

- 内存管理：尽管Symbian操作系统不使用（也没有相应的设备）虚拟内存映射，但它通过按页管理实现对内存访问，并支持页的置换，也就是说支持页面换入，但不支持页面换出。

### 12.2.6 通信与多媒体

Symbian操作系统以多种方式协助通信。我们很难简单地对其进行概述而不提及通信特点。通信的模型遵循面向对象、微内核以及客户机/服务器结构。Symbian操作系统的通信结构是以模块形式建立的，从而允许新的通信机制方便地接入操作系统。从用户层接口到新的协议实现，到新的设备驱动，模块可以实现任何功能。由于这样的微内核实现，可以引入新的模块并动态地加载到系统操作中。

由于Symbian操作系统只针对智能手机平台，因此有许多独特的特点。它具有一个可插拔的消息结构，可以引入新的消息类型，并可以通过消息服务器动态加载的模块来实现。消息系统被设计为层次结构，各层由特定类型的对象来实现。例如，消息传递对象不同于消息类型对象。一种形式的消息传递，例如手机无线传递（类似于CDMA）可以传送几种不同类型的消息（标准文本消息类型、SMS类型或者如BIO消息等的系统指令）。通过实现新的对象并将其加载到内核中可以引入新的传递方法。

Symbian操作系统的核心设计有专门针对多媒体的各种API。多媒体设备以及上下文由特殊的服务器和用户自定义的结构（允许用户实现描述新的或现存的上下文以及怎样对上下文进行处理的模块）来处理。与实现消息的方法相类似，多媒体是由多种形式相互作用的对象来实现。声音播放的方式被设计为一个与各种声音格式的实现方式相互作用的对象。

## 12.3 Symbian操作系统中的进程和线程

Symbian操作系统是一个多任务操作系统，像其他操作系统那样，使用了进程和线程的概念。然而，Symbian操作系统的内核结构以及它对资源稀缺性的处理方式影响了它看待这些多任务对象的方式。

### 12.3.1 线程和纳线程

对于多任务，Symbian操作系统更倾向于线程，并且是建立在线程概念上的，而不是把进程作为多任务的基础。线程构成了多任务的中心单元。操作系统简单地把一个进程看成是具有一个进程控制块和某个内存空间的线程的集合。

Symbian操作系统对于线程的支持是基于纳线程的纳核。纳核仅提供简单的线程支持，每个线程是由一个基于纳核的纳线程来支持的。纳核为纳线程提供调度、同步（线程间通信）以及计时服务。纳线程运行在特权模式下，需要一个栈来存储它们的运行时刻环境数据。纳线程不能运行在用户态。这就意味着操作系统能够对每个纳线程保持紧密的控制。每个纳线程需要一个数据的极小集来运行：实质上就是它的栈以及栈的大小。操作系统保持对其他一切的控制，比如每个线程使用的代码，以及在运行时刻的栈上存储线程的上下文。

同进程具有状态一样，纳线程也具有线程状态。Symbian操作系统的纳核使用的模型在基本模型中增加了一些状态。除了基本状态以外，纳线程还可以处于如下状态：

- 挂起。这就是当一个线程挂起另一个线程时的状态，与等待状态不同，在等待状态下一个线程是被某个上层对象阻塞（例如，一个Symbian操作系统线程）。
- 快速信号量等待。处于这个状态的线程正在等待一个快速信号量（哨兵变量的一种）得到信号通知。快速信号量是纳核级别的信号量。
- DFC等待。处于这种状态的线程正在等待一个延迟的函数调用或者要被加入到DFC队列中的DFC。DFC用在设备驱动实现中。它们代表对于内核的调用，可被Symbian操作系统内核层排入队列并且调度执行。
- 休眠。休眠线程正在等候特定长度的时间过去。
- 其他。还有一种通用状态，是当开发人员为纳线程实现额外的状态时使用的。当开发人员为新的手机平台（称作个性层）扩展纳核功能时使用该状态。进行这个工作的开发人员也必须实现这些状态与他们的扩展实现之间的来回跳转。

下面将纳线程思想与传统进程思想作比较。纳线程实际上是一个完全轻量级的进程。它具有极小的