简介： 将守护进程和主进程都设计成windows服务，并使用命名管道进行通讯。

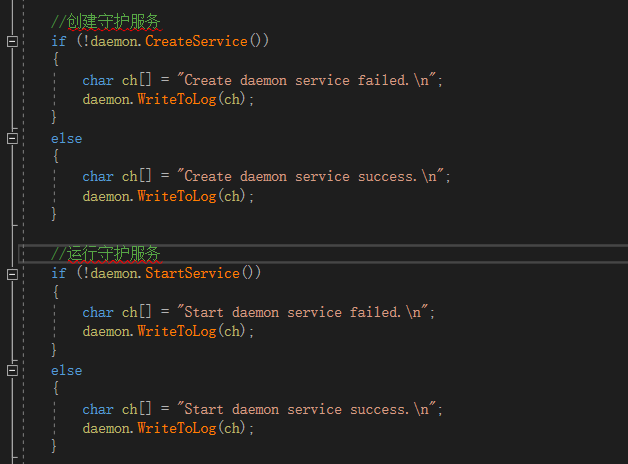
开发工具：VS2017

语言：C++

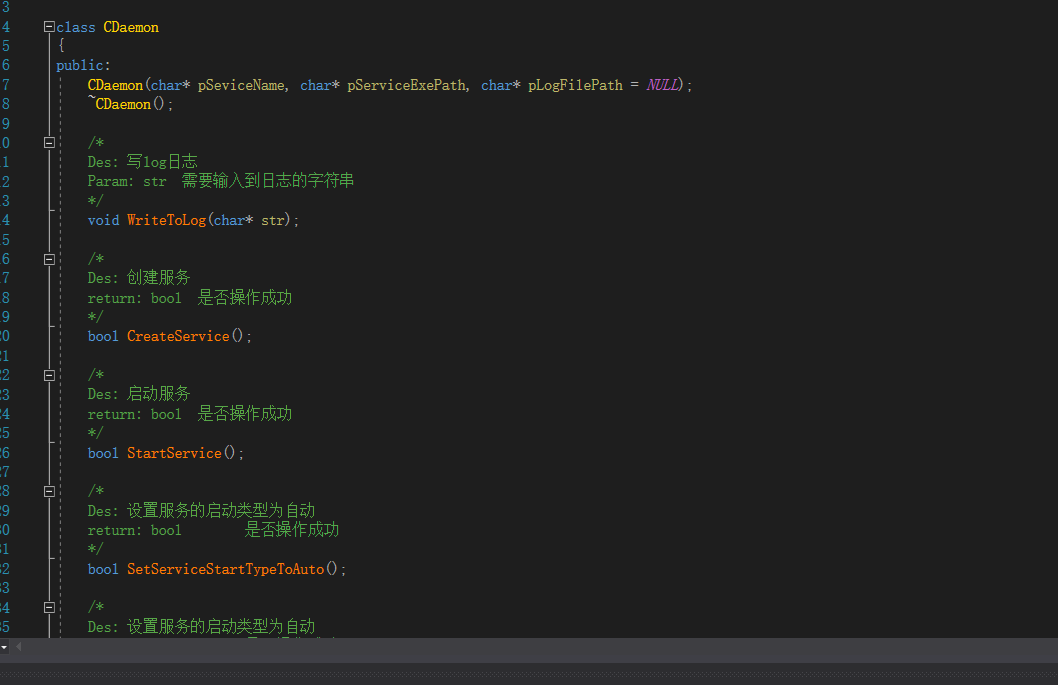
实现概述：创建一个启动守护服务的exe文件，启动守护服务时，守护服务去创建主服务，并检测并改正主服务的一切行为！

