

URB(USB Request Block, USB 请求块)的 IRP(I/O Request Packet, I/O 请求包), 来实现对 USB 设备信息的发送和接收。采用这种分层驱动程序的设计方法有两个优点: (1)多个 USB 设备可以通过 USB 底层驱动程序来协调它们的工作; (2)编写分层驱动程序较之编写单一驱动程序相对简单, 且可以节省内存和资源, 不易出错。若应用程序想对设备进行 I/O 操作, 它需调用 Windows API 函数, I/O 管理器将此请求构造成一个合适的 I/O 请求包 IRP 并把它传递给 USB 功能驱动程序。USB 功能驱动程序接收到这个 IRP 以后, 根据 IRP 中包含的具体操作代码, 构造相应 USB 请求块并把此 URB 放到一个新的 IRP 中, 然后把此 IRP 传递 USB 底层驱动程序, USB 底层驱动程序根据 IRP 中所含的 URB 执行响应的操作(如从 USB 设备读取数据), 并把操作结果返还给 USB 功能驱动程序。USB 功能驱动程序接收到此 IRP 后, 将操作结果通过 IRP 返还给 I/O 管理器, 最后 I/O 管理器将此 IRP 操作结果返还给应用程序, 至此应用程序对 USB 设备的一次 I/O 操作完成。USB 功能驱动程序除负责处理应用程序的 I/O 请求外, 还要处理 PnP 请求(如设备启动请求 IRP\_MN\_START\_DEVICE, 设备删除 IRP\_MN\_REMOVE\_DEVICE 等)。通过对这些 PnP 请求的处理, USB 功能驱动程序可支持设备的热插拔和即插即用功能。

#### 4.1 USB 启动的过程

为了叙述清晰, 本章在介绍 USB 驱动程序设计之前, 先介绍 USB 的启动过程。USB 系统主要由主控制器(Host Controller)、USB Hub 和 USB 外设(Peripherals Node)组成系统拓扑结构, 如图 4-2 所示。

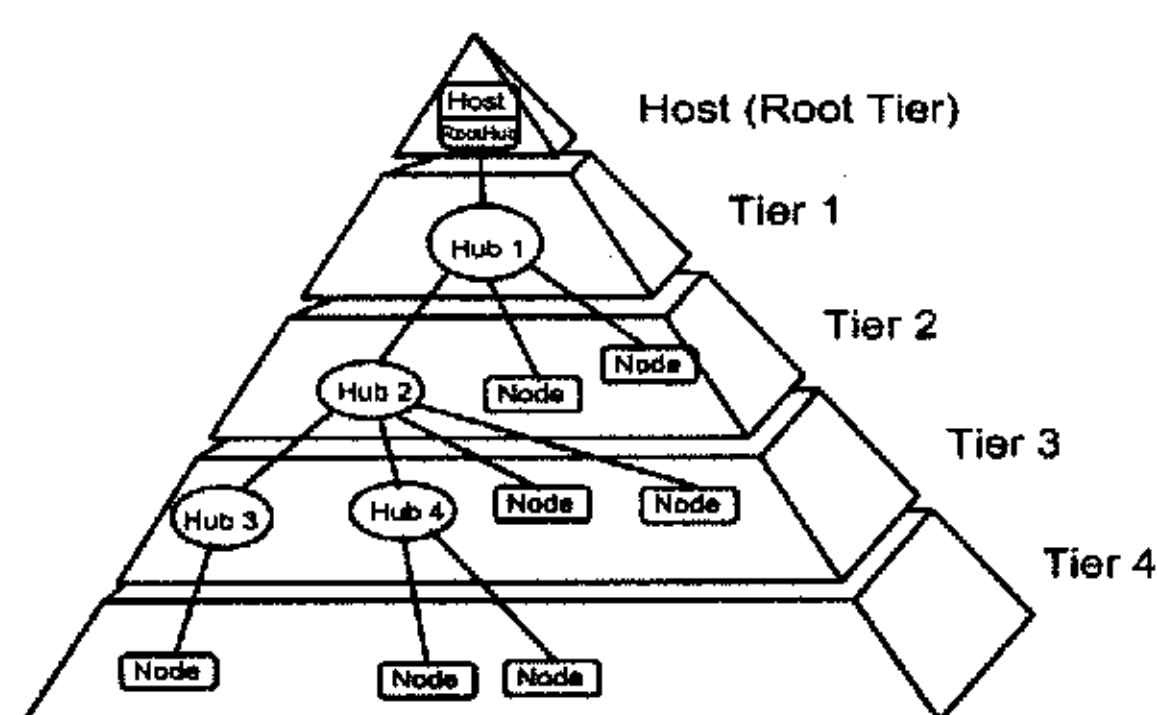


图 4-2 USB 系统拓扑结构

在应用程序可以与一个 USB 设备通信之前, 主机需要知道设备支持哪些传输类型和终端, 主机也必须分配一个地址给设备, 主机通过一个被称为枚举的信

息交换来完成这些工作。枚举过程：集线器的一个任务就是检测设备的连接与断开，每个集线器都有一个中断流程来通知主机报告这些事情。在系统启动的时候，主机查询它的根集线器来了解有哪些设备已经连接上了，包括其他集线器和连接到这些集线器的设备。在启动后，主机持续周期性地查询了解是否有设备连接或断开。一旦发现一个新设备，主机发送一系列的请求给这个设备的集线器，使这个集线器在主机和这个设备之间建立一个通信渠道。然后主机试图枚举这个设备，枚举是使得主机的设备驱动程序能与这个设备通信的最基本的信息交换。这个过程由如下动作组成：分配一个地址给设备，从设备读取描述数据，分配和载入一个设备驱动程序以及从接收到的数据中选择一个配置。然后设备就被配置完毕，并且准备好使用它的配置中支持的任何终端来传输数据。主机的枚举是通过给终端 0 发送包含标准 USB 请求的控制传输。所有的 USB 设备必须支持控制传输、标准 USB 请求和终端 0。对一个成功的枚举来说，设备必须对每一个请求响应并返回请求的信息。从用户的角度看，枚举应该是不可见和自动的，除了一些情况下如申明发现一个新设备和是否成功配置这个新设备的窗口，有时在第一次使用时，用户需要提供一个有 INF 文件和设备驱动程序的磁盘。当枚举结束时，Windows 把新的设备加入到控制面板的设备管理器显示中。应依次单击"开始"菜单(Start menu)->设置(Settings)->控制面板(Control Panel)->设备管理器(Device Manager),在设备管理器可以看到新加的设备。当一个用户断开一个外设的连接时，Windows 自动地从这个显示中移掉这个设备。在一个 USB 的外设中，外设的程序代码包含了主机将请求的信息，并且程序代码必须能识别和响应这些信息的请求。在 Windows 不需要编写枚举的程序，因为 Windows 自动处理枚举过程。Windows 将查找一个被称为 INF 文件的特殊文本文件，这个文件会告诉 Windows 哪个驱动程序适合这个设备。

## 4.2 USB 驱动程序结构

### 4.2.1 USB 接口软件结构

USB 接口软件结构如图 4-3 所示。PC 机底层驱动程序包括 HUB 驱动、总线类驱动和主机控制器驱动，它们负责处理总线枚举、电源管理以及 USB 事务的其它方面，这些驱动不需要编程者开发，Windows 操作系统提供这类驱动程序。编程者需要开发的程序为：USB 控制器固件程序，控制器的接收应用程序，设