<<Windows驱动开发技术详解>>

图书基本信息

书名：<<Windows驱动开发技术详解>>

13位ISBN编号：9787121068461

10位ISBN编号：712106846X

出版时间：2008-1

出版时间：电子工业出版社

作者：张帆等

页数：530

<<Windows驱动开发技术详解>>

内容概要

《Windows驱动开发技术详解（珍藏版）》由浅入深、循序渐进地介绍了Windows驱动程序的开发方法与调试技巧。

《Windows驱动开发技术详解（珍藏版）》共分23章，内容涵盖了Windows操作系统的基本原理、NT驱动程序与WDM驱动程序的构造、驱动程序中的同步异步处理方法、驱动程序中即插即用功能、驱动程序的各种调试技巧等。

同时，还针对流行的PCI驱动程序、 USB驱动程序、虚拟串口驱动程序、摄像头驱动程序、SDIO驱动程序进行了详细的介绍，《Windows驱动开发技术详解（珍藏版）》最大的特色在于每一节的例子都是经过精挑细选的，具有很强的针对性。

力求让读者通过亲自动手实验，掌握各类Windows驱动程序的开发技巧，学习尽可能多的Windows底层知识。

《Windows驱动开发技术详解（珍藏版）》适用于中、高级系统程序员，同时也可用做高校计算机专业操作系统实验课的补充教材。

作者简介

张帆，毕业于北京理工大学电子工程系，曾就职于威盛电子有限公司，现就职于北京创毅视讯科技有限公司。
长期从事PCI、USB、SDIO、串口、摄像头等设备的Windows驱动程序开发。
对Windows操作系统内核有深入的研究，并且有丰富的Windows驱动程序开发经验。