实现步骤：

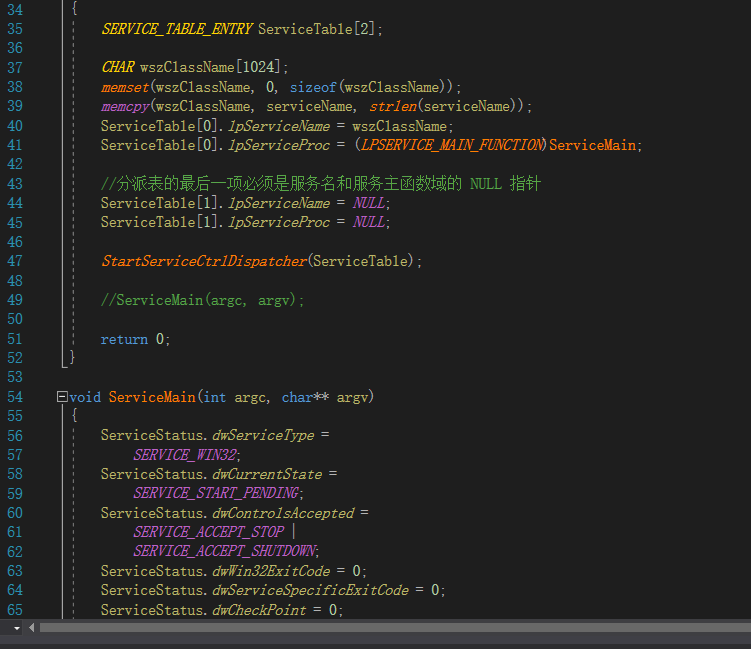
1. 创建一个启动守护服务的exe。（windows的服务的创建和属性修改必须要有“管理员权限”，即VS设置为管理员权限启动）。



