

高手进阶2.6内核的Linux嵌入式系统应用 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/253/2021_2022__E9_AB_98_E6_89_8B_E8_BF_9B_E9_c103_253077.htm

随着多媒体技术与通讯技术相结合的信息技术的快速发展和互联网的广泛应用,PC时代也过渡到了后PC时代。在数字信息技术和网络技术高速发展的后PC时代,嵌入式技术越来越与人们的生活紧密结合。操作系统为用户使用计算机及其外部设备提供最基本的接口程序,管理计算机上的资源。随着应用领域的扩大,为了适应不同的应用场合,考虑到系统的灵活性、可伸缩性以及可裁剪性,一种以应用为中心、以计算机技术为基础、软硬件可裁剪、适应应用系统对功能、可靠性、成本、体积、功耗要求严格的专用计算机系统嵌入式操作系统随之诞生。Linux操作系统是一种性能优良、源码公开且被广泛应用的免费操作系统,由于其体积小、可裁减、运行速度快、良好的网络性能等优点,可以作为嵌入式操作系统。随着2.6内核的发布,Linux向现有主流的RTOS提供商在嵌入式系统市场提出了巨大挑战,例如VxWorks和WinCE,具有许多新特性,将成为更优秀的嵌入式操作系统。Linux的低成本和开放性,为其在嵌入式系统领域的应用营造了肥沃的土壤。本文着重介绍Linux 2.6内核的新特性及其嵌入式应用中的优势,并将其移植到嵌入式平台中,成功支持H.264编解码多媒体系统。

1 Linux 2.6内核针对嵌入式开发显著特点

实时可靠性是嵌入式应用较为普遍的要求,尽管Linux 2.6并不是一个真正的实时操作系统,但其改进的特性能够满足响应需求。Linux 2.6已经在内核主体中加入了提高中断性能和调度响应时间的改进,其中有三个最显著

的改进:采用可抢占内核、更加有效的调度算法以及同步性的提高。在企业服务器以及嵌入式系统应用领域,Linux 2.6 都是一个巨大的进步。在嵌入式领域,Linux 2.6 除了提高其实时性能,系统的移植更加方便,同时添加了新的体系结构和处理器类型包括对没有硬件控制内存管理方案的 MMU-less 系统的支持,可以支持大容量内存模型、微控制器,同时还改善了 I/O 子系统,增添更多的多媒体应用功能 < 4 >。

1.1 可抢占内核

在先前的内核版本中(包括2.4内核)不允许抢占以核心态运行的任务(包括通过系统调用进入内核模式的用户任务),只能等待它们自己主动释放CPU。这样必然导致一些重要任务延时以等待系统调用结束。一个内核任务可以被抢占,为的是让重要的用户应用程序可以继续运行。这样做最主要的优势是极大地增强系统的用户交互性。

2.6内核并不是真正的RTOS(Real Time Operation System)

其在内核代码中插入了抢占点,允许调度程序中止当前进程而调用更高优先级的进程,通过对抢占点的测试避免不合理的系统调用延时。2.6内核在一定程度上是可抢占的,比2.4内核具备更好的响应性。但也不是所有的内核代码段都可以被抢占,可以锁定内核代码的关键部分,确保CPU的数据结构和状态始终受到保护而不被抢占。软件需要满足最终时间限制与虚拟内存请求页面调度之间是相互矛盾的。慢速的页错误处理将会破坏系统的实时响应性,而2.6内核可以编译无虚拟内存系统避免这个问题,这是解决问题的关键,但要求软件设计者有足够的内存来保证任务的执行。

1.2 有效的调度程序

版本是2.6的 Linux内核使用了由 Ingo Molnar开发的新的调度器算法,称为O(1)算法,如图1所示。它在高负载情况下执行得极其出色,并且当有很多处理器并行时也可以很好地扩

展。过去的调度程序需要查找整个ready task队列,并且计算它们的重要性以决定下一步调用的task,需要的时间随task数量而改变。 $O(1)$ 算法则不再每次扫描所有的任务,当task就绪时被放入一个活动队列中,调度程序每次从中调度适合的task,因而每次调度都是一个固定的时间。任务运行时分配一个时间片,当时间片结束,该任务将放弃处理器并根据其优先级转到过期队列中。活动队列中任务全部调度结束后,两个队列指针互换,过期队列成为当前队列,调度程序继续以简单的算法调度当前队列中的任务。这在多处理器的情况更能提高SMP的效率,平衡处理器的负载,避免进程在处理器间的跳跃。

1.3 同步原型与共享内存

多进程应用程序需要共享内存和外设资源,为避免竞争采用了互斥的方法保证资源在同一时刻只被一个任务访问。Linux内核用一个系统调用来决定一个线程阻塞或是继续执行来实现互斥,在线程继续执行时,这个费时的系统调用就没有必要了。Linux2.6所支持的Fast User-Space Mutexes 可以从用户空间检测是不是需要阻塞线程,只在需要时执行系统调用终止线程。它同样采用调度优先级来确定将要执行的进程。多处理器嵌入式系统各处理器之间需要共享内存,对称多处理技术对内存访问采用同等优先级,在很大程度上限制了系统的可量测性和处理效率。Linux2.6则提供了新的管理方法NUMA(Non Uniform Memory Access)。NUMA根据处理器和内存的拓扑布局,在发生内存竞争时,给予不同处理器不同级别权限以解决内存抢占瓶颈,提高吞吐量。

1.4 POSIX线程及NPTL

新的线程模型基于一个1:1的线程模型(一个内核线程对应一个用户线程),包括内核对新的 NPTL(Native POSIX Threading Library)的支持,这是对以前内核线程方法的明显改

进。2.6内核同时还提供POSIX signals和POSIX high-resolution timers。POSIX signals不会丢失,并且可以携带线程间或处理器间的通信信息。嵌入式系统要求系统按时间表执行任务,POSIX timer可以提供1kHz的触发器使这一切变得简单,从而可以有效地控制进度。

1.5 微控制器的支持 Linux2.6内核加入了多种微控制器的支持。无MMU的处理器以前只能利用一些改进的分支版本,如uClinux,而2.6内核已经将其整合进了新的内核中,开始支持多种流行的无MMU微控制器,如Dragonball、ColdFire、Hitachi H8/300。Linux在无MMU控制器上仍旧支持多任务处理,但没有内存保护功能。同时也加入了许多流行的控制器的支持,如S3C2410等。

100Test 下载频道开通,各类考试题目直接下载。详细请访问 www.100test.com