

2.6 有关进程和线程的研究

在第1章里,我们介绍了有关操作系统结构的当前研究工作。在本章和下一章里,我们将更专注于有关进程的研究。随着时间推移,一些问题会比其他问题解决得更好。多数研究倾向于从事新的课题,而不是围绕着有数十年历史的题目进行研究。

作为一个例子,关于进程概念的研究已经获得良好的解决方案。几乎所有的系统都把一个进程视为一个容器,该容器用以聚集相关的资源,如地址空间、线程、打开的文件、保护许可等。不同的系统聚集资源的方式略有差别,但是差别仅在于工程处理方面。基本思想不会有较大的争议,且有关进程的课题也几乎没有新的研究在进行。

线程是比进程更新的概念,但是它们同样也经过了相当多的考虑。仍然偶尔会出现关于线程的论文,例如,关于在多处理器上的线程集群(Tam等人,2007)或是一个进程中的线程数量如何扩展到100 000(Von Behren等人,2003)。

现在,进程同步问题已经相当成熟和固定,但是每隔一段时间还是会有一篇论文,例如关于无锁并发处理的问题(Fraser和Harris,2007)或是实时系统中的无阻塞同步问题(Hohmuth和Haertig,2001)。

调度(单处理器和多处理器)还是—些研究者感兴趣的话题。—些正在研究的主题包括移动设备上的能耗节省调度(Yuan和Nahrstedt,2006)、超线程级调度(Bulpin和Pratt,2005)、当CPU空闲时该做什么(Eggert和Touch,2005)以及虚拟时间调度(Nieh等人,2001)。但是,很少有实际系统的设计者会因为缺乏像样的线程调度算法而整天苦恼,所以这似乎是一个由研究者推动而不是需求推动的研究类型。总而言之,进程、线程与调度不像它们曾经那样,是研究的热点。这些研究已经前进得很多了。

2.7 小结

为了隐蔽中断的影响,操作系统提供了一个由并行运行的顺序进程组成的概念模型。进程可以动态地创建和终止。每个进程都有自己的地址空间。

对于某些应用而言,在一个进程中使用多个控制线程是有益的。这些线程被独立调度,每个线程有自己的堆栈,但是在一个进程中的所有线程共享一个公共地址空间。线程可以在用户空间或内核中实现。

进程之间通过进程间通信原语彼此通信,如信号量、管程或消息。这些原语用来确保同一时刻不会有两个进程在临界区中,免除了出现混乱的情形。进程可以处在运行、可运行或阻塞状态,并且在该进程或其他进程执行某个进程间通信原语时,可以改变其状态。线程间通信也是类似的。

进程间通信原语可以用来解决诸如生产者—消费者问题、哲学家就餐问题和读者—写者问题等。即使有了这些原语,也要仔细设计以避免出错和死锁。

已经有一大批研究出来的调度算法。某些算法主要用于批处理系统中,如最短作业优先调度算法。其他算法常用在批处理系统和交互式系统中,它们包括轮转调度、优先级调度、多级队列、保证调度、彩票调度以及公平分享调度等。有些系统将调度策略和调度机制清晰地分离,这样可以使用户对调度算法进行控制。

习题

1. 图2-2中给出了三个进程状态。在理论上,三个状态可以有六种转换,每个状态两个。但是,图中只给出了四种转换。有没有可能发生其他两种转换中的一个或两个?
2. 假设要设计一种先进的计算机体系结构,它使用硬件而不是中断来完成进程切换。CPU需要哪些信息?请描述用硬件完成进程切换的工作过程。
3. 在所有当代计算机中,至少有部分中断处理程序是用汇编语言编写的。为什么?
4. 当中断或系统调用把控制转给操作系统时,通常将内核堆栈和被中断进程的运行堆栈分离。为什么?
5. 多个作业能够并行运行,比它们顺序执行完成的要快。假设有两个作业同时开始执行,每个需要10分钟的CPU时间。如果顺序执行,那么最后一个作业需要多长时间可以完成?如果并行执行又需要多长时间?假设I/O等待占50%。
6. 在本章中说明的图2-11a的模式不适合用于使用内存高速缓存的文件服务器。为什么不适合?每个进程可以有自己的高速缓存吗?
7. 如果创建一个多线程进程,若子进程得到全部父进程线程的副本,会出现问题。假如原有线程