

## <<Windows 7设备驱动程序开发>>

### 图书基本信息

书名：<<Windows 7设备驱动程序开发>>

13位ISBN编号：9787115265791

10位ISBN编号：7115265798

出版时间：2012-1

出版时间：人民邮电

作者：里夫斯

页数：220

译者：张猛,纪小玲,周姝嫣

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<Windows 7设备驱动程序开发>>

### 内容概要

设备驱动程序是非常特殊的软件，应用程序通过它才能与外界沟通。

Windows

7与外界的所有沟通都必须有设备驱动程序的参与。

本书介绍了编写Windows

7设备驱动程序所需的知识，涵盖了用户模式驱动程序开发、内核模式驱动程序开发、WDF架构、驱动程序调试等主题。

作者展示了如何利用微软提供的强大工具和模型，高效地开发稳定、健壮的驱动程序。

通过本书，你将学会：

如何使用WDF减少开发时间，提高系统稳定性，增强实用性；

如何利用UMDF和KMDF进行开发；

如何以最佳方式设计、开发、调试用户模式驱动程序和内核模式驱动程序；

如何管理I/O请求和队列、自管理I/O、同步、锁、即插即用、电源管理和设备枚举；

如何利用COM开发UMDF；

如何利用安全的默认设置、参数验证、Unicode计数字符串和安全的设备命名技巧，确保用户模式驱动程序的安全。

无论是开发实验器材、通信硬件或其他设备的Windows驱动，本书都有助于加速产品的上市。

## 作者简介

Ronald D. Reeves

工程学和计算机专业双料博士。

他具有40多年的计算机硬件、软件和驱动程序的设计和开发经验，是软件开发和咨询公司Software Genesis的创始人兼董事长。

他是美国知名的技术作家、咨询顾问和教师，是Win32 System Services ( Prentice Hall,

2001 ) 的作者之一，曾为两本介绍设备驱动程序的图书做过技术审稿人。

目前他在劳伦斯理工大学和克利里大学教授研究生课程。

书籍目录

第一部分 设备驱动程序体系结构概述

第1章 对象

- 1.1 对象的本质
- 1.2 软件对象是什么
- 1.3 预备知识
- 1.4 软件组件

第2章 WDF 体系结构

- 2.1 WDF 组件的功能
- 2.2 WDF 的设计目标
- 2.3 WDF 中的设备和驱动程序支持
- 2.4 WDF 驱动程序模型
- 2.5 WDF 对象模型
  - 2.5.1 内核模式对象
  - 2.5.2 用户模式对象
- 2.6 即插即用和电源管理支持
- 2.7 集成的I/O 排队和取消
  - 2.7.1 并发
  - 2.7.2 I/O 模型
  - 2.7.3 I/O 请求流程
  - 2.7.4 设备I/O 请求
  - 2.7.5 即插即用和电源管理请求
- 2.8 WMI 请求 ( 仅限内核模式驱动程序 )
- 2.9 驱动程序框架
  - 2.9.1 内核模式框架
  - 2.9.2 用户模式框架
- 2.10 Windows 内核
  - 2.10.1 反射器
  - 2.10.2 驱动程序宿主进程
  - 2.10.3 驱动程序管理器
- 2.11 开发和测试工具
  - 2.11.1 PFD
  - 2.11.2 SDV
  - 2.11.3 框架验证器
  - 2.11.4 跟踪日志
  - 2.11.5 调试器扩展
  - 2.11.6 适用性和版本控制

第二部分 用户模式驱动程序

第3章 Windows 7 用户模式驱动程序概述和操作

- 3.1 在用户模式中支持的设备
- 3.2 UMDF 模型概述
  - 3.2.1 UMDF 对象模型
  - 3.2.2 UMDF 对象
- 3.3 驱动程序回调接口
- 3.4 UMDF 驱动程序功能
  - 3.4.1 身份模拟

