

### [转]驱动程序开发—概述(1)

这个系列的文章是我在学习驱动开发时在网络上找的资料，因为觉得写的不错就转在这了~

很久没有网了，出了一段时间的差，近来，莫名的就有点郁闷！前不久在大富翁上发了一份帖子是关于delphi程序员的发展，大家的反应并不都是很好。于是开始觉得可以考虑换个方向。以前我是做MIS开发的。换哪个方向呢？人越多的方向，好像越是没有前途。想想当初上大学，那可是越多人考的学校，学费越贵啊！可现在的职业呢？越多人干的事，越是没有前途了。考虑来考虑去，决定学习一下驱动程序的开发吧！于是从网上查找了一些资料，看懂的觉得蛮不错适合我这种小学生的就贴了出来，算是学习笔记吧！

#### 用户模式与内核模式

从Intel80386开始，出于安全性和稳定性的考虑，该系列的CPU可以运行于ring0~ring3从高到低四个不同的权限级，对数据也提供相应的四个保护级别。运行于较低级别的代码不能随意调用高级别的代码和访问较高级别的数据，而且也只有运行在ring0层的代码可以直接对物理硬件进行访问。由于WindowsNT是一个支持多平台的操作系统，为了与其他平台兼容，它只利用了CPU的两个运行级别。一个被称为**内核模式**，对应80x86的ring0层，是操作系统的核心部分，设备驱动程序就是运行在该模式下；另一个被称为**用户模式**，对应80x86的ring3层，操作系统的用户接口部分（就是我们通常所说的win32 API）以及所有的用户应用程序都运行在该级别。操作系统对运行在内核模式下的代码是不设防的，所以不管是建设还是破坏内核模式下的编程都是值得去研究的。

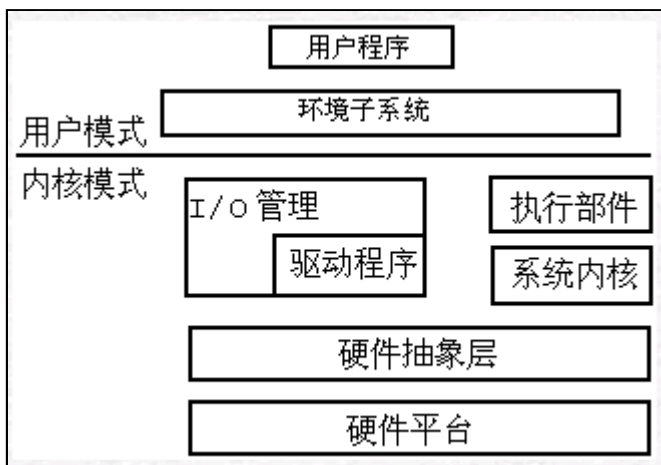


图1-WIN2000系统的分层结构

在物理硬件与系统核心之间有一个硬件抽象层（HardwareAbstractionLayer），它屏蔽了不同平台硬件的差异，向操作系统的上层提供了一套统一的接口。从图中我们还可以看到，设备驱动程序（DeviceDriver）是被I/O管理器（I/OManager）包围起来的，即驱动程序与操作系统上层的通信全部都要通过I/O管理器。这给驱动程序的编写带来了很大的便利，因为很多诸如接收用户的请求、与用户程序交换数据、内存映射、挂接中断、同步等等麻烦的工作都由I/O管理器代劳了。

#### 驱动程序的分类

驱动程序并不像所有人想的那样一定要和硬件打交道，我粗略的把他分为两类：硬驱动和软驱动。硬驱动就是对硬件直接编程进行控制，这类驱动通常必须遵守硬件的通信协议，直接对硬件进行端口访问、中断响应、DMA传输。它包括：串、并行口，键盘，文件系统，SCSI，网络等驱动程序；另外一种软驱动呢？不需要直接对硬件就行操作。我认为他可以理解为它是在硬驱动之上的一层更为高级的驱动。我想学习的主要是软驱动。

一般来说，设备驱动程序的任务主要有两个：第一，接受来自用户程序的读写请求，把用户的数据传送给设备，或把从设备接收到的数据传送给用户；第二，轮询设备或处理来自设备的中断请求，完成数据传输。

#### 驱动程序的结构

在这里，我主要介绍WDM的结构。WDM(Windows driver module)是什么东西呢？在Windows98\95下面，也许你听得最多的是VXD，我只知道VXD是一种驱动程序，和WDM差不多的东西。只是因为Windows2000是WindowsNT那条线过来的东西，要加上两个主要的新功能：即插即用（Plug and Play）和电源管理（Power Menage），又不能用Windows98\95那一套，所以就搞出一个叫WDM这么个东西，来支持PNP和PM。

