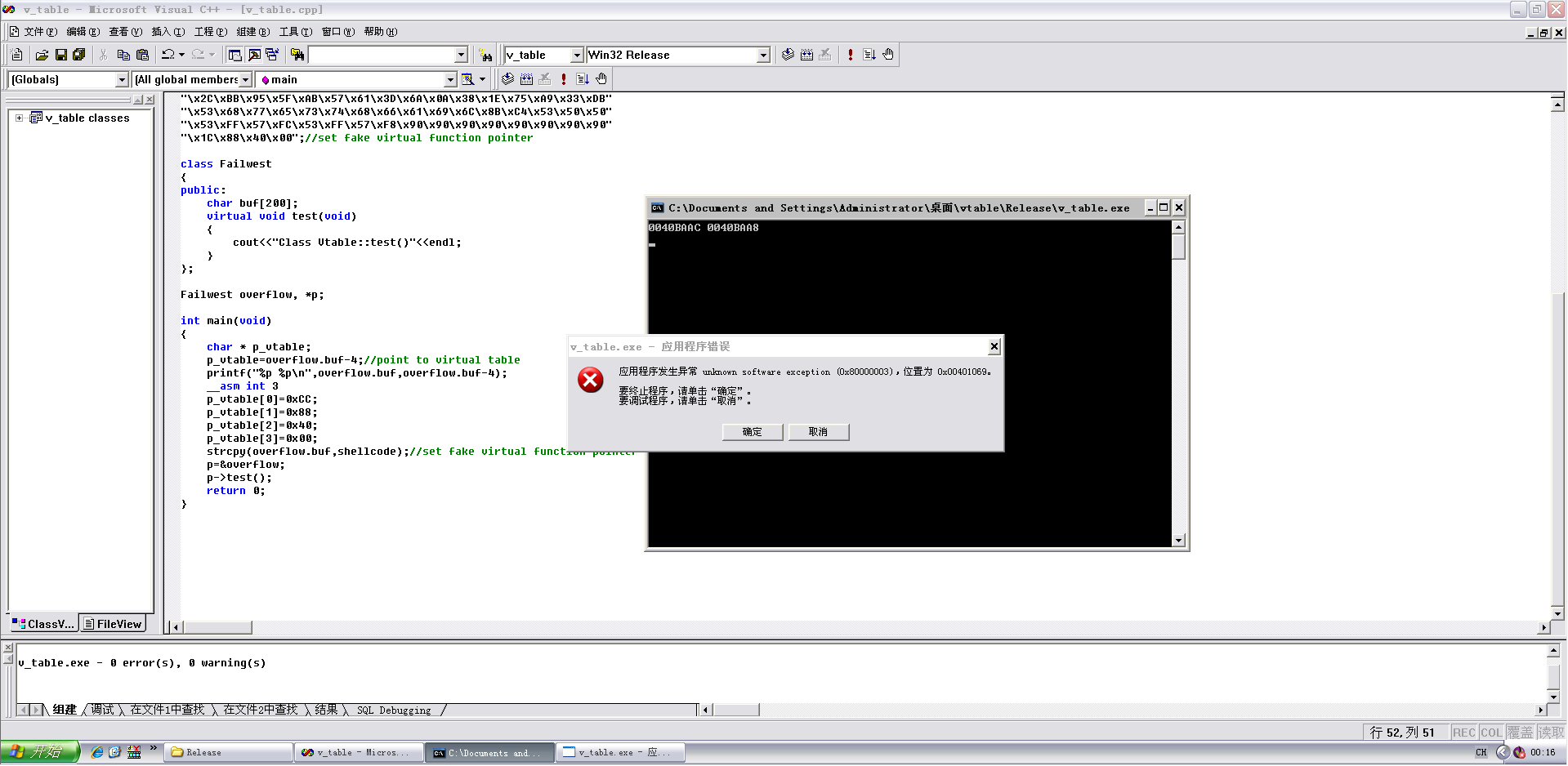
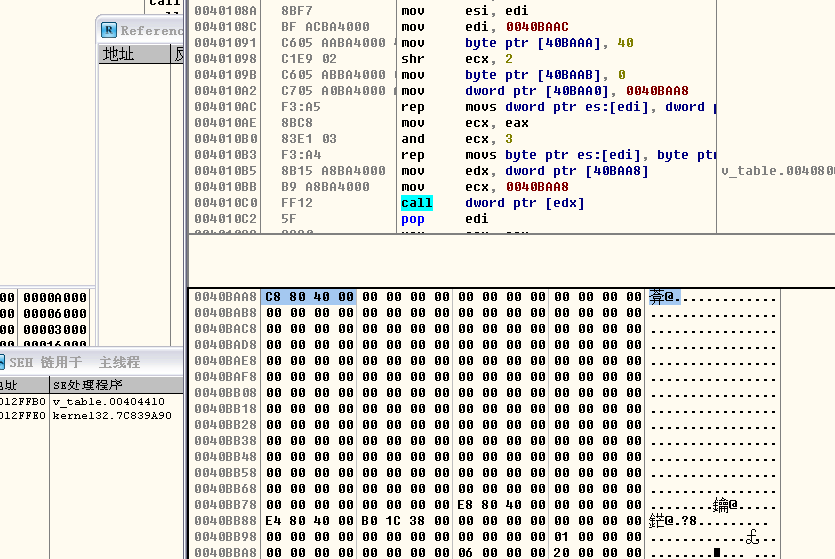
攻击虚函数是一件很有意思的事

