《漏洞利用及渗透测试基础》实验报告

姓名：艾明旭 学号：2111033 班级：1121

**实验名称：**

堆溢出dword shoot 攻击实例

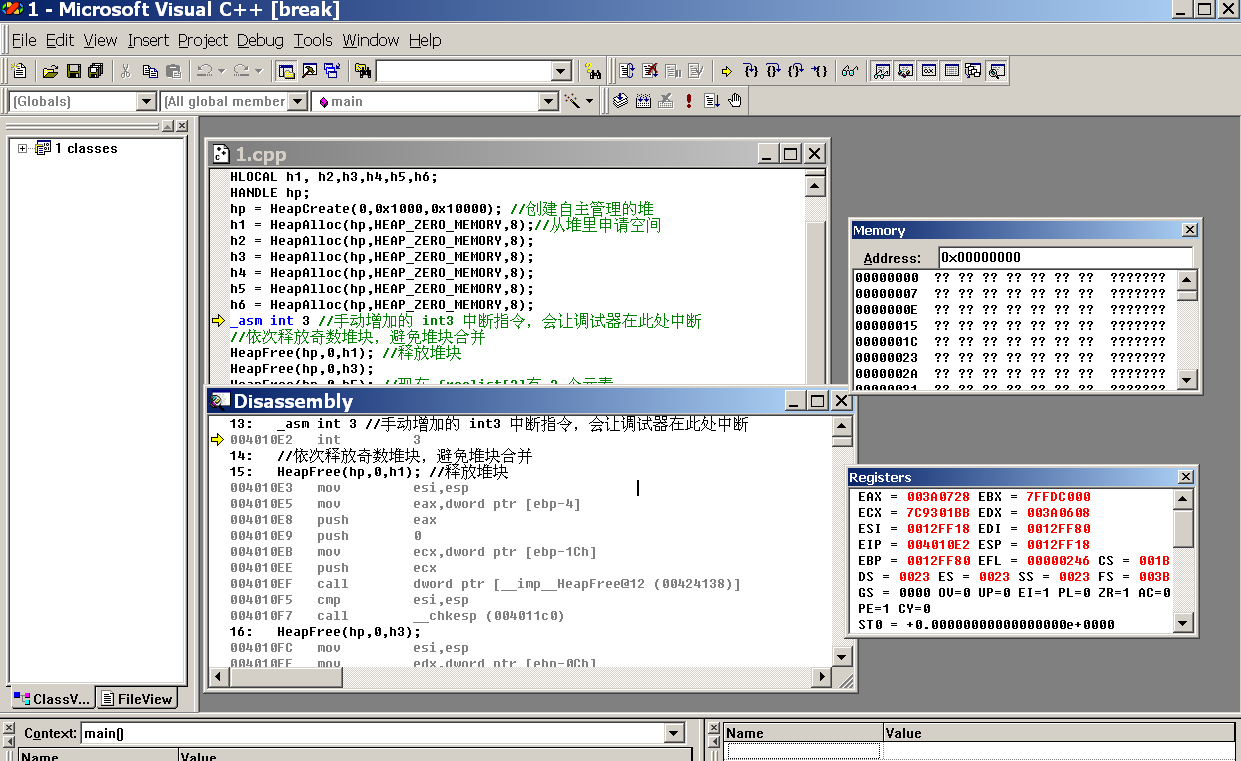
**实验要求：**

根据第四章示例，在XP环境下进行VC6反汇编调试，观察三次申请三次释放，堆溢出的实例和相应的反汇编窗口的变化，再次申请 8 字节的堆区内存，加上块首是 16 个字节，因此会从 freelist[2]所标识的空表中摘取第一个空闲堆块出来，即 h1。同时如果我们手动修改 h1 块首中的指针，应该能够观察到 DWORD SHOOT 的发生