## <<Windows 7设备驱动程序开发>>

- 3.4.2 设备属性存储
- 3.5 I/O 请求流
  - 3.5.1 I/O 请求调度
  - 3.5.2 创建、清理和关闭请求
  - 3.5.3 创建、读、写和设备I/O 控制请求
- 3.6 I/O 队列
  - 3.6.1 调度类型
  - 3.6.2 队列和电源管理
- 3.7 I/O 请求对象
  - 3.7.1 根据I/O 请求获取缓冲区
  - 3.7.2 向I/O 目标发送I/O 请求
  - 3.7.3 为I/O 请求创建缓冲区
  - 3.7.4 取消和挂起的请求
  - 3.7.5 完成I/O 请求
  - 3.7.6 自适应超时
- 3.8 自我管理I/O
- 3.9 同步问题
- 3.10 锁
- 3.11 即插即用和电源管理通知
- 3.12 设备枚举和启动
- 3.13 关闭设备电源和移除设备
- 3.14 构建、测试和调试
  - 3.14.1 安装和配置
  - 3.14.2 版本控制和更新
- 第4章 针对UMDF 进行驱动程序编程
  - 4.1 Windows I/O 概述
  - 4.2 COM 简介
  - 4.3 UMDF 体系结构
  - 4.4 必需的驱动程序功能
  - 4.5 UMDF 示例驱动程序
    - 4.5.1 最小UMDF 驱动程序：Skeleton 驱动程序
    - 4.5.2 Skeleton 驱动程序的类、对象和接口
  - 4.6 驱动程序动态链接库和导出
    - 4.6.1 驱动程序入口点：DllMain
    - 4.6.2 获得类对象：DllGetClassObject
  - 4.7 COM 支持函数
    - 4.7.1 IUnknown 方法
    - 4.7.2 IClassFactory 接口
    - 4.7.3 驱动程序回调对象
    - 4.7.4 设备回调对象
  - 4.8 以Skeleton 驱动程序为基础进行开发
    - 4.8.1 自定义导出文件
    - 4.8.2 自定义源文件
    - 4.8.3 自定义INX 文件
    - 4.8.4 自定义Comsup.cpp 文件
    - 4.8.5 将特定于设备的代码加入Driver.cpp
    - 4.8.6 将特定于设备的代码加入Device.cpp

## <<Windows 7设备驱动程序开发>>

### 第5章 使用COM 开发UMDF 驱动程序

#### 5.1 入门指南

##### 5.1.1 COM 基础知识

##### 5.1.2 HRESULT

#### 5.2 使用UMDF COM 对象

##### 5.2.1 获得UMDF 对象的接口

##### 5.2.2 引用计数

#### 5.3 基本的基础设施实现

##### 5.3.1 DIIMain

##### 5.3.2 DIIGetClassObject

##### 5.3.3 驱动程序对象的类工厂

##### 5.3.4 实现UMDF 回调对象

##### 5.3.5 实现QueryInterface

### 第三部分 内核模式驱动程序

### 第6章 Windows 7 内核模式驱动程序概述与操作

#### 6.1 KMDF 支持的设备

#### 6.2 KMDF 组件

#### 6.3 KMDF 驱动程序结构

#### 6.4 KMDF 和WDM 驱动程序比较

#### 6.5 设备对象和驱动程序角色

##### 6.5.1 过滤器驱动程序和过滤器设备对象

##### 6.5.2 功能驱动程序和功能设备对象

##### 6.5.3 总线驱动程序和物理设备对象

##### 6.5.4 旧版设备驱动程序和控制设备对象

#### 6.6 KMDF 对象模型

##### 6.6.1 方法、属性和事件

##### 6.6.2 对象层次结构

##### 6.6.3 对象特性

##### 6.6.4 对象上下文

##### 6.6.5 对象创建和删除

#### 6.7 KMDF I/O 模型

##### 6.7.1 I/O 请求处理程序

##### 6.7.2 I/O 队列

##### 6.7.3 I/O 请求对象

##### 6.7.4 从I/O 请求检索缓冲区

##### 6.7.5 I/O 目标

##### 6.7.6 创建I/O 请求缓冲区

##### 6.7.7 取消和挂起请求

##### 6.7.8 完成I/O 请求

##### 6.7.9 自托管I/O

##### 6.7.10 访问IRP 和WDM 结构

### 第7章 即插即用和电源管理

#### 7.1 即插即用和电源管理概述

#### 7.2 设备枚举和启动

##### 7.2.1 功能或过滤器设备对象的启动顺序

##### 7.2.2 物理设备对象的启动顺序

##### 7.2.3 设备电源关闭和移除

## <<Windows 7设备驱动程序开发>>

- 7.3 WMI 请求处理程序
- 7.4 同步问题
  - 7.4.1 同步范围
  - 7.4.2 执行级别
  - 7.4.3 锁
  - 7.4.4 同步机制的交互作用
- 7.5 安全
  - 7.5.1 安全默认值
  - 7.5.2 参数验证
  - 7.5.3 Unicode 计数字符串
  - 7.5.4 安全设备命名技术
- 第8章 内核模式的安装和生成
  - 8.1 WDK 生成工具
  - 8.2 生成环境
  - 8.3 生成项目
  - 8.4 生成Featured Toaster
    - 8.4.1 Makefile 和Makefile.inc
    - 8.4.2 源文件
    - 8.4.3 生成过程
  - 8.5 安装KMDF 驱动程序
    - 8.5.1 WDF 辅助安装程序
    - 8.5.2 INF
    - 8.5.3 KMDF 驱动程序的INF
    - 8.5.4 wdffeatured.inf
  - 8.6 目录文件和数字签名
  - 8.7 安装Featured Toaster
  - 8.8 测试KMDF 驱动程序
    - 8.8.1 PREfast
    - 8.8.2 SDV
    - 8.8.3 KMDF 日志
    - 8.8.4 KMDF 验证程序
    - 8.8.5 调试KMDF 驱动程序
    - 8.8.6 内核调试
    - 8.8.7 KMDF 驱动程序功能
  - 8.9 调试宏和例程
  - 8.10 WDF 调试程序扩展命令
  - 8.11 使用WPP 跟踪与KMDF 驱动程序
  - 8.12 使用WinDbg 与Featured Toaster
  - 8.13 版本控制和动态绑定
- 第9章 为KMDF 编写驱动程序
  - 9.1 KMDF 与WDM 示例之间的差别
  - 9.2 KMDF 示例中使用的宏
  - 9.3 KMDF 驱动程序结构和概念
    - 9.3.1 对象创建
    - 9.3.2 对象上下文区
    - 9.3.3 I/O 队列
    - 9.3.4 I/O 请求

