实现目的：监视文件创建，文件删除，文件重命名。屏蔽某名称文件的创建。

框架：MiniFilter。

驱动层逻辑：初始化框架。

定义过滤操作（IRP\_MJ\_CREATE和IRP\_MJ\_SET\_INFORMATION）。

初始化框架，开始过滤。

定义通信端口。

初始化通信端口。

开始通信。

用户层逻辑：打开通信端口。

主线程循环输入需要过滤的文件名。

开新线程接收驱动发回来的文件信息。

坑点：

1. 下载的7600框架样本NTDDI\_VERSION NTDDI\_WINXPSP2需要定义这个才能兼容XP系统。
2. CallBack过滤里面要以IRP\_MJ\_OPERATION\_END结尾。
3. 过滤分为Pre操作和Post操作，在Pre操作里面有一个FLT\_PREOP\_SUCCESS\_WITH\_CALLBACK和FLT\_PREOPSUCCESS\_NO\_CALLBACK代表有没有后置操作。
4. IRP\_MJ\_CREATE里面可能会创建文件夹而不是文件。经实验发现，PFLT\_CALLBACK\_DATA里面的Iopb->Paraments.Create.Options的四个字节，第一个字节表示CreateFile的DispositionValues，后面三个字节表示OptionFlags。OptionFlags里面有一个选项是CreateDirectory，如果存在就是文件夹。
5. IRP\_MJ\_CREATE里面有可能是打开文件而不是创建文件，这里处理了一下是打开文件还是创建文件。如果是FILE\_OPEN选项直接过滤掉，不属于想操作的类型。如果是FILE\_OPEN\_IF或者FILE\_OVERWRITE\_IF则去调用CreateFile（FILE\_OPEN）判定文件是否存在。如果存在说明是打开操作，否则是创建操作。
6. FltGetDestinationFileNameInformation 重命名操作则是有一点不一样，是调用这个函数获取想要改的名字。
7. 在PreCreate里面无法使用FltIsDirectory这个函数来判断是否是文件夹目录，但是在PreDelete蜜汁可以（在PostCreate则是可以使用这个函数判断，但是PostCreate有个问题就是一个创建消息会发送2-3次）。x
8. FltIsDirectory这个函数有点神奇，看名字就能看出是一个SendMessage的函数，但是这个函数同样可以从驱动接收信息，只不过SendBuffer不能为空SendLength不能为0。因为一定要发送数据之后才能接受信息回来。这里用了一个取巧的办法因为每次发送的字符串都是WCHAR最小的大小都是2字节，因此特意定义一个1字节的缓冲区，每次发送一个字节，这样在驱动进行判断，如果是1则说明应用层的目的是为了接收消息回去，而不是为了发送消息上来。
9. MiniFilter有一点很有趣，就是是一个NT驱动，但是可以不存在unload函数，只需要写一个unload函数挂到过滤器上即可，而不需要给Driver写一个卸载函数。
10. 需要导入fltMgr.lib这个lib。