重要的组成部分是嵌入式操作系统, 通常包括系统内核、与硬件相关的底层驱动程序、设备驱动接口、通信协议、图形界面、标准化浏览器等^[1].

内核是嵌入式操作系统的核心组成部分. 它主要负责任务管理、进程间通信管理、存储器管理等. 任务管理模块主要负责进程的创建和删除; 进程间通信模块通过消息传递机制负责进程间的并发控制和通信; 存储器管理负责对内存资源进行有效的管理.

嵌入式操作系统有很多, 如 VxWorks, PalmOS, Windows CE, 嵌入式 Linux, $\mu\text{C}/\text{OS}-\text{II}$ 等. 这其中嵌入式 Linux 和 $\mu\text{C}/\text{OS}-\text{II}$ 是源代码公开的操作系统, 可以通过分析源代码来深刻认识他们的管理机制. 而 Windows CE 被认为是高端嵌入式设备最好的操作系统之一^[2]. 本文以这 3 个嵌入式

操作系统为例, 通过对它们关键实现技术的分析和比较, 为不同领域选择合适的嵌入式操作系统提供参考. 本文所分析的 Linux 内核指的是 2.6 系列内核.

1 嵌入式操作系统的分析和比较

下面主要从任务管理、进程间的同步与通信机制、存储器管理等方面比较分析这 3 种嵌入式操作系统.

1.1 任务管理 任务管理是嵌入式操作系统的核心和灵魂, 它决定了嵌入式操作系统的整体性能. 通常任务管理涉及到以下技术: 优先级设置、时间确定性、基于优先级抢占式调度、时间片轮转调度、多任务调度机制^[3].

$\mu\text{C}/\text{OS}-\text{II}$ 是基于静态优先级的完全占先式的实时内核, 它总是运行就绪状态下优先级最高的任务. $\mu\text{C}/\text{OS}-\text{II}$ 中任何时刻都不可能同时有相同优先级的任务在就绪表中, 优先级号同时也是任务的标识, 因此它没有实现同优先级的调度算法. $\mu\text{C}/\text{OS}-\text{II}$ 通过任务就绪表中的 OSRdyGrp 和 OSRdyTbl[0...7] 变量以及任务调度器 OSS2ched(), 来完成对最多 64 个不同优先级任务的调度^[3].

Windows CE 是一个可抢占的多任务的操作

• 收稿日期: 2006-08-31

基金项目: 国家自然科学基金资助项目(60573104)

作者简介: 金亮(1980-), 男, 湖南人, 硕士生, 主要从事嵌入式操作系统、嵌入式系统应用方面的研究.

系统,它采用基于动态优先级的时间片轮转算法.在 Windows CE 系统中进程是资源分配的基本单位,而线程是调度的基本单位.系统为每一个优先级的线程建立一个可运行队列(最多可有 256 个),在一个队列中排列着相同优先级的可运行线程,线程每次运行一个固定的时间片(一般是 10ms)^[4].

嵌入式 Linux 也是一种可抢占的多任务操作系统,采用了一种叫 O(1)的调度算法,这种算法开销恒定,可以保证不管输入有多大,调度程序都可以在恒定的时间内完成,实时性得到很大的加强^[5].O(1)程序定义在 kernel/sched.c 中,是基于动态优先级的调度方法.对于相同优先级,嵌入式 Linux 采用时间片轮转方法,调度程序可以根据进程的优先级动态地调整分配给它的时间片,从而保证了优先级高的进程执行的频率高,执行时间长.

表 1 显示了 3 种嵌入式操作系统调度机制的比较.

表 1 3 种嵌入式操作系统调度机制的比较

Tab.1 The difference of three embedded operating systems in scheduling mechanism

操作系统	$\mu\text{C}/$ OS-II	Windows CE	嵌入式 Linux
内核抢占	是	是	是
优先级变化	静态	动态	动态
调度算法	基于固定优先级抢占式调度	基于动态优先级抢占式调度	基于动态优先级抢占式调度:O(1)算法
同优先级调度	无	有	有
优先级数量	64	256(线程)	100
调度时间的可确定性	是	是	否

1.2 进程的同步与通信机制 进程之间为了保护共享资源需要互相同步和通信.同步通常是通过信号量(semaphore)、互斥型信号量(mutex)、事件标志(event flag)等方法来实现同步,而通信服务则是通过消息邮箱(message box)、消息队列(message query)、管道(pipe)和共享内存(shared memory)来提供^[3].

嵌入式操作系统中由于使用了互斥量等数据

结构,因此常常会面临优先级反转(priority inversion)问题.优先级反转是一种不确定的延迟形式,任何嵌入式操作系统都必须处理这一问题^[3].一般的例子是:一个高优先级任务等待一个互斥量,而这个互斥量正被一个低优先级任务所拥有,如果这时这个低优先级任务被一个或多个中等优先级的任务剥夺了运行,那么高优先级任务就必须等到中等优先级任务完成后才有可能继续执行,这样就发生了优先级反转,即这个高优先级任务被一个不相关的较低优先级任务阻止了执行,实时性难以得到保障^[6].

