

建立了相应的机制，而这些专门的模块称为信息类型模块（MTM模块）。短信息处理包括多个方面，MTM模块需要处理所有这些需求。用户界面类MTM模块需要实现多种供用户查看和处理短信息的方式，包括如何阅读短信息，如何被告知短信息发送进度等。客户端MTM模块处理寻址、创建、回复短信息；而服务器端MTM模块需要实现面向服务器的相关短信息管理功能，如目录管理、特定信息的管理等。

根据所使用通信类型的不同，这些模块以不同的方式彼此依赖。例如，实现使用蓝牙的协议，我们只需要物理器件上层的PRT模块即可。某些IrDA协议也是如此。而基于PPP的TCP/IP实现则需要使用PRT模块、TSY模块和CSY模块；不基于PPP的TCP/IP协议则不需要TSY模块和CSY模块，但是其PRT模块需要直接连接到网络设备驱动上。

4. 结构模块化

基于模块化的思想在这样一个栈式的模型实现中是很有用的。在这个分层的设计中，从例子中可以看出，抽象带来的优势是很明显的。考虑TCP/IP协议的实现。一个PPP连接既可以直接使用CSY模块，也可以选择GSM或普通调制解调器的TSY实现，后者实际底层仍由CSY模块来实现。未来新的电话技术出现后，当前的结构仍然可以起作用，我们只需要为新的电话实现添加一个TSY模块。另外，细调TCP/IP协议栈不需要修改任何其依赖的模块，只需要简单地调整TCP/IP PRT模块。这样广泛的模块化意味着在已有结构上很容易添加新代码、丢弃旧代码，当前代码的修改不会对整个系统带来巨大的变化，也不需要大量的重新安装。

最后，图12-5在应用层添加了子层。应用程序通过CSY模块和协议实现层中的协议模块进行交互。虽然我们可以认为这些模块属于协议实现层的一部分，但更清晰的表示是，这些模块在协助应用程序进行操作。例如，在使用红外接口将短信息发送到手机的过程中，应用程序会在应用程序中使用IRCOMM CSY模块，通过协议实现层的短信息实现模块来完成。同样，在这样一个过程中，模块化带来了很大的优势，应用程序可以关注实现其擅长的功能，而不是通信过程。

12.9 小结

Symbian操作系统是一个为智能手机平台设计的面向对象的操作系统。它的微内核设计只提供了很小的纳核，只实现了最快和最简单的内核功能。Symbian操作系统通过客户机/服务器的体系结构，将对系统资源的访问分配给用户态的服务器。Symbian操作系统虽然是为智能手机设计的，但其也拥有很多通用操作系统的特性：进程和线程、内存管理、文件系统支持、丰富的通信支持。同时，Symbian操作系统也实现了一些独特的特性，比如，活动对象使等待外部事件更为迅速、没有虚拟内存使得内存管理更富有挑战性、支持面向对象的设备驱动程序采用双层抽象结构。

习题

1. 对下列的每一个服务，描述其在如Symbian操作系统这样的微内核操作系统中，是在用户态还是内核态执行。
 - 调度线程的执行。
 - 打印一个文档。
 - 应答蓝牙搜索信号。
 - 管理线程对屏幕的访问。
 - 在短信息到达时发出声音。
 - 中断当前执行并接听电话。
2. 列举微内核设计带来的三个效率提升。
3. 列举微内核设计带来的三个效率问题。
4. Symbian操作系统将其内核分割为纳核和Symbian内核两部分。如动态内存管理之类的服务被认为过于复杂而不能进入纳核。描述动态内存管理中的复杂模块，解释为什么不能将其放进微内核。
5. 我们讨论过，活动对象使得I/O操作更有效率。你认为应用程序是否能够同时使用多个活动对象？系统在多个I/O事件发生时如何响应？
6. Symbian操作系统中的安全是否关注软件安装和应用程序的Symbian签名？这是否足够安全？是否会有某个场景，应用程序可以不必安装即被运行？（提示：考虑手机数据输入的所有可能方式）
7. 在Symbian操作系统中，广泛应用了基于服务的对共享资源的保护。列举三种在微内核环境下，这种方式协调资源的优势。思考这些优势对不同体系结构的影响。