其实想想, 现在的技术名词还不是一般的多啊! 总之wdm大家都叫它windows驱动程序模型。

Windows2000里有叫即插即用管理器和I/O(此I/O非彼I/O端口)管理器的两个东西。比如说我在机器上插了一张符合PCI规范的PCI卡。即插即用管理器会发现这张卡插在第N个插槽上, 然后即插即用管理器会说它找到了这样一张卡, 它就去找有没有现成的驱动程序, 如果没有找到, 它会告诉我们, 我找到了这样一张卡, 请你插入这张卡的驱动程序盘。好, 我们就把驱动程序盘给它, 即插即用管理器会去找驱动程序盘上的INF文件, 找到后它会比较PCI卡上的标志和INF文件里的标志是否相同, 如果相同, 它就会依照INF文件里提供的路径去找驱动程序, 找到之后就可以交给I/O管理器, I/O管理器会装载这个驱动程序。I/O管理器在做了一些接口的工作后, 即插即用管理器会先分配好相关的资源给PCI卡, 比如说I/O端口空间、内存空间和中断向量, 然后告诉这张卡的驱动程序, 我给你分配了这些资源, 你看怎么的。如果你没有怎么的或不敢怎么的, 那就赶快记下这些资源, 以备后用。

下面说I/O管理器这个东西。上面我们讲到I/O管理器装载这个驱动程序, 驱动程序有一个大门, 还有N多的小门。I/O管理器先从大门进去(因为I/O管理器只找得到大门, I/O管理器是不是很傻, NO, 当然有它的道理, 你别问我: I/O管理器怎么找到大门的? 驱动程序无非就是一些文件, I/O管理器把这么些文件加载到系统中去), 去找一样东西: 进小门的地图。我们要在大门进去的房间里放这张地图(驱动程序都是我们造的, 我们当然有驱动程序的地图啦)。I/O管理器找到了地图, 就可以自由进出大小门了。——这些大小门说白了就是函数(不要问我函数是什么东东), 小门的地图就是函数的地址。I/O管理器知道了这些函数的地址, 当然就可以调用这些函数啦。还有一个叫IRP的东西, 中文名叫I/O请求包。我们这样来理解它: 在用户的应用程序这一端, 要和驱动程序对话, 它们之间不是简单的调用函数(至于为什么, 我现在也不知道), 应用程序和驱动程序之间有I/O管理器隔着, 应用程序对驱动程序的操作, 首先由I/O管理器处理成一个包, 这个包里面有应用程序请求的操作内容、传送的数据等等一些东西, 然后I/O管理器把这个包扔给驱动程序, 驱动程序依照包里的请求, 完成操作, 把该回传的数据放进包里, 再把包扔还给I/O管理器, I/O管理器再把数据返回给应用程序。——这里所说的包, 就是IRP。

这里说的只是WDM结构的一部分, 但是有了这一部分知识, 其它部分就不难懂了。通过上面的介绍你看见了什么。你可以想象得出驱动程序是什么样子的吗? 我是这样想的。驱动程序好像就是一个函数库, 只不过在大门的地方放了一张地图。而这个大门与地图我们见到过吗? 好像有点像dll文件呢。在早些时候我学dll的时候, 我就只会用dll来存放函数。

posted on 2007-11-14 16:59 咖啡猪 阅读(290) 评论(0) 编辑 收藏 所属分类: C++驱动开发

[社区](#) [新闻](#) [新用户注册](#) [刷新评论列表](#)

标题

姓名

主页

Email

(只有博主才能看到)

验证码



内容(请不要发表任何与政治相关的内容)

Remember Me?

[登录](#) [使用高级评论](#) [新用户注册](#) [返回页首](#) [恢复上次提交](#)

[使用Ctrl+Enter键可以直接提交]

**相关文章:**

[Castle 开发系列文章](#)  
[asp.net控件开发基础\(1\)](#)  
[驱动程序开发流程 \(初学者适用\)](#)  
[“在系统启动时至少有一个服务或驱动程序产生错误”解决办法](#)  
[Winform下的Datagrid的列风格\(3\)—DataGridTimePickerColumn\(转\)](#)  
[ASP.NET AJAX入门系列 \(1\) : 概述](#)

**相关链接:**

[程序员的网上家园](#)  
[Java程序员如何轻松留学美国](#)  
[这里有程序员必听的音乐100首](#)  
[一个连英语都不会的.NET程序员能走多远?](#)  
[想知道博客园有多少美女程序员吗?](#)  
[移山之道-VSTS软件开发指南](#)  
[道不远人-解析ASP.NET 2.0 控件开发](#)

**所属分类的其他文章:**

[\[转\]驱动开程序发—安装\(6\)](#)  
[\[转\]驱动程序开发—编译前传\(4\)](#)  
[\[转\]驱动程序开发—编译正传\(5\)](#)  
[\[转\]驱动程序开发—Hello Word\(3\)](#)  
[\[转\]驱动程序开发—工具篇\(2\)](#)

**最新IT新闻:**

[谷歌在企业版Google Apps中整合视频共享功能](#)  
[联想ThinkPad X61存在严重散热问题](#)  
[雅虎拟于9月29日关闭旗下社交网站Mash](#)  
[2008年9月2日科技博客精选](#)  
[Myspace推出新手机版欧洲社交网站](#)  
[博客园新闻频道](#) [博客园首页](#) [社区](#)

<a href="#">ERP软件系统免费资料</a> 国内外顶尖ERP供应商及其软件信息 免费获取ERP软件对比评估报告! <b>V V</b>	<a href="#">ARM开发方案 海天雄电子</a> 我公司是一家嵌入式产品开发为主的高科技企业,提供各种开发平台和产品方案
---------------------------------------------------------------------------------	----------------------------------------------------------------------

Powered by:  
[博客园](#)  
Copyright © 咖啡猪