

```

DriverObject->DriverStartIo = StartIo;

DriverObject->MajorFunction[IRP_MJ_PNP] = DispatchPnp;

DriverObject->MajorFunction[IRP_MJ_POWER] = DispatchPower;

DriverObject->MajorFunction[IRP_MJ_SYSTEM_CONTROL] = DispatchWmi;

...

servkey.Buffer = (PWSTR) ExAllocatePool(PagedPool, RegistryPath->Length +
sizeof(WCHAR));

if (!servkey.Buffer)

    return STATUS_INSUFFICIENT_RESOURCES;

servkey.MaximumLength = RegistryPath->Length + sizeof(WCHAR);

RtlCopyUnicodeString(&servkey, RegistryPath);

return STATUS_SUCCESS;
}

```

前三条语句为驱动程序的其它入口点设置了函数指针。在这里，我用了能表达其功能的名字命名了这些函数：DriverUnload、AddDevice、StartIo。每个 WDM 驱动程序必须能处理 PNP、POWER、SYSTEM\_CONTROL 这三种请求；应该在这里为这些请求指定派遣函数。在早期的 Windows 2000 DDK 中。IRP\_MJ\_SYSTEM\_CONTROL 曾被称作 IRP\_MJ\_WMI，所以我把系统控制派遣函数命名为 DispatchWmi。在省略号处，你可以插入设置其它 MajorFunction 指针的代码。

如果驱动程序需要访问设备的服务键，可以在这里备份 RegistryPath 串。例如，如果驱动程序要作为 WMI 生产者，则需要备份这个串。这里我假设已经在某处声明了一个类型为 UNICODE\_STRING 的全局变量 servkey。返回 STATUS\_SUCCESS 指出函数成功。如果函数失败，应该返回 NTSTATUS.H 中的一个错误代码，或者返回用户定义的错误代码。STATUS\_SUCCESS 的值为 0。

#### 4.2.4 USB 驱动程序的初始化

I/O 管理器提供了一个服务函数：IoRegisterDriverReinitialization。它可以为非 WDM 驱动程序解决一个奇特的问题。非 WDM 驱动程序需要在 DriverEntry 例程中枚举它的硬件，并在其硬件的所有可能实例被识别前装入内存并初始化。

例如，鼠标和键盘设备就是这样。假定 DriverEntry 枚举了所有鼠标或键盘硬件并为它们创建了设备对象，但如果 DriverEntry 例程运行的太快，那么这些驱动程序将不能正常工作。因此，它们必须使用 IoRegisterDriverReinitialization 函数寄存一个例程，之后 I/O 管理器在某个驱动程序检测到新硬件存在时再回调这个寄存例程。最后“再初始化例程”运行，同时也把自身寄存为下一次回调的函数。

WDM 驱动程序不需要寄存再初始化例程，因为它们不需要用自己的代码去检测硬件。PnP 管理器自动把新硬件匹配到正确的 WDM 驱动程序上，并调用该驱动程序的 AddDevice 例程，再由 AddDevice 例程做所有必要的初始化工作。

#### 4.2.5 USB 驱动程序的卸载

在 WDM 驱动程序中，DriverUnload 例程的作用就是释放 DriverEntry 例程在全局初始化过程中申请的任何资源，但它几乎没什么可做。如果你在 DriverEntry 中备份了 RegistryPath 串，应该在这里释放备份所占用的内存：

```
VOID DriverUnload(PDRIVER_OBJECT DriverObject)
{
    RtlFreeUnicodeString(&servkey);
}
```

如果 DriverEntry 例程返回一个失败状态代码，系统将不再调用 DriverUnload 例程。所以，不能让 DriverEntry 例程出错后产生任何副作用，必须在它返回错误代码前消除副作用。

### 4.3 INF 文件

INF 是 Device Information File 的英文缩写，是 Microsoft 公司为硬件设备制造商发布其驱动程序推出的一种文件格式，INF 文件中包含硬件设备的信息或脚本以控制硬件操作。在 INF 文件中指明了硬件驱动该如何安装到系统中，源文件在哪里、安装到哪一个文件夹中、怎样在注册表中加入自身相关信息等等。安装监视器、调制解调器和打印机等设备所需的驱动程序，都是通过 INF 文件，正是 INF 的功劳才使得 Windows 可以找到这些硬件设备的驱动并正确安装。当我们通过“开始→控制面板→添加删除程序→Windows 安装程序”来添加系统组件的时候，INF 文件将会自动调用。而在其他场合下，则需要在 INF 文件上点击鼠标右键，然后选择“安装”，才能顺利安装应用程序。在 C: /Windows/INF 文