

[转]驱动程序开发—Hello Word(3)

看了好多天的书！特别到书店买了《Windows 200/XP WDM 设备驱动开发》这本书，在这里我不想怎么评论它！对于高手来说，我觉得她一定不能满足，但是对于像我这样想入门的人来说，仿佛看了半天，还是不知道从何下手。什么原理、模型、分层等等讲不讲，讲！绝对应该讲！但是你得快点告诉我怎么先弄一个像“Hello Word!”的什么简单来不能再简单的完整的例子给我呀！到网上找阿找啊！那些高手啊！也不为我们新手写点图文并茂的上手资料。没办法！结合自己的需要再参考一些别人的东东，算是自己的一点不成熟的想法吧！

我觉得下面这个介绍非常不错！我能看懂，所以贴了出来。

我道为什么找不到“Hello Word!”呢？原来在驱动开发的例子里是没有所谓的“Hello World”程序的。这主要还是因为网络上的WDM资料太少造成的。但是程序的入口点呢？C语言有Main()，用VC的常看见的是WinMain()，Delphi开发的是Program里的Begin，但是驱动开发呢？那也是应该有程序的入口点啊。后来我才明白了，那就是DriverEntry()函数。还有一个问题让我怀疑了老半天，那就是驱动开发的源程序中需不需要include头文件呀？为什么会怀疑呢？那是因为我看了半天的书都没有看到一个完整的驱动程序结构。真的是郁闷。下面是我看到的一个完整的结构，我先放上来，让大家看看驱动开发的结构吧。

```
/*
程序名称: Hello World for WDM
文件名称: HelloWDM.cpp
日期: 2002-8-16
*/

//一定要的头文件，声明了函数模块和变量:
#include "HelloWDM.h"

/*
函数名称: DriverEntry()
功能描述: WDM程序入口（原来的WinMain被换成了DriverEntry，也是驱动程序的大门）
*/

//extern "C"是必须的，表示“用C链接”。如果你的文件名是HelloWDM.c的话，这句可以省略。
extern "C"
NTSTATUS DriverEntry( IN PDRIVER_OBJECT DriverObject, //IN 是一个关键字表示这是一个输入参数，
PDRIVER_OBJECT是一个数据结构的指针，就像PCHAR一样，这个数据结构是什么样子的，后面我会列出来。
她描述了一个驱动设备对象。
IN PUNICODE_STRING RegistryPath)//参数RegistryPath指定了驱动程序注册表键的路径，因为驱动程序安装后总会在系统注册表里留下一点东西的。
{
//指定“添加设备”消息由函数“HelloWDMAddDevice()”来处理:
DriverObject->DriverExtension->AddDevice = HelloWDMAddDevice;
//指定“即插即用”消息由函数“HelloWDMPnp()”来处理:
DriverObject->MajorFunction[IRP_MJ_PNP] = HelloWDMPnp;

//返回一个NTSTATUS值STATUS_SUCCESS。几乎所有的驱动程序例程都必须返回一个NTSTATUS值，这些值在NTSTATUS.H DDK头文件中有详细的定义。
return STATUS_SUCCESS;
}
```

导航

- 博客园
首页
新随笔
联系
订阅 XML
管理

统计

- 随笔 - 15
文章 - 1
评论 - 5
引用 - 0

与我联系

发短消息

搜索

找我看

常用链接

- 我的随笔
我的空间
我的短信
我的评论
更多链接

留言簿(2)

- 给我留言
查看留言

我参与的团队

上海.NET俱乐部(2/2222)

随笔分类

- C#(7) (rss)
C++ 驱动开发(6) (rss)
Net(1) (rss)

随笔档案

- 2008年8月 (1)
2008年7月 (2)
2008年6月 (4)
2007年11月 (6)
2007年8月 (1)
2007年7月 (1)

最新评论 XML

阅读排行榜

- 1. [转]驱动程序开发—Hello Word(3)(299)
2. [转]驱动程序开发—概述(1)(289)
3. [转]驱动程序开发—工具篇(2)(286)
4. [转]驱动开程序发—安装(6)(191)
5. [转]驱动程序开发—编译前传(4)(184)

评论排行榜

}

//NTSTATUS也是一个数据类型，上面我所说的消息有点不准确的，准确地说是“I/O请求包”，不过如果像我们以前理解消息那样来理解也无不可，我觉得两者太想了。无非就是上层的应用程序通过它来告诉驱动程序，你要给我什么服务吧！IRP_MJ_PNP就是即插即用处理的请求。你没发觉上面其实是在制造进入各个房间的“小门”

/*****

函数名称: HelloWDMAddDevice()

功能描述: 处理“添加设备”消息

*****/

```
NTSTATUS HelloWDMAddDevice(IN PDRIVER_OBJECT DriverObject,
                          IN PDEVICE_OBJECT PhysicalDeviceObject)
```

{

//定义一个NTSTATUS类型的返回值:

NTSTATUS status;

//定义一个功能设备对象 (Functional Device Object):

PDEVICE_OBJECT fdo;

//创建我们的功能设备对象，并储存到fdo中:

status = IoCreateDevice(

DriverObject, //驱动程序对象

sizeof(DEVICE_EXTENSION), //要求的设备扩展的大小

NULL, //设备名称，这里为NULL

FILE_DEVICE_UNKNOWN, //设备的类型，在标准头文件WDM.H或NTDDK.H中列出的

FILE_DEVICE_xxx值之一

0, //各种常量用OR组合在一起，指示可删除介质、只读等。

FALSE, //如果一次只有一个线程可以访问该设备，为TRUE，否则为FALSE

&fdo); //返回的设备对象

//NT_SUCCESS宏用于测试IoCreateDevice内核是否成功完成。不要忘记检查对内核的所有调用是否成功。

NT_ERROR宏不等于!NT_SUCCESS，最好使用!NT_SUCCESS，因为除了错误外，它还截获警告信息。

if (!NT_SUCCESS(status))

return status;

