

上下文,当纳线程进出处理器时进行切换。每个纳线程和进程一样具有一个状态。对于纳线程来说,关键是纳核对它们的紧密控制,以及构成每个纳线程上下文的极小数据集。

Symbian操作系统线程依赖于纳线程,内核增加除纳核提供的功能之外的支持。标准程序使用的用户模式线程由Symbian操作系统线程执行。每个Symbian操作系统线程包含一个纳线程并且添加自己的运行时刻栈到纳线程使用的栈中。Symbian操作系统线程可以通过系统调用在内核模式下进行操作。Symbian操作系统也能为执行增加例外处理以及退出信号。

Symbian操作系统线程在纳线程实现之上实现自己的状态集。由于Symbian操作系统线程将一些功能性增加到纳线程实现中,因此新的状态反映了构成Symbian操作系统线程的新的思想。Symbian操作系统添加了Symbian操作系统线程可以进入的新的七种状态,来关注Symbian操作系统线程可能出现的特殊阻塞条件。这些特殊状态包括在信号量上的等待和挂起(正常的)、互斥变量以及条件变量。由于Symbian操作系统的实现处于纳线程之上,因此这些状态从某种方面上来说是由纳线程状态实现的,通常都是用不同的方式使用挂起的纳线程状态。

12.3.2 进程

Symbian操作系统的进程,就是在单一进程控制块结构下,具有单一存储空间,归于一类的Symbian操作系统的线程组。可能只有一个执行的线程,或者一个进程控制块下有很多线程。Symbian操作系统线程和纳线程已经定义了进程状态和进程调度的概念。因此,调度一个进程实际上是通过调度一个线程以及初始化数据需要使用的正确的进程控制块来完成的。

Symbian操作系统线程通过几种方式,在一个单一进程的组下工作在一起。首先,有一个主线程被标志为进程的起始点。其次,线程共享调度参数。也就是说,进程通过一种调度方法——改变进程参数,来改变所有线程的参数。第三,线程共享包括设备和其他对象描述符的存储空间对象。最后,当一个进程终止时,内核终止该进程的所有线程。

12.3.3 活动对象

活动对象是线程的特有形式,用这种方式实现以便减轻它们带给操作环境的负担。Symbian操作系统的设计者意识到,应用中的线程在很多情况下可能会发生阻塞。由于Symbian操作系统致力于通信工具方面,因此许多应用程序具有类似的执行模式:它们向一个通信套接字写数据或者通过管道发送信息,然后在等待接收者的响应时阻塞。这样设计活动对象,是为了当它们从这种阻塞状态返回时,具有进入被调用代码的单一入口点,这简化了它们的实现。由于活动对象运行在用户空间,因此它们具有Symbian操作系统线程的特性。它们本身具有自己的纳线程,并且能够加入Symbian操作系统的其他线程构成操作系统的进程。

假若活动对象仅仅是Symbian操作系统线程,有人就会问操作系统从这种简化的线程模型中得到了什么益处。活动对象的关键点体现在调度上。所有的活动对象在等待事件的时候驻留在一个单一进程中,对系统而言可以作为一个单一的线程。内核不必连续地检查每一个活动对象是否被解除阻塞。因此,单一进程中的活动对象,能够由在一个单一线程中执行的单一调度器来协调。通过将其他方面作为多线程执行的代码结合到一个线程中,通过构建固定的入口点进入代码,以及通过使用一个单一调度器来协调它们的执行,活动对象构成了标准线程的一种高效、轻量版本。

认识到活动对象和Symbian操作系统进程结构在何处融合成为一体是很重要的。当一个传统线程通过系统调用进入等待状态从而阻塞自己的运行时,操作系统仍然需要检查这个线程。在上下文切换期间,操作系统需要花费时间检查处于等待状态的阻塞进程,决定是否需要将其移动到就绪状态。活动对象把自己放入等待状态以等待特定的事件,因此,操作系统不需要去检查它们,而只是在特定的事件发生后移动它们。结果就是更少的线程检测以及更好的性能。

12.3.4 进程间通信

在类似Symbian操作系统的多线程环境中,进程间通信对系统性能是至关重要的。线程,特别是系统服务形式的线程经常通信。

套接字是Symbian操作系统使用的基本通信模型。它是两个端点之间抽象的通信管道。这一抽象是