[**服务程序原理和实现**](https://www.cnblogs.com/Sheenagh/p/12219466.html)

**1、什么是服务**

　　在运行框中输入services.msc，服务窗口中显示的都是服务。  
　　服务:是一种应用程序类型，它在后台运行。  
　　系统有2种服务：一种叫win32服务,他运行在用户态,对应的映像文件是.EXE或.DLL；另外一种叫系统服务,它运行在内核态,对应的映像文件是.SYS也就是驱动程序。除了运行态不同外，就是在注册表中除了在HKEY\_LOCAL\_MACHINE/SYSTE/CurrentControlSet/Services下都有一个服务名外,系统服务还多了一个设备硬健HKEY\_LOCAL\_MACHINESystem/CurrentControlSet/Enum子键。因为是驱动程序，在删除一些内核木马时.这个建默认是无法删除的,因为需要SYSTEM权限。不过右键->权限->添加->高级就可以搞定。  
　　若要在HKEY\_LOCAL\_MACHINE/SYSTE/CurrentControlSet/Services下存在键值需要代码写入注册表。

**2、服务的基础知识**

　　服务应用程序:就是接下来我们要实现的程序,他是一个EXE文件..也可以是.DLL,这里我们是sysnap.exe  
　　服务控制程序:控制服务应用程序的功能块，也是服务应用程序同服务管理器（SCM）之间的桥梁  
　　服务控制管理器:负责加载和初始化AUTO\_ATRT的服务程序,SCM维护着注册表中的服务数据库，位于：HKEY\_LOCAL\_MACHINE/SYSTEM/CurrentControlSet/Services。其下的子键就是安装的服务和驱动服务。每个子键的名称就是服务名，当安装的时候由服务安全程序的CreateService 函数指定

**3、一个服务的创建流程和基本组成**

　　A：编写我们的main()函数,该函数必须在30秒内调用StartServiceCtrlDispatcher 函数，这样exe文件就在SCM中注册了  
　　B：编写ServiceMain()，ServiceMain()要立即调用RegisterServiceCtrlHandler 注册服务控制处理函数,然后用RegisterServiceCtrlHandler返回的句柄向SCM发送状态信息,接着开始完成实际的服务任务和工作线程,一旦线程开始,ServiceMain()就等待一个事件的发生,直到服务停止,ServiceMain()才返回  
　　C：编写控制处理器ServiceCtrlHandler,接受来自SCM的请求并作出反应,其实请求一般是下面几个值  
　　　　停止服务：SERVICE\_CONTROL\_STOP  
　　　　暂停服务：SERVICE\_CONTROL\_PAUSE  
　　　　恢复被暂停的服务：SERVICE\_CONTROL\_CONTINUE  
　　　　返回服务的更新状态信息：SERVICE\_CONTROL\_INTERROGATE  
　　D：编写服务要实现的主要功能函数

**4、一个完整的服务**

包括以下：  
　　main():他告诉SCM 关于ServiceMain()的一些信息  
　　ServiceMain():开始ServiceThread(),告诉SCM关于控制处理器的一些信息  
　　ServiceCtrlHandler:接受来自SCM的请求并做出响应  
　　InitThread():由ServiceMain()打开,执行我们的任务,,就是建立一个线程来运行我们的任务

**4.1 main()**

　　SCM是一个管理系统所有服务的进程,当SCM启动某个服务时,它等待某个进程的主线程来调用StartServiceCtrlDispatcher(),这样把调用进程的住线程转换为控制分配器,控制分配器启动一个新线程,新线程运行分配表里每个服务的ServiceMain()。

**4.2 ServiceMain()**

　　是服务的入口点.它运行在一个单独的线程中,主要是为服务注册控制处理器,它指示控制分配器调用ServiceCtrlHandler()来处理SCM的请求,注册完成后将返回一个句柄  
　　通过调用SetServiceStatus,用这个句柄和SERVICE\_STATUS向SCM报告服务状态,因为这样的动作经常发生,因此可以把这个过程写成一个函数ReportStatusToSCMgr()  
　　　　RegisterServiceCtrlHandler(strServiceName, (LPHANDLER\_FUNCTION)ServiceCtrlHandler);  
接着调用ReportStatusToSCMgr()向SCM报告服务状态 ReportStatusToSCMgr();  
　　创建一个事件,在函数的最后将调用该事件来保持函数的运行知道SCM发出停止请求才返回 CreateEvent();  
　　创建一个线程来运行我们的服务函数 sysnap();  
　　最后ServiceThread()完成后返回ServiceMain(),ServiceMain()调用 WaitForSingleObject()

**4.3 ServiceCtrlHandler()**

　　检查SCM发送了什么请求并且做出反应，当用户关闭系统，所有的控制处理要调用SetServiceStatus设置SERVICE\_ACCEPT\_SHUTDOWN控制码去接收SERVICE\_CONTROL\_SHUTDOWN控制码,如果服务需要时间去清除，它可以发送 STOP\_PENDING状态消息，连同一个等待时间，这样，服务控制器在报告系统服务关闭之前才知道应该待多长时间，无论如何，都有一个服务控制器需要等待的时间，防止服务停留在shutdown状态。要改变这个时间限制，可以修改HKEY\_LOCAL\_MACHINE/SYSTEM/CurrentControlSet/Control中的WaitToKillServiceTimeout值。