//创建一个设备扩展对象dx，用于存储指向fdo的指针:

PDEVICE_EXTENSION dx = (PDEVICE_EXTENSION)fdo->DeviceExtension;

dx->fdo = fdo;

//用IoAttachDeviceToDeviceStack函数把HelloWDM设备挂接到设备栈:

dx->NextStackDevice = IoAttachDeviceToDeviceStack(fdo, PhysicalDeviceObject);

//设置fdo的flags。有两个“位”是必须改变的，一个是必须清除DO_DEVICE_INITIALIZING标志，如果在DriverEntry例程中调用IoCreateDevice()，就不需要清除这个标志位。还有一个是必须设置DO_BUFFER_IO标志位:

fdo->Flags |= DO_BUFFERED_IO | DO_POWER_PAGABLE;

fdo->Flags &= ~DO_DEVICE_INITIALIZING;

//返回值:

return STATUS_SUCCESS;

1. [转]驱动开程序发—安装(6)(2)
2. 一步一步学Remoting之一:从简单开始 (2)
3. 在博客园安家了(1)
4. 论富客户端程序的webservice实现 (0)
5. [转]驱动程序开发—概述(1)(0)

http://www.cnblogs.com/4kapple/archive/2007/11/14/959418.html

2008-9-2

```

}

/*****
函数名称: HelloWDMPnp()
功能描述: 处理“即插即用”消息
*****/
NTSTATUS HelloWDMPnp(IN PDEVICE_OBJECT fdo,
                   IN PIRP Irp)
{
    //创建一个设备扩展对象dx, 用于存储指向fdo的指针:
    PDEVICE_EXTENSION dx=(PDEVICE_EXTENSION)fdo->DeviceExtension;

    //首先要通过函数IoGetCurrentIrpStackLocation()得到当前的IRP, 并由此得到Minor Function:
    PIO_STACK_LOCATION IrpStack = IoGetCurrentIrpStackLocation(Irp);
    ULONG MinorFunction = IrpStack->MinorFunction;

    //然后把这个Minor Function传递给下一个设备栈:
    IoSkipCurrentIrpStackLocation(Irp);
    NTSTATUS status = IoCallDriver(dx->NextStackDevice, Irp);

    //处理“即插即用”次功能代码:
    //当Minor Function等于IRP_MN_REMOVE_DEVICE时, 说明有设备被拔出或卸下, 这时要取消资源分配并删除设备:
    if (MinorFunction==IRP_MN_REMOVE_DEVICE)
    {
        //取消设备接口:
        IoSetDeviceInterfaceState(&dx->ifSymLinkName, FALSE);
        RtlFreeUnicodeString(&dx->ifSymLinkName);

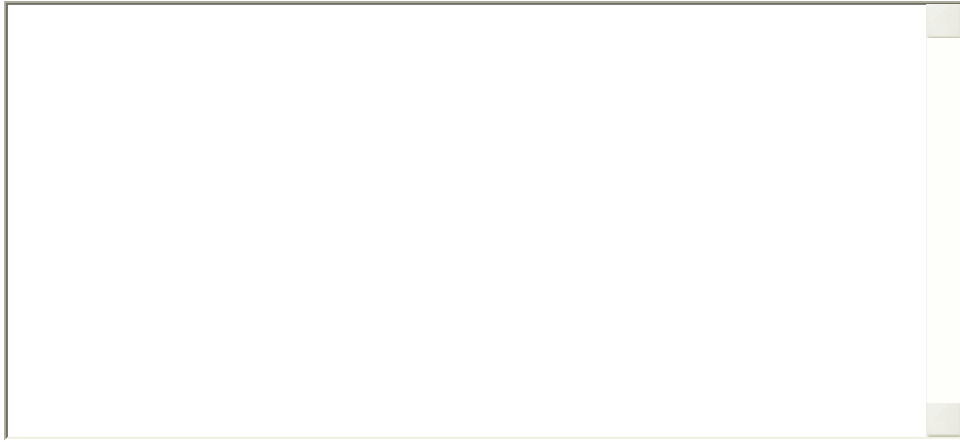
        //调用IoDetachDevice()把fdo从设备栈中脱开:
        if (dx->NextStackDevice)
            IoDetachDevice(dx->NextStackDevice);

        //删除fdo:
        IoDeleteDevice(fdo);
    }

    //返回值:
    return status;
}

/*****
程序名称: Hello World for WDM
文件名称: HelloWDM.h
作者: 罗聪
日期: 2002-8-16
*****/

```

Remember Me?

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

[使用Ctrl+Enter键可以直接提交]
该文被作者在 2007-11-14 17:05 编辑过

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\(转\)](#)
- [转: WDM驱动程序设计](#)

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?](#)
- [移山之道-VSTS软件开发指南](#)
- [程序员留学海外必看](#)
- [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—工具篇\(2\)](#)
- [\[转\]驱动程序开发—概述\(1\)](#)

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](#)

Dynamic C 开发软件 Dynamic C是一个集成式开发环境, 包括 C 语言编译器、编辑器、装载器和调试器	意法半导体STM32产品系列 Cortex-M3内核,超低功耗,高集成度 灵活的FSMC.丰富的外设.来ST官网!
---	--

Powered by:
[博客园](#)
Copyright © 咖啡猪