当我拿到代码的时候，并不是急着去运行然后弹窗，因为很明显的有一种潜意识，跳转什么的地址肯定是需要自己重新确定的，所以我们来观察一下代码，可以看到手动修改了虚函数的指针，那我们就把存放虚函数指针的地址输出来

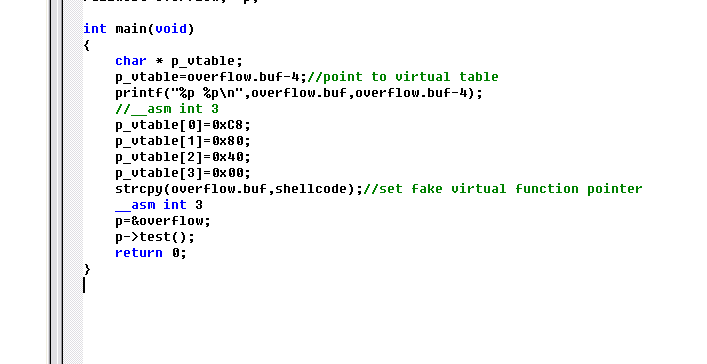


是不是觉得好机智，好了既然输出了存放虚函数指针的地址，那我们就去看看

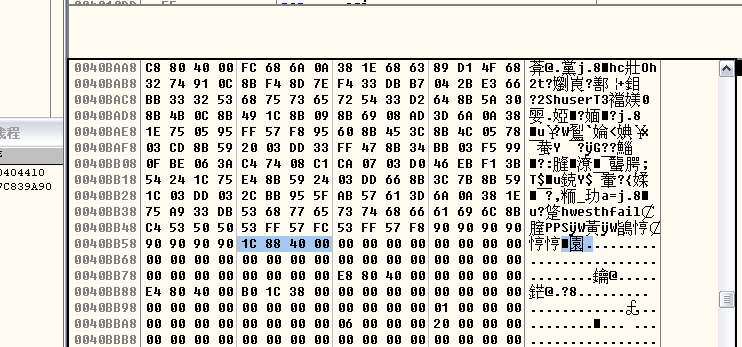


虚函数的地址是0x004080C8，后面就是buf的空间了

那我们现在修改一下代码，看看shellcode覆盖到哪里过