**5、原理**

　　Service（服务程序）由SCM（Service Control Manager）管理，运行Service时，需要由sc（控制器）执行启动命令。sc向SCM提出服务控制请求，SCM向Service传递控制命令，并接收其返回的值。PS：sc无法直接向Service下达命令，必须通过SCM传达。  
SCM可以理解为是一个抽象化的服务接口，一般通过函数OpenSCManager调用。所有Service都是由sc调用StartService() API启动的，若Service为自启动服务，则由SCM调用StartService()启动。

**5.1 Service启动过程**

[1]sc调用StartService()  
　　sc调用StartService()的同时，SCM会创建相应的Service进程，然后执行Service进程的EP代码。

[2]Service进程调用StartServiceCtrlDispatcher()  
　　为了以服务形式运行，必须在服务进程内部调用StartServiceCtrlDispatcher() API来注册服务主函数SvcMain()的地址d调用StartServiceCtrlDispatcher()时，返回sc的StartService()函数。SCM调用Service进程的服务主函数SvcMain()。

[3]服务进程调用SetServiceStatus()  
　　虽然已经创建了Service进程，但尚未以服务的形式运行。当前状态仍为SERVICE\_START\_PENDING。在服务主函数SvcMain()内部调用SetServiceStatus(SERVICE\_RUNNING) API后，才正式以Service进程形式运行。

　　对于EXE文件形态的Windows服务程序而言，必须在其EP代码内部调用StartServiceCtrlDispatcher() API，将服务主函数（SvcMain()）的地址通知给SCM。对于DLL文件形式为Windows服务而言，服务主函数（默认为ServiceMain）为导出函数，SCM会调用运行导出函数，所以不需要另外调用StartServiceCtrlDispatcher() API。

**6、调试Windows服务**

VS或者IDA

**6.1 IDA**

1）使用OD加载程序，直接在ServiceMain(0x1000C8C0)处右键->此处为EIP  
问题：会不会造成什么寄存器、栈的访问错误？一般不会，ServiceMain()是一个较独立的函数（名字都叫Main函数了），很少接收外界传来的参数，所以可以将EIP指过去直接运行。  
　　若需要接收参数，就需要创建一个，不然会内存访问错误。

**6.2 VS**

**6.2.1 本地**

　　在调试配置中生成你的服务。  
　　安装你的服务。  
　　从 “服务控制管理器”、“服务器资源管理器”或代码启动服务。  
　　使用管理凭据启动 Visual Studio,那么你就能附加到系统进程。  
　　（可选）在 Visual Studio 菜单栏上，选择“工具”、“选项” 。 在“选项”对话框中，选择“调试”、“符号”，选择“Microsoft 符号服务器”复选框，然后选择“确定”按钮 。  
　　在菜单栏上，从“调试” 或“工具” 菜单选择“附加到进程” 。 （键盘：Ctrl+Alt+P）这将显示“进程” 对话框。  
　　选择“显示所有用户的进程”复选框 。  
注意：连接类型：默认值；附加到：自动：本机代码；勾选左下角的“显示所有用户的进程”  
可以鼠标左键选择Available Processes中任意一个，接着在英文输入的情况下输入想要的exe文件。  
保证exe文件和pdb文件在同一个文件夹中。  
在调试过程中，每一次修改代码都需要重新编译，然后重新启动服务。

在“可用进程” 部分,为服务选择进程，然后选择“附加” 。  
可以查找并选择要附加到的一个或多个进程。不知道进程名字，可到https://docs.microsoft.com/zh-cn/visualstudio/debugger/attach-to-running-processes-with-the-visual-studio-debugger?view=vs-2019#BKMK\_Scenarios查询  
“附加到进程”对话框处于打开状态时，进程可以在后台启动和停止，因此正在运行的进程列表可能不总是最新内容。可随时选择“刷新”查看当前列表。

**6.2.2 附加到运城计算机的进程**

要求：远程调试器 (msvsmon.exe) 必须在远程计算机上运行。  
1、在 Visual Studio 中，选择“调试” > “附加到进程”（或按 Ctrl+Alt+P），打开“附加到进程”对话框。  
2、在大多数情况下，“连接类型”应为“默认”。 在“连接目标”框中，使用以下方法之一选择远程计算机：  
选择下拉箭头旁边的“连接目标”，并从下拉列表中选择计算机名称。  
　　键入中的计算机名称连接目标框，然后按Enter。  
　　验证 Visual Studio 将所需的端口添加到计算机名称，将出现在格式： <远程计算机名称 >： 端口  
　　选择查找按钮旁边连接目标框，以打开远程连接对话框。 远程连接对话框会列出本地子网上，或直接连接到您的计算机的所有设备。  
3、单击“刷新”，填充“可用进程”列表。  
4、在“可用进程”列表中，查找并选择要附加到的一个或多个进程。  
若要快速选择一个进程，请在“筛选进程”框中键入其名称或首字母。  
如果不知道进程名称，请浏览列表或参阅常见调试方案，了解一些常见的进程名称。  
若要查找所有用户帐户下运行的进程，请选择“显示所有用户的进程”复选框。  
5、在“附加到”字段中，确保已列出计划调试的代码类型。 默认的“自动”设置适用于大多数应用类型。  
若要手动选择代码类型：  
单击“选择”。  
在“选择代码类型”对话框中，选择“调试这些代码类型”。  
选择你想要调试的代码类型。  
选择 确定。  
6、选择“附加”。

提示：该进程的名称将与你的服务的可执行文件相同。出现 “附加到进程” 对话框。  
选择相应的选项，然后选择“确定” 以关闭对话框。  
备注：你现在处于调试模式。  
设置任意你想要在代码中使用的断点。  
访问服务控制管理器并操作你的服务，发送停止、暂停和继续命令以命中你的断点。