嵌入式操作系统应尽量避免出现优先级反转,通常可以采用优先级置顶协议(priority ceiling protocol)和优先级继承协议(priority inheritance protocol)这 2 种方法.优先级置顶协议是指所有获得互斥量的任务把它们的优先级提升到一个事先规定好的值.该方法的缺点是:需要事先知道使用这个互斥量的所有任务的最高优先级.如果这个事先规定的值太高,它相当于一个全局锁禁止了所有的调度.优先级继承协议是指拥有互斥量的任务的优先级被提升到与下一个在等待这个互斥量的最高优先级任务的优先级相等.该技术不需要预先知道准备使用这个互斥量的任务的优先级,当一个更高优先级任务处于等待时,只有拥有互斥量的任务的优先级被提升.该方法减少了对别的任务进行调度的影响,但是它增加了每次同步调用的开销^[3].

表 2 显示了 3 种嵌入式操作系统同步和通信机制的比较.

表 2 3 种嵌入式操作系统同步和通信机制的比较

Tab.2 The difference of three embedded operating systems in synchronization and communication mechanism

操作系统	$\mu\text{C}/$ OS-II	Windows CE	嵌入式 Linux
同步与通信机制	信号量、互斥量、邮箱、队列、事件标志等	事件、互斥变量、信号量等	原子操作、自旋锁、信号量、读/写锁、全局内核锁等
避免优先级反转方法	优先级置顶	优先级继承	优先级继承

1.3 存储器管理 存储管理也是嵌入式操作系统的重要内容之一,它的主要目标是通过存储资源的有效管理来提高存储器的利用率,为程序的并发运行提供一个良好的环境^[7].这里的存储器主要指的是内存,外存的管理并不包括在存储管理功能中.

$\mu\text{C}/\text{OS}-\text{II}$ 操作系统支持静态或动态的内存分配方式.它总是分配连续的存储空间,并且直接分配实存储器地址,没有虚拟存储管理,因此它不支持存储管理单元(MMU). $\mu\text{C}/\text{OS}-\text{II}$ 系统采用了分区的存储机制:为了消除多次动态分配与释放内存所引起的内存碎片的问题,把连续的大块内存按分区来管理,每个分区中都包含整数个大小相同的内存块,但不同分区之间内存块的大小可以不同^[8].利用这种机制, $\mu\text{C}/\text{OS}-\text{II}$ 对 `malloc()` 函数和 `free()` 函数进行了改造,使得它们可分配和释放固定大小的内存块,并且使这 2 个函数的执行时间也固定下来.当用户需要动态分配内存时,选择一个适当的分区,按块来分配内存.释放内存时将该块放回它以前所属的分区.

Windows CE 操作系统采用的是分页内存管理,使用虚拟存储方式来分配内存.内存管理的最小内存单元是页(1~4 KB).并且在虚拟内存之上提供了逻辑内存管理,逻辑内存管理的基本单位是堆,进程是堆分配的基本运行单位.Windows CE 系统通过 MMU 将虚拟地址转换为物理地址,它为全部应用程序实现的虚拟地址空间为 2 GB,每个应用程序实现的虚拟地址空间为 32 MB^[9].

嵌入式 Linux 系统中内存管理的基本单位也是页(大多数 32 位体系结构支持 4 KB 的页),它的不同之处在于,它将具有相似特性的页进行分区.Linux 中使用了 3 种区:ZONE-DMA(执行 DMA 操作的页),ZONE-NORMAL(正常映射的页),ZONE-HIGHMEM(“高端内存”)^[10].

因此,在内存分配函数中,设置了分配器标志.分配器标志有 3 种:行为修饰符,区修饰符和类型.通过这些标志来指导内核该从哪个区,使用什么方式来分配内存 Linux.

表 3 显示了 3 种嵌入式操作系统存储器管理机制的比较.

表 3 3 种嵌入式操作系统存储器管理机制的比较

Tab.3 The difference of three embedded operating systems in memory management mechanism

操作 系统	$\mu\text{C}/$ OS- II	Windows CE	嵌入式 Linux
支持 MMU	不支持	支持	支持
存储保护	无	有	有
存储机制	分区	分页	分页
分配方式	静态或动态分配连续区	动态分配虚拟页,实际操作时为进程分配堆空间	动态分配虚拟页,对相似页进行分区管理

2 适应性分析

这 3 种嵌入式操作系统在实际中都有很多的商业应用,通过上面的分析,我们可以看到它们的一些特点.在了解这些特点后,我们就可以对这 3 种嵌入式操作系统进行适应性分析.主要是对稳定性,可移植性,系统定制能力,开发难度及可利用资源,成本等选型原则进行讨论,从而得到这 3 种嵌入式操作系统的适用范围.