## <<Windows 7设备驱动程序开发>>

9.4 最小的KMDF 驱动程序：SimpleToaster

9.4.1 创建WDF 驱动程序对象：DriverEntry

9.4.2 创建设备对象、设备接口和I/O 队列：EvtDriver-DeviceAdd

9.4.3 设备对象和设备上下文区

9.4.4 设备接口

9.4.5 默认I/O 队列

9.4.6 处理I/O 请求：EvtIoRead、EvtIoWrite、EvtIoDevice-Control

9.5 纯软件驱动程序示例

9.5.1 文件创建和关闭请求

9.5.2 其他的设备对象特性

9.5.3 设置其他设备对象属性

第10章 为即插即用和电源管理编写程序

10.1 注册回调函数

10.2 管理电源策略

10.3 通电和断电时的回调函数

10.4 支持唤醒信号的回调函数

第11章 为WMI 支持编写程序

11.1 WMI 体系结构

11.2 注册为WMI 数据提供程序

11.3 处理WMI 请求

11.4 WDM 驱动程序的WMI 要求

11.5 WMI 类名和基类

11.6 触发WMI 事件

11.7 解决具体的WMI 问题

11.7.1 驱动程序的WMI 类并不出现在\root\wmi 命名空间中

11.7.2 不能访问驱动程序的WMI属性或方法

11.7.3 未接收驱动程序的WMI事件

11.7.4 改变WMI 请求的安全设置并不生效

11.8 测试WMI 驱动程序支持的技术

11.8.1 WMI IRP 和系统事件日志

11.8.2 WMI WDM 提供程序日志

11.9 WMI 事件跟踪

第12章 编写KMDF 硬件驱动程序

12.1 支持设备中断

12.1.1 创建中断对象

12.1.2 创建中断对象的代码

12.1.3 启用和禁用中断

12.1.4 启用中断的代码

12.1.5 禁用中断的代码

12.1.6 启用中断后和禁用中断前的处理

12.2 处理中断

12.2.1 EvtInterruptIsr 回调函数的代码

12.2.2 中断的延迟处理

12.3 映射资源

12.3.1 映射资源的代码

12.3.2 取消映射资源的代码



## <<Windows 7设备驱动程序开发>>

### 第13章 编写多个I/O队列程序并编写I/O程序

#### 13.1 编写I/O队列简介

#### 13.2 创建和配置队列

##### 13.2.1 为写请求创建队列的代码

##### 13.2.2 为读请求创建队列的代码

##### 13.2.3 为设备I/O控制请求创建队列的代码

#### 13.3 处理并行队列的请求

##### 13.3.1 处理I/O请求的代码

##### 13.3.2 执行缓冲I/O

#### 13.4 将请求转发给队列

#### 13.5 从手动队列获取请求

#### 13.6 读取和写入注册表

#### 13.7 监视器计时器：自管理I/O

##### 13.7.1 启动和重启自管理I/O设备

##### 13.7.2 设备断电和移除期间的自管理I/O

##### 13.7.3 实现监视器计时器

附录 驱动程序信息网站

参考文献

## 章节摘录

版权页：插图：WDF的主要设计目标包括：简化即插即用和电源管理的驱动程序支持，在内核模式和用户模式下都能使用即插即用和电源管理。

无缝地处理即插即用事件和电源事件，对于系统的可靠性和良好用户体验来说至关重要，但要想正确实现也出奇地复杂。

这种复杂性多数是因为驱动程序必须确定处理每个即插即用或电源管理请求的正确方式。

正确的处理取决于驱动程序的位置、设备栈、设备当前的状态、操作系统当前的状态，有时还取决于设备或系统即将出现的状态变化的特性。

这类支持通常需要数千行代码来处理复杂的、依赖于状态的各种情况。

大多数驱动程序都需要编写代码来处理那些它们根本不支持的请求。

WDF将状态跟踪和决策制定逻辑集中在框架内，而不再要求在每个驱动程序中完成这些逻辑。

WDF对即插即用和电源管理的支持基于以下原则。

编辑推荐

Windows 7设备驱动程序开发必备 融汇作者多年开发经验 深入底层探究研发之道

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>