史彩成：博士后，北京理工大学信息科学技术学院副教授，资深电子系统专家，主要从事图像处理、激光信号处理、数据融合及ASIC设计等领域的研究工作。

书籍目录

Memory Address） 5.1.3 用户模式地址和内核模式地址 5.1.4 Windows驱动程序和进程的关系 5.1.5 分页与非分页内存 5.1.6 分配内核内存 5.2 在驱动中使用链表 5.2.1 链表结构 5.2.2 链表初始化 5.2.3 从首部插入链表 5.2.4 从尾部插入链表 5.2.5 从链表删除 5.2.6 实验 5.3 Lookaside结构 5.3.1 频繁请内存的弊端 5.3.2 使用Lookaside 5.3.3 实验 5.4 运行时函数 5.4.1 内存间复制（非重叠） 5.4.2 存间复制（可重叠） 5.4.3 填充内存 5.4.4 内存比较 5.4.5 关于运行时函数使用的注意事项 5.4.6 实验 5.5 使用C++特性分配内存 5.6 其他 5.6.1 数据类型 5.6.2 返回状态值 5.6.3 检查内存可用性 结构化异常处理（try-except块） 5.6.5 结构化异常处理（try-finally块） 5.6.6 使用宏需要注意的地方 5.6.7 断言 5.7 小结 第6章 Windows内核函数 本章介绍了Windows内核模式下的一些常用内核函，这些函数在驱动程序的开发中将会经常用到。
6.1 内核模式下的字符串操作 6.1.1 ASCII字符串和宽字符串 6.1.2 ANSI_STRING字符串与UNICODE_STRING字符串 6.1.3 字符初始化与销毁 6.1.4 字符串复制 6.1.5 字符串比较 6.1.6 字串转化成大写 6.1.7 字符串与整型数字相互转换 6.1.8 ANSI_STRING字符串与UNICODE_STRING字符串相互转换 6.2 内核模式下的文件操作 6.2.1 文件的创建 6.2.2 文件的打开 6.2.3 获取或修改文属性 6.2.4 文件的写操作 6.2.5 文件的读操作 6.3 内核模式下的注册表操作 6.3.1 创建关闭注册表 6.3.2 打开注册表 6.3.3 添加、修改注册表键值 6.3.4 查询注册表 6.3.5 枚举子项 6.3.6 枚举子键 6.3.7 删除子项 6.3.8 其他 6.4 小结 第7章 派遣函数 本章重点介绍了驱动程序中的处理IRP请派遣函数。
所有对设备的操作最终将转化为IRP请求，这些IRP请求会被传送到派遣函数处理。
7.1 IRP与派遣函数 7.1.1 IRP 7.1.2 IRP类型 7.1.3 对派遣函数的简单处理 7.1.4 通过设备链接打开备 7.1.5 编写一个更通用的派遣函数 7.1.6 跟踪IRP的利器IRPTrace 7.2 缓冲区方式读写操作 7.2.1 冲区设备 7.2.2 缓冲区设备读写 7.2.3 缓冲区设备模拟文件读写 7.3 直接方式读写操作 7.3.1 直接取设备 7.3.2 直接读取设备的读写 7.4 其他方式读写操作 7.4.1 其他方式设备 7.4.2 其他方式读写 IO设备控制操作 7.5.1 DeviceIoControl与驱动交互 7.5.2 缓冲内存模式IOCTL 7.5.3 直接内存模式IOCTL 7.5.4 其他内存模式IOCTL 7.6 小结 第2篇 进阶篇第8章 驱动程序的同步处理 本章介绍驱动程序中常用的同步处理办法，并且将内核模式下的同步处理方法和用户模式下的同步处理方法做了比较。
另外，本章还介绍了中断请求级、自旋锁等同步处理机制。
8.1 基本概念 8.1.1 问题的引出 8.1.2 同步与异步 8.2 中断请求级 8.2.1 中断请求（IRQ）与可编程中断控制器（PIC） 8.2.2 高级可编程控制器（APIC） 8.2.3 中断请求级（IRQL） 8.2.4 线程调度与线程优先级 8.2.5 IRQL的变化 8.2.6 IRQL与内存分页 8.2.7 控制IRQL提升与降低 8.3 自旋锁 8.3.理 8.3.2 使用方法 8.4 用户模式下的同步对象 8.4.1 用户模式的等待 8.4.2 用户模式开启多线程 8 用户模式的事件 8.4.4 用户模式的信号灯 8.4.5 用户模式的互斥体 8.4.6 等待线程完成 8.5 内核模式下的同步对象 8.5.1 内核模式下的等待 8.5.2 内核模式下开启多线程 8.5.3 内核模式下的事件对象 8.5.4 驱动程序与应用程序交互事件对象 8.5.5 驱动程序与驱动程序交互事件对象 8.5.6 内核模式下的信号灯 8.5.7 内核模式下的互斥体 8.5.8 快速互斥体 8.6 其他同步方法 8.6.1 使用自旋锁进行同步 8.6.2 使用互锁操作进行同步 8.7 小结 第9章 IRP的同步 本章详细地介绍了IRP的同步处理方法和异步处理方法。
另外，本章还介绍了StartIO例程、中断服务例程、DPC服务例程。
9.1 应用程序对设备的同步异步操作 9.1.1 同步操作与异步操作原理 9.1.2 同步操作设备 9.1.3 异步操作设备（方式一） 9.1.4 异步操作设备（方式二） 9.2 IRP的同步完成与异步完成 9.2.1 IRP的同步完成 9.2.2 IRP的异步完成 9.2.3 取消IRP 9.3 StartIO例程 9.3.1 并行执行与串行执行 9.3.2 StartIO例程 9.3.3 示例 9.4 自定义的StartIO 9.4.1 多个串行化队列 9.4.2 示例 9.5 中断服务例程 9.5.1 中作的必要性 9.5.2 中断优先级 9.5.3 中断服务例程（ISR） 9.6 DPC例程 9.6.1 延迟过程调用例程（DPC） 9.6.2 DpcForISR 9.7 小结 第10章 定时器 本章总结了在内核模式下的四种等待方法，读者以利用这些方法灵活地用在自己的驱动程序中。
最后本章还介绍了如何对IRP的超时情况进行处理。
10.1 定时器实现方式一 10.1.1 I/O定时器 10.1.2 示例代码 10.2 定时器实现方式二 10.2.1 DPC定时