(1) 从前 1 节的分析我们可以看到, $\mu\text{C}/\text{OS}-\text{II}$ 是一个结构非常简单、功能完备和实时性很强的嵌入式操作系统内核.它最大的特点是小巧,非常适合应用在一些 ROM 和 RAM 都十分有限的嵌入式系统中,如:单片机系统^[11]. $\mu\text{C}/\text{OS}-\text{II}$ 绝大部分源代码是用 C 语言写的,并且公开源代码,因此具有非常好的可移植性.由于 $\mu\text{C}/\text{OS}-\text{II}$ 仅是一个实时内核,这就意味着它不像其它操作系统那样能够提供给用户一些 API 函数接口,因此可利用的资源比较有限,还有很多工作需要用户自己去完成.另外, $\mu\text{C}/\text{OS}-\text{II}$ 对网络通信的支持并不是很好^[1],因此并不适用于构建一个和网络应用密切相关的嵌入式系统.

(2) Windows CE 与 Windows 系列有很好的兼容性,它使用一个缩减的 WIN32 API,非常容易给熟悉 Windows 编程的开发者使用.Windows CE 主要针对小容量、移动式、智能化、模块化的实时嵌入式应用.它为建立针对掌上设备、无线设备的动态应用程序和服务提供了一种功能丰富的操作系统平台.Windows CE 能在多种处理器体系结构上

运行,具有比较好的系统移植性.它最大的特点是它有非常好的图形用户界面,开发速度比较快,有强大的通信能力,因此非常适合用于信息家电,移动计算领域^[9].但是它没有开放源代码,使应用开发人员很难实现产品的完美定制.另外在效率、功耗方面的表现并不出色,而且占用较多的系统内存,应用程序庞大.此外,使用 Windows CE 来开发嵌入式系统需要向微软公司为每一例应用交纳许可证费用,这使得使用 Windows CE 的嵌入式产品难以实现低成本.

(3) 嵌入式 Linux 和普通 Linux 一样拥有一个稳定、高效和安全的内核.稳定是 Linux 操作系统本身具备的一个很大优点.正因为 Linux 系统非常健壮及安全,所以嵌入式 Linux 很适合于那些长时间工作以及对可靠性有较高要求的领域.另一方面,嵌入式 Linux 系统又非常小巧灵活,易于裁剪. Linux 是源代码公开的系统,它在全世界有着广泛的开发者群体,能够对大量的硬件和外设芯片进行支持.此外,它无须交纳许可证费,能够极大减少使用成本.因此适合开发对实时性要求不高的小容量、低成本的各种产品,特别适用于开发与网络应用密切相关的嵌入式设备.但是 Linux 做为嵌入式操作系统也有一些不足的地方.首先 Linux 不是一个实时的操作系统,如果要提供实时功能需要添加实时软件模块,而这些模块运行的内核空间正是操作系统实现调度策略、硬件中断异常和执行程序的部分.由于这些实时软件模块是在内核空间运行

的,因此代码错误可能会破坏操作系统从而影响整个系统的可靠性,这对于实时应用将是一个非常严重的弱点.

参考文献:

- [1] 宋延昭. 嵌入式操作系统介绍及选型原则[J]. 工业控制计算机, 2005, 18(7): 41-42.
- [2] 陈向群, 王雷, 等. Windows CE NET 系统分析及实验教程[M]. 北京: 机械工业出版社, 2003.
- [3] 季志均, 马文丽, 陈虎, 等. 4 种嵌入式实时操作系统关键技术分析[J]. 计算机应用研究, 2005(9): 4-8.
- [4] Christian M NETTER, Luiz F BACELLAR. Assessing the real-time properties of Windows CE 3.0[J]. IEEE International Symposium on Object-Oriented Real-time Distributed Computing, 2001.
- [5] Robert LOVE. Linux kernel development[M]. Sams Press, 2003.
- [6] 李飞. 实时操作系统的比较[J]. 电子世界, 2003(10): 26-27.
- [7] 郑灵翔. 嵌入式系统设计与应用开发[M]. 北京: 北京航空航天大学出版社, 2006.
- [8] 任哲. 嵌入式实时操作系统 $\mu\text{C}/\text{OS}-\text{II}$ 原理及应用[M]. 北京: 北京航空航天大学出版社, 2005.
- [9] 田东风. Windows CE 应用程序设计[M]. 北京: 机械工业出版社, 2003.
- [10] BOKHARI S H. The Linux operating system[J]. Computer, 2001(8): 74-79.
- [11] 吴非, 樊晓光. 嵌入式实时操作系统 $\mu\text{C}/\text{OS}-\text{II}$ 与 eCos 的比较[J]. 单片机与嵌入式系统设计, 2004, (10): 15-17.

Analysis of key techniques based on three embedded operating systems

JIN Liang, ZHANG Xue-jie

(Department of Computer Science and Engineering, Yunnan University, Kunming 650091, China)

Abstract: Kernels are the key components of the operating systems. Task scheduling, task synchronizing and communicating, memory allocating are the essential techniques of realizing kernel. These key techniques of three operating systems including the $\mu\text{C}/\text{OS}-\text{II}$, Windows CE and embedded Linux were compared and analyzed, and their differences were discussed. Some selecting principles of embedded operating systems on the basis of thorough analysis of applicability were proposed.

Key words: embedded system; $\mu\text{C}/\text{OS}-\text{II}$; Windows CE; embedded Linux