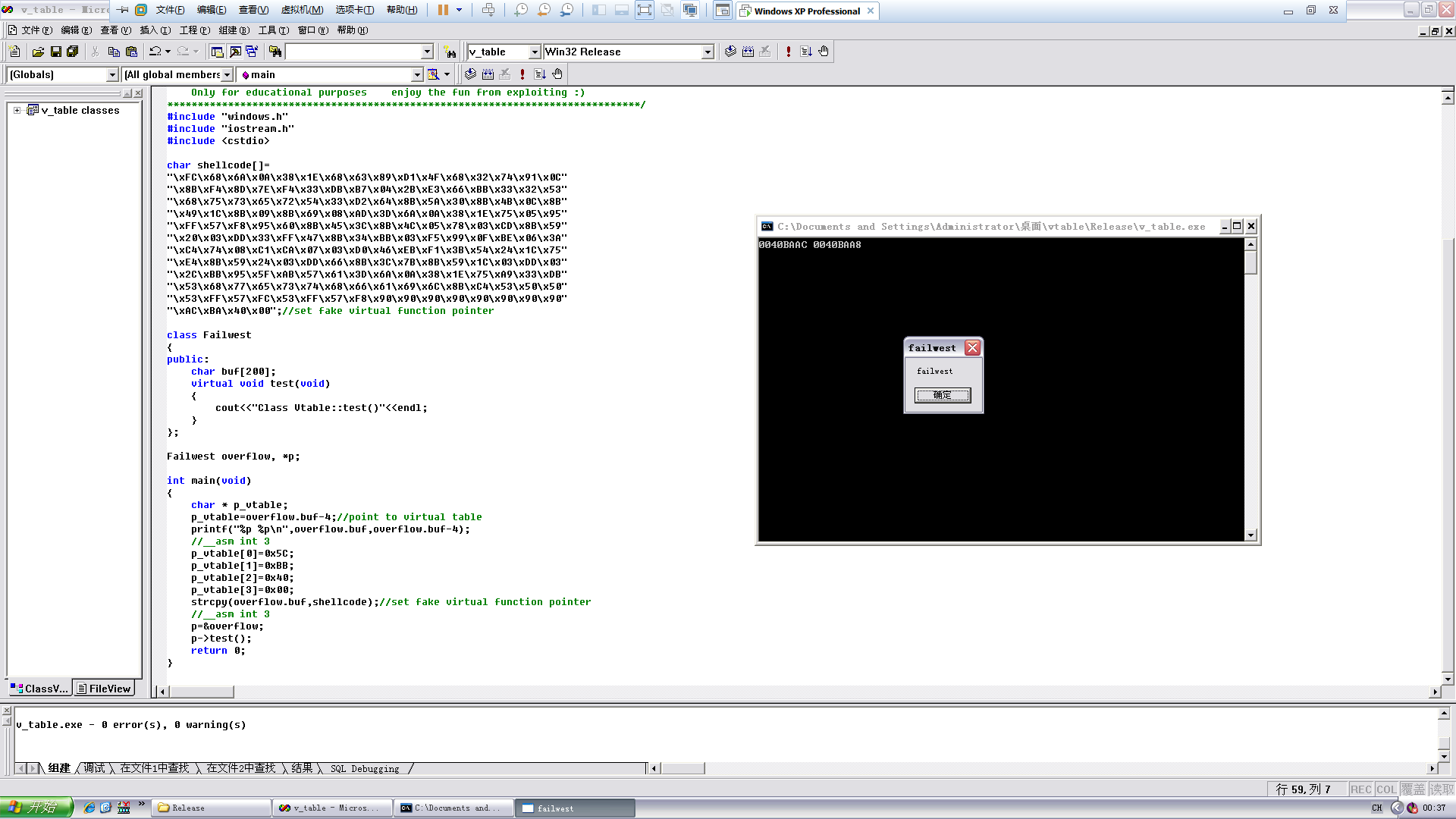


这样赋值完就可以断下来直接观察内存了



可以看到最后是0x0040BB5C，所以我们需要修改虚表指向这个地址，不是这个地址的值！！！！！！

然后我们需要修改这个地址的值为0x0040BAAC，这是shellcode在内存的起始位置，所以弄好后是这样的



世界就是这么美好

解释一下流程，我们手动修改虚表的地址直接指向shellcode的某处，然后某处存的值是shellcode的起始地址，这就模仿了先取虚表地址，然后再去虚表地址所指向的内存单元取虚函数地址的流程一样