器 10.2.2 示例代码 10.3 等待 10.3.1 第一种方法：使用KeWaitForSingleObject 10.3.2 第二种方法：使用KeDelayExecutionThread 10.3.3 第三种方法：使用KeStallExecutionProcessor 10.3.4 第四种方法：使用定时器 10.4 时间相关的其他内核函数 10.4.1 时间相关函数 10.4.2 示例代码 10.5 IRP的超时处理 10.5.1 原理 10.5.2 示例代码 10.6 小结 第11章 驱动程序调用驱动程序 本章主要介绍了如何在程序中调用其他驱动程序。
比较简单的方法是将被调用的驱动程序以文件的方式操作。
比较高级的方法是构造各种IRP，并将这些IRP传送到被调用的驱动程序中。
11.1 以文件句柄形式调用其他驱动程序 11.1.1 准备一个标准驱动 11.1.2 获得设备句柄 11.1.3 同步调用 11.1.4 异步调用方法一 11.1.5 异步调用方法二 11.1.6 通过符号链接打开设备 11.2 通过设备指针调用其他驱动程序 11.2.1 用IoGetDeviceObjectPointer获得设备指针 11.2.2 创建IRP传递给驱动的派遣函数 11.2.3 用IoBuildSynchronousFsdRequest创建IRP 11.2.4 用IoBuildAsynchronousFsdRequest创建IRP 11.2.5 用IoAllocateIrp创建IRP 11.3 其他方法获得设备指针 11.3.1 用ObReferenceObjectByName获得设备指针 11.3.2 剖析IoGetDeviceObjectPointer 11.4 小结 第12章 分层驱动程序 本章主要介绍了层驱动的概念。
分层驱动可以将功能复杂的驱动程序分解为多个功能简单的驱动程序。
多个分层的驱动程序形成一个设备堆栈，IRP请求首先发送到设备堆栈的顶层，然后依次穿越每层的设备堆栈，最终完成IRP请求。
12.1 分层驱动程序概念 12.1.1 分层驱动程序的概念 12.1.2 设备堆栈与挂载 12.1.3 I/O堆栈 12.1.4 下转发IRP 12.1.5 挂载设备对象示例 12.1.6 转发IRP示例 12.1.7 分析 12.1.8 遍历设备栈 12.2 完成程 12.2.1 完成例程概念 12.2.2 传播Pending位 12.2.3 完成例程返回STATUS_SUCCESS 12.2.4 完成例程返回STATUS_MORE_PROCESSING_REQUIRED 12.3 将IRP分解成多个IRP 12.3.1 原理 12.3.2 准备底层驱动 12.3.3 读派遣函数 12.3.4 完成例程 12.3.5 分析 12.4 WDM驱动程序架构 12.4.1 WDM层驱动程序 12.4.2 WDM的加载方式 12.4.3 功能设备对象 12.4.4 物理设备对象 12.4.5 物理设备对象与即插即用 12.5 小结 第13章 让设备实现即插即用 本章首先介绍即插即用的概念和驱动程序支持即插即用功能的必要性。
另外，本章还介绍如何利用WDM驱动程序开发框架设计支持即插即用功能的驱动程序。
13.1 即插即用概念 13.1.1 历史原因 13.1.2 即插即用的目标 13.1.3 Windows中即插即用相关组件 13.1.4 遗留驱动程序 13.2 即插即用IRP 13.2.1 即插即用IRP的功能代码 13.2.2 处理即插即用IRP派遣函数 13.3 通过设备接口寻找设备 13.3.1 设备接口 13.3.2 WDM驱动中设置接口 13.3.3 应用程寻找接口 13.3.4 查看接口设备 13.4 启动和停止设备 13.4.1 为一个实际硬件安装HelloWDM 13.4.2 动设备 13.4.3 转发并等待 13.4.4 获得设备相关资源 13.4.5 枚举设备资源 13.4.6 停止设备 13.5 即用的状态转换 13.5.1 状态转换图 13.5.2 IRP_MN_QUERY_STOP_DEVICE 13.5.3 IRP_MN_QUERY_REMOVE_DEVICE 13.6 其他即插即用IRP 13.6.1 IRP_MN_FILTER_RESOURCE_REQUIREMENTS 13.6.2 IRP_MN_QUERY_CAPABILITIES 13.7 小结 第14章 电源管理 本章主要介绍了如何在WDM驱动程序中进行电源处理。
电源处理主要是处理好电源状态和设备状态。
14.1 WDM电源管理模型 14.1.1 概述 14.1.2 热插拔 14.1.3 电源状态 14.1.4 设备状态 14.1.5 状换 14.2 处理IRP_MJ_POWER 14.3 处理IRP_MN_QUERY_CAPABILITIES 14.3.1 DEVICE_CAPABILITIES 14.3.2 一个试验 14.4 小结 第3篇 实用篇第15章 I/O端口操作 本章总结多种I/O端口操作的方法。
这些方法本质上是一样的，都是将端口输入输出的汇编指令运行在内核模式中。
15.1 概述 15.1.1 从DOS说起 15.1.2 汇编实现 15.1.3 DDK实现 15.2 工具软件WinIO 15.2.1 Win介 15.2.2 使用方法 15.3 端口操作实现方法一 15.3.1 驱动端程序 15.3.2 应用程序端程序 15.4 端作实现方法二 15.4.1 驱动端程序 15.4.2 应用程序端程序 15.5 端口操作实现方法三 15.5.1 驱动端程序 15.5.2 应用程序端程序 15.6 端口操作实现方法四 15.6.1 原理 15.6.2 驱动端程序 15.6.3 应用端程序 15.7 驱动PC喇叭 15.7.1 可编程定时器 15.7.2 PC喇叭 15.7.3 操作代码 15.8 操作并口设备 15.8.1 并口设备简介 15.8.2 并口寄存器 15.8.3 并口设备操作 15.9 小结 第16章 PCI设备驱动

