

内核扩展是引导过程的一部分。这些特殊的设备驱动在调度器启动之后加载并且启动。它们执行对于操作系统非常重要的功能: DMA服务、显示管理、对外设的总线控制(例如USB总线)。之所以提供它们有两个原因。首先, 它与我们已经看作是微内核设计特征的面向对象设计抽象相称。其次, 它允许Symbian操作系统所处的不同平台运行专门的设备驱动, 从而不需要重新编译内核而使用硬件。

### 12.5.3 直接存储器访问

设备驱动经常使用DMA, Symbian操作系统支持DMA硬件的使用。DMA硬件包含一个控制一系列DMA通道的控制器。每个通道提供内存和设备间的单一方向的通信, 因此, 数据的双向传输需要两个DMA通道。至少有一对DMA通道是专用于显示LCD控制器的。此外, 大多数平台提供一定数量的常规DMA通道。

一旦一个设备把数据传送到内存, 就会激发一个系统中断。PDD为了传输设备使用DMA硬件提供的DMA服务, 这里传输设备是指与硬件接口的设备驱动的一部分。在PDD与DMA控制器之间, Symbian操作系统实现两层软件: 一个软件的DMA层, 一个与DMA硬件接口的内核扩展。DMA层把自身分成平台独立层和平台相关层。作为内核扩展, DMA层在引导进程中是内核启动的第一批设备驱动的一个。

由于特殊的原因, 对DMA的支持是比较复杂的。Symbian操作系统支持许多不同的硬件配置, 但是没有提供缺省的DMA配置。与DMA硬件的接口是标准化的, 由平台无关层来提供。平台相关层和内核扩展由生产厂商提供, 这样Symbian操作系统就如对其他设备一样处理DMA硬件: 在LDD和PDD构件中具有设备驱动。由于DMA硬件本身是一个设备, 并且它并行了Symbian操作系统支持所有设备的方式, 因此这种实现支持的方式是合理的。

### 12.5.4 特殊情况: 存储介质

Symbian操作系统中存储介质驱动是PDD的一种特殊形式, 文件服务器排他地使用它们来实现对存储介质设备的访问。因为智能手机既可以容纳固定的存储介质也可以容纳移动的存储介质, 所以存储介质驱动必须识别和支持多种形式的存储介质。Symbian操作系统对介质的支持包括一个标准的LDD和为用户提供的接口API。

Symbian操作系统中的文件服务器能够同时支持多达26个不同的设备。本地设备, 像在Windows中一样, 通过驱动器号来区分。

### 12.5.5 阻塞I/O

Symbian操作系统通过活动对象处理阻塞I/O。设计者认识到等待I/O事件的所有线程的负荷会影响系统中的其他线程这一事实。活动对象使得阻塞I/O调用可以由操作系统来处理, 而不是进程自身。活动对象由一个调度器进行协调并且在一个单独的线程中执行。

当活动对象使用一个阻塞I/O调用时, 它用信号通知操作系统并且把自身挂起。当调用完成时, 操作系统唤醒排起的进程, 该进程如同带有数据返回的函数一样继续执行。区别只是对于活动对象的一个观点: 它不能调用一个函数并期待一个返回值; 它必须调用一个特殊的函数并且使该函数设置阻塞I/O, 但是立刻返回。操作系统接管等待过程。

### 12.5.6 可移动存储器

可移动存储器带给操作系统设计人员一个有趣的两难处境。当往读取槽插入一张安全数据(Secure Digital, SD)卡时, 该卡就同其他设备一样成为一个设备。它需要一个控制器、一个驱动、一种总线结构, 而且很有可能通过DMA与CPU进行通信。然而, 对这类模型移除存储介质是一个很严重的问题: 操作系统怎样检测插入和移除? 这一模型如何适应一张介质卡的不存在? 还有更加复杂的情况, 一些设备槽能够兼容不止一种类型的设备。例如, 一张SD卡, 一张miniSD卡(带有适配器), 以及一张多媒体卡都使用同一类插槽。

Symbian操作系统使用可移动存储器的很多共同性来实现对它们的支持。每种可移动存储器通常具有如下特点:

- 1) 所有的设备必须支持插入和移除。