剖析Windows系统服务调用机制(上) PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/140/2021_2022__E5_89_96_E6_9E_90Wind_c100_140139.htm 剖析Windows系统服务调用机制 Author: Brief E-Mail: Brief#fz5fz.org Homepage: http://www.fz5fz.org/ amp. http://www.safechina.net/ Date: 07-18-2003 一> 序言 Windows系统服务调用是存在于Windows系统中的一个关键接口，常常称作System Call ，Sysem Service Call 或 System Service Dispatching等，在此我们就权且称之为Windows系统服务调用，它提供了操作系统环境由用户态切换到内核态的功能。虽然在国外关于Windows系统服务调用的讨论比较多，但却很少看到比较详细的中文资料，希望本文能够为和作者一样对Windows底层感兴趣并且是刚刚接触的朋友提供一些帮助。文章中将以一个内核级的进程监视/隐藏工具T-ProcMon为例来详细讨论Windows系统服务调用的相关技术细节。另需注意本文讨论的技术仅适用于基于Windows NT内核的操作系统，并以Windows 2000为例。 二> Windows 2000系统体系结构 微软Windows 2000是一个主要面向网络服务器的操作系统，因此它和以前大家比较熟悉的Windows 9x有很大的区别。但是对于讨论一个因商业策略而出现的个人桌面操作系统的确没有太大的价值。所以我们将主要介绍一些关于NT系统内部结构的细节。Windows 2000在实现其自身目标的过程中，我们有必要讲解一些它的特性。 1. 可扩展性(Extensibility) Windows 2000操作系统是一个面向未来的系统，所以它非常注重自身的扩展性，因为在将来可能有许多市场等方面的原因导致我们必须添加或删除目前

操作系统的一些组件，这就必须要求操作系统有较强的可扩展性。为了满足扩充/删除的各种需求，Windows 2000提供了一个重要的设计思想就是子系统(Subsystem)。我们可以将一些需要扩展的操作系统功能作为一个子系统添加到Windows 2000内，就像OS/2，POSIX等一样。当然还有一个特性就是，我们可以通过为系统服务调用添加钩子来修改系统的各项行为，这就为我们提供了一个了解系统内部并扩展系统功能的机会。 2. 可靠性和健壮性(Reliability and Robust) 一个系统存在的最基本的要求就是它的稳定性，没有稳定的环境就做不出任何满意的产品。为了满足这项要求，Windows 2000提出了基于对象的访问控制权限的措施。现代的大多数微处理器都支持两种模式：用户模式(User/Normal)和内核模式(Kernel/Privileged)。操作系统组件和关键的系统组件处于内核模式，而一般用户模式的程序只能访问私有地址空间和执行非特权等级的指令。如果用户要调用一些内核组件的功能，就得通过系统服务调用来实现。 3. 兼容性(Compatibility) Intel和Microsoft能够做到今天的一个很重要的因素就是他们支持对过去存在系统的兼容。这一点非常的关键，没有人愿意三天两头的更换系统，当然也很少有人有这个经济实力。Windows 2000为了实现对其他系统的兼容，如Dos，16位Windows等，出现了环境子系统。而在Windows 2000中必须存在的环境子系统是Win32，它是其他子系统的基础，其他子系统都是一些表面的接口，而实际上是调用了Win32提供的接口，而Win32最终也是通过系统服务调用来与内核联系的。虽然操作系统为各种环境子系统提供了不同的动态链接库，而且其中的API函数名称往往也是不同的，不过这个函数

的最终都是通过相同的系统服务调用进入内核来实现的。 4. 易维护性(Maintainability) 作为一个大型的项目，Windows 2000的维护也成为了一个大型的工程。而如此巨大的项目没有很好的维护性是无法发展下去的。为此，Windows 2000使用了分层的思想，这也是一种操作系统体系结构模型。其中，系统服务调用将系统的内核模式代码和用户模式代码隔离开来，子系统使用系统服务调用为用户提供应用程序编程接口(API)，而系统服务调用向下调用执行体实现各项功能。 就像在上文我们提到的操作系统存在的两种模式，这是建立在处理器的基础之上的。按理说，一般处理器可以提供从Ring0到Ring3的四种处理器模式，但是它们必须提供至少两种，那就是Ring0和Ring3。而一些特殊处理器指令只能在内核模式执行，而一些地址空间必须在内核模式才可以被访问。Windows 2000就利用了这个特点，将操作系统和其他关键组件保护起来，只有在内核模式才可以访问执行，而一般的用户程序就只能在用户态执行咯，这样就可以避免一些用户程序对操作系统代码的破坏，也就是大家看到的Windows 2000明显比Windows 9x稳定得多的主要原因。下面我们给出了Windows 2000的体系结构简图： 系统支持进程，服务进程，应用程序，环境子系统 应用程序编程接口 基于NTDLL.dll的本地系统服务 (用户模式) ---------- 系统服务调用 (内核模式) 执行体 系统内核，设备驱动程序 硬件抽象层 三> Windows 2000本机系统服务(Native API) Windows 2000本机系统服务又称为Windows本机应用程序编程接口，它是由执行体(Executive)为用户模式和内核模式的程序提供的系统服务集

。它包含两种类型的函数：Windows 执行系统服务的系统服务调度占位程序；子系统，子系统DLL和其他本机映像使用的内部支持函数。 从用户模式调用本机系统服务是通过NTDLL.dll来实现的。表面上，Win32函数为编程人员提供了很多接口来实现我们想要的功能，但是这些Win32函数只不过是本机应用程序编程接口的一个包装器而已，它们将本机API包装起来，调用本机系统服务来实现用户期望的功能。也就是说NTDLL.dll只是系统服务调用接口在用户模式下的一个外壳。关于用户模式下的Windows本机系统服务的相关信息，请参见我以前写的一篇文章《探测Windows2K/XP/2003本机系统信息》。 我们再谈谈从内核模式调用系统服务吧，这时就不是由NTDLL.dll导出系统服务调用的函数接口了，而是由ntoskrnl.exe来实现的，它会提供两种形式的函数：ZwXxx和NtXxx，在此我们就不多说了。大家应该注意到了，在上面我们介绍的Windows 2000系统体系结构中的系统服务调用，执行体和内核都是存在于ntoskrnl.exe(在多处理器中为ntkrnlmp.exe)之中，并且是分层的。 