

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

备功能驱动程序, PC 机应用程序

USB 控制器固件程序实现设备的枚举以及端点与主机的通信。控制器的接收应用程序接收主机下载的文件。设备功能驱动程序为应用程序和底层驱动程序之间提供接口。当一个应用程序启动一个 API 调用后, Windows 把调用传递给设备驱动程序, 设备驱动程序把请求传递到底层驱动程序, 底层驱动程序对硬件进行相应的操作。PC 机应用程序功能是实现主机文件的下载。

4.2.2 USB 驱动程序相关概念

设备: 这里的设备仅指在编写驱动程序时将设备看成一个整体。同一个设备可以有几种不同的配置

配置: 对设备的若干种配置方法中的一种, 在驱动程序中, 配置用一些结构来表示。从一个配置结构中, 可以知道设备有多少接口。

接口: 设备中功能相关或相近的一组端点的集合。在编写驱动程序时, 可以从接口描述符中获取相关信息。

端点: 从用户的角度看, 可以直接进行 IO 数据流操作的设备中的基本单位。端点是单向的, 如果要对设备进行双向的 IO, 必须至少两个端点。

管道: 一个端点与客户程序进行 IO 操作时使用的中介。管道与端点是一一对应的, 端点侧重于静态的概念, 管道侧重于动态的概念。

URB (USB Request Block): USB 请求块。对 USB 进行操作请求都应调用系统例程将其转化为一个 URB 结构, 然后使用系统级的 IRP 将其提交。

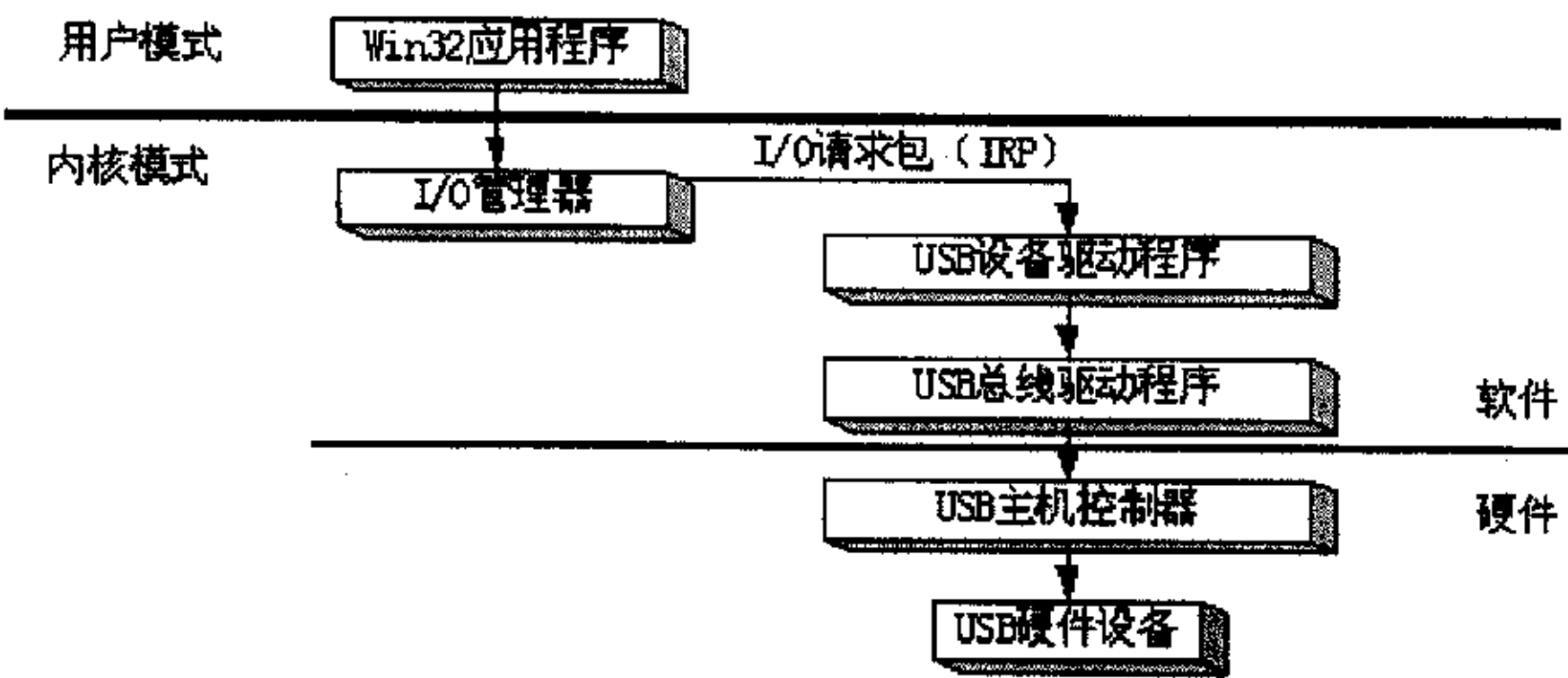


图 4-3 USB 接口软件结构

4.2.3 USB 驱动程序的入口

DriverEntry 是内核模式驱动程序主入口点常用的名字。I/O 管理器按下面