主要介绍PCI设备的驱动开发。
首先介绍了PCI总线协议。
作为驱动程序员，开发PCI驱动程序首先要了解PCI配置空间。
根据读取PCI配置空间，可以得到PCI设备的所有资源。
另外，本章还总结了四种获取PCI配置空间的方法。
16.1 PCI总线协议 16.1.1 PCI总线简介 16.1.2 PCI配置空间简介 16.2 访问PCI配置空间方法一 16.2.
两个重要寄存器 16.2.2 示例 16.3 访问PCI配置空间方法二 16.3.1 DDK函数读取配置空间 16.3.2
例 16.4 访问PCI配置空间方法三 16.4.1 通过即插即用IRP获得PCI配置空间 16.4.2 示例 16.5 访问P
配置空间方法四 16.5.1 创建IRP_MN_READ_CONFIG 16.5.2 示例 16.6 PCI设备驱动开发示例 16.6.1
开发步骤 16.6.2 中断操作 16.6.3 操作设备物理内存 16.6.4 示例 16.7 小结 第17章 USB设备驱动
首先介绍了USB总线协议的基本框架，其中包括USB总线的拓扑结构，USB通信的流程，还有USB的四
种传输模式。
另外，本章介绍了如何编写USB总线设备的驱动程序。
17.1 USB总线协议 17.1.1 USB设备简介 17.1.2 USB连接拓扑结构 17.1.3 USB通信的流程 17.1.4 US
四种传输模式 17.2 Windows下的USB驱动 17.2.1 观察USB设备的工具 17.2.2 USB设备请求 17.2.3
备描述符 17.2.4 配置描述符 17.2.5 接口描述符 17.2.6 端点描述符 17.3 USB驱动开发实例 17.3.1
能驱动与物理总线驱动 17.3.2 构造USB请求包 17.3.3 发送USB请求包 17.3.4 USB设备初始化 17.3.5
USB设备的插拔 17.3.6 USB设备的读写 17.4 小结 第18章 SDIO设备驱动 本章首先介绍了SDIO协
，讲述了SD内存卡和SDIO卡的兼容问题。
然后介绍了SDIO协议中的发送命令、回应命令、传送数据等相关协议。
随后，本章又介绍了Windows中，DDK提供的对SDIO卡设备的支持。
然后介绍了如何利用总线驱动，使SDIO设备初始化，接收中断，发送和接收数据等操作。
18.1 SDIO协议 18.1.1 SD内存卡概念 18.1.2 SDIO卡概念 18.1.3 SDIO总线 18.1.4 SDIO令牌 18.1.
SDIO令牌格式 18.1.6 SDIO的寄存器 18.1.7 CMD52命令 18.1.8 CMD53命令 18.2 SDIO卡驱动开
框架 18.2.1 SDIO Host Controller驱动 18.2.2 SDIO卡的初始化 18.2.3 中断回调函数 18.2.4 获得和设
属性 18.2.5 CMD52 18.2.6 CMD53 18.3 SDIO开发实例 18.4 小结 第19章 虚拟串口设备驱动
绍了串口开发的框架模型，在串口的AddDevice例程中需要暴露出一个串口的符号连接，另外在相应
的注册表中需要进行设置。
在串口与应用程序的通信中，主要是一组DDK定义的IO控制码，这些IO控制码负责由应用程序向驱动
发出请求。
19.1 串口简介 19.2 DDK串口开发框架 19.2.1 串口驱动的入口函数 19.2.2 应用程序与串口驱动的通
信 19.2.3 写的实现 19.2.4 读的实现 19.3 小结 第20章 摄像头设备驱动程序 本章主要介绍了微软
的摄像头驱动框架。
在该框架中，微软提供了类驱动和小驱动的概念。
对于驱动程序员的任务就是编写小驱动程序。
20.1 WDM摄像头驱动框架 20.1.1 类驱动与小驱动 20.1.2 摄像头的类驱动与小驱动 20.1.3 编写小驱
动程序 20.1.4 小驱动的流控制 20.2 虚拟摄像头开发实例 20.2.1 编译和安装 20.2.2 虚拟摄像头入口
函数 20.2.3 对STREAM_REQUEST_BLOCK的处理函数 20.2.4 打开视频流 20.2.5 对视频流的读取 20
小结 第4篇 提高篇第21章 再论IRP 本章将相关IRP的操作做了进一步的总结。
首先是转发IRP，归纳了几种不同的方式。
其次总结了创建IRP的几种不同方法。
创建IRP总的来说分为创建同步IRP和创建异步IRP。
对于创建同步IRP，操作比较简单，I/O管理器会负责回收IRP的相关内存，但是使用不够灵活。
对于创建异步IRP，操作比较复杂，程序员需要自己负责对IRP及相关内存回收，但使用十分灵活。
21.1 转发IRP 21.1.1 直接转发 21.1.2 转发并且等待 21.1.3 转发并且设置完成例程 21.1.4 暂时挂起
当前IRP 21.1.5 不转发IRP 21.2 创建IRP 21.2.1 IoBuildDeviceIoControlRequest 21.2.2 创建有超时
的IOCTL IRP 21.2.3 用IoBuildSynchronousFsdRequest创建IRP 21.2.4 关