CDaemon：封装了关于windows服务的所有的可能操作的方法。



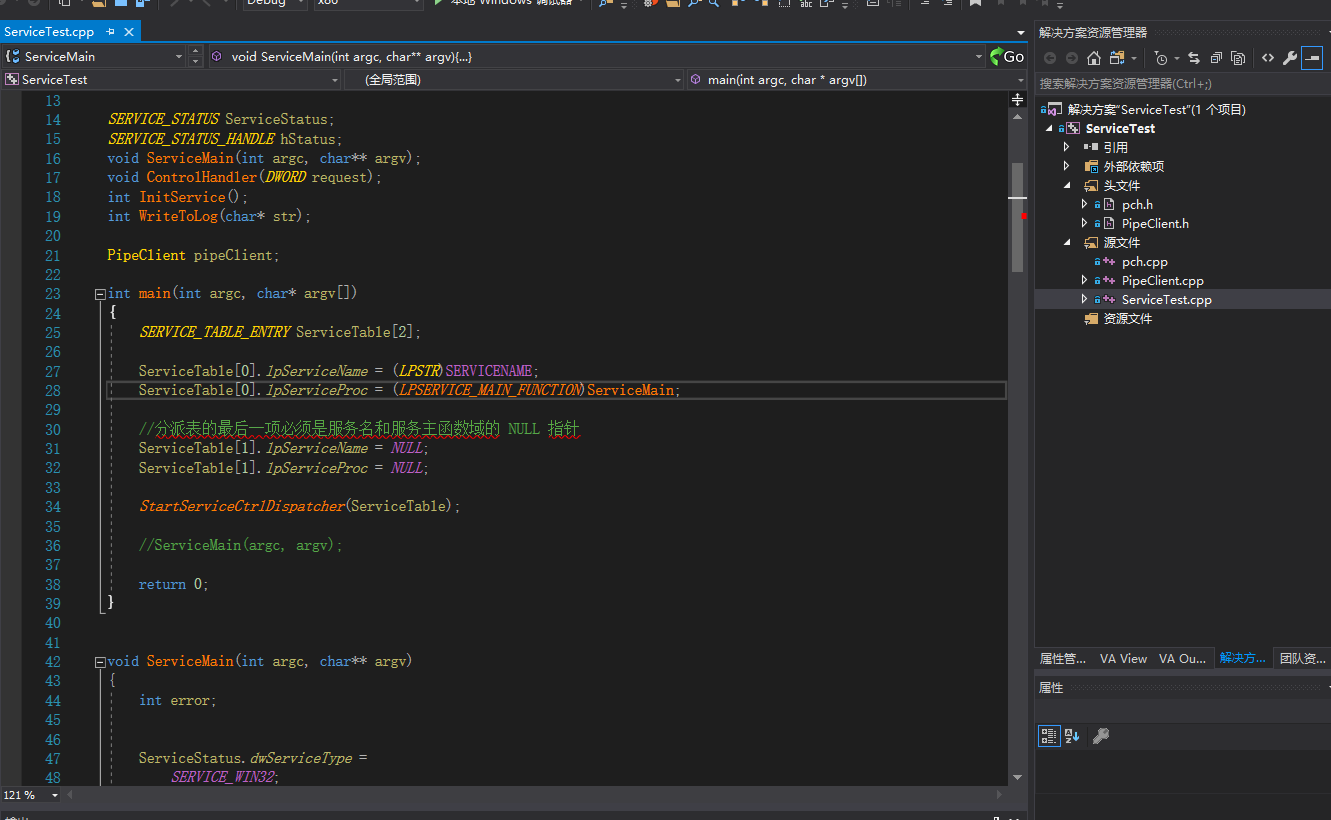
1. 创建守护服务：



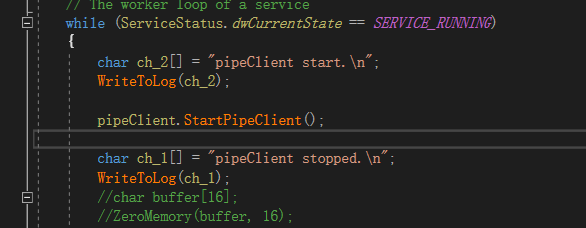
守护服务有一个命名管道的服务端！

注意：windows中的字符串的转换一定要避免强转！！！必须要根据其原始类型一步步转换，否则有可能出现你看到的名字和windows系统中跑的名字不一致导致打开服务失败的bug！

1. 创建主服务：



主服务的创建其实跟守护服务的创建几乎一致。主服务有一个命名管道的客户端。



1. 命名管道通讯：

由于守护进程的特殊使命，所有，守护进程端的命名管道服务器必须支持异步通讯！服务端每隔10s给客户端发送消息，如果客户端在一个超时时间范围内没有回发消息给服务端，则认定客户端卡死或者停止运行。此时根据主服务的状态去重启主服务或者重新创建并启动主服务。

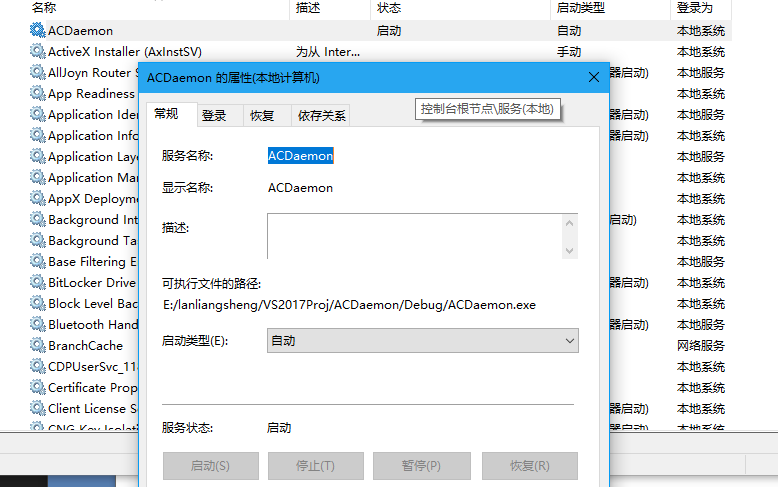
Windows下命名管道的异步有两种方式：（1）使用重叠结构OverLapped （2）使用完成实例（原理和IOCP一样）。

本例中，因为不涉及高并发，所以使用上述（1）中的方法。例子参考windows官方例子，注意：官方例子没有对重叠结构进行初始化，这会导致*GetOverlappedResult函数一直为false，error为ERROR\_INVALID\_PARAMETER！！！*

*Windows官方文档地址：*[*https://docs.microsoft.com/zh-cn/windows/desktop/ipc/named-pipe-server-using-overlapped-i-o*](https://docs.microsoft.com/zh-cn/windows/desktop/ipc/named-pipe-server-using-overlapped-i-o)

*做成结果：*

1. *守护服务为一直启动，无法点击停止等*

**

1. *主服务无论怎么操作让其停止，守护服务都会在2个超时周期内重启主服务。（1个超时周期后守护服务重启主服务，在1-2个超时周期内完成（理论上））。*

*注意：*

*Windows上删除服务： 使用管理员身份启动CMD，执行sc delete “服务名称”！ （sc也能创建一个服务。。。。。。） sc是windows的服务控制管理器！！！*