章节摘录

第1篇　入门篇　　第1章　从两个最简单的驱动谈起　　Windows驱动程序的编写，往往需要开发人员对Windows内核有深入了解和大量的内核调试技巧，稍有不慎，就会造成系统的崩溃。
因此，初次涉及Windows驱动程序开发的程序员，即使拥有大量Win32程序的开发技巧，往往也很难入门。

本章向读者呈现两个最简单的Windows驱动程序，一个是NT式的驱动程序，另一个是WDM式的驱动程序。
这两个驱动程序没有操作具体的硬件设备，只是在系统里创建了虚拟设备。
在随后的章节中，它们会作为基本驱动程序框架，被本书其他章节的驱动程序开发所复用。
笔者将带领读者编写代码、编译、安装和调试程序。
相信对第一次编写驱动程序的读者来说，这将是非常激动和有趣的。
代码的具体讲解将分散在后面的章节论述。
现在请和笔者一起，开始Windows驱动编程之旅吧！

1．1 DDK的安装　　在编写第一个驱动之前，需要先安装微软公司提供的Windows驱动程序开发包DDK（Driver Development Kit）。
笔者计算机里安装的是Windows XP 2462版本的DDK，建议读者安装同样版本或者更高版本的DDK，如图1-1所示。

在安装的时候请选择完全安装，即安装DDK的所有部件，如图1-2所示。
因为除了DDK的基本编译环境外，DDK还提供了大量的源代码和实用工具，这对于Windows驱动程序的初学者进行学习和编写驱动程序将是非常有用的。

安装完毕后，会在开始菜单中出现相应的项目。
其中，主要用到的是BuildEnvironment，如图1-3所示。
该版本的DDK会同时安装上Windows 2000和Windows XP的编译环境。

媒体关注与评论

本书是作者结合教学和科研实践经验编写而成的，不仅详细介绍了Windows内核原理，并且介绍了编程技巧和应用实例，兼顾了在校研究生和工程技术人员的实际需求，对教学、生产和科研有现实的指导意义，是一本值得推荐的专著。

中国工程院院士　　毛二可

编辑推荐

原创经典，威盛一线工程师倾力打造。
深入驱动核心，剖析操作系统底层运行机制，通过实例引导，快速学习编译、安装、调试的方法。
从Windows最基本的两类驱动程序的编译、安装、调试入手讲解，非常容易上手，用实例详细讲解PCI、USB、虚拟串口、虚拟摄像头、SDIO等驱动程序的开发，归纳了多种调试驱动程序的高级技巧，如用WinDBG和VMWARE软件对驱动进行源码级调试，深入Windows操作系统的底层和内核，透析Windows驱动开发的本质。
本书是作者结合教学和科研实践经验编写而成的，不仅详细介绍了Windows内核原理，而且介绍了编程技巧和应用实例，兼顾了在校研究生和工程技术人员的实际需求，对教学、生产和科研有现实的指导意义，是一本值得推荐的专著。
——中国工程院院士　　院士推荐　　目前，电子系统设计广泛采用通用操作系统，达到降低系统的设计难度和缩短研发周期。
实现操作系统与硬件快速信息交换是电子系统设计的关键。
通用操作系统硬件驱动程序的开发，编写者不仅需要精通硬件设备、计算机总线，而且需要Windows操作系统知识以及调试技巧。
学习和掌握Windows硬件驱动程序的开发是电子系统设计人员必备的能力。
本书是作者结合教学和科研实践经验编写而成的，不仅详细介绍了Windows内核原理，并且介绍了编程技巧和应用实例，兼顾了在校研究生和工程技术人员的实际需求，对教学、生产和科研有现实的指导意义，是一本值得推荐的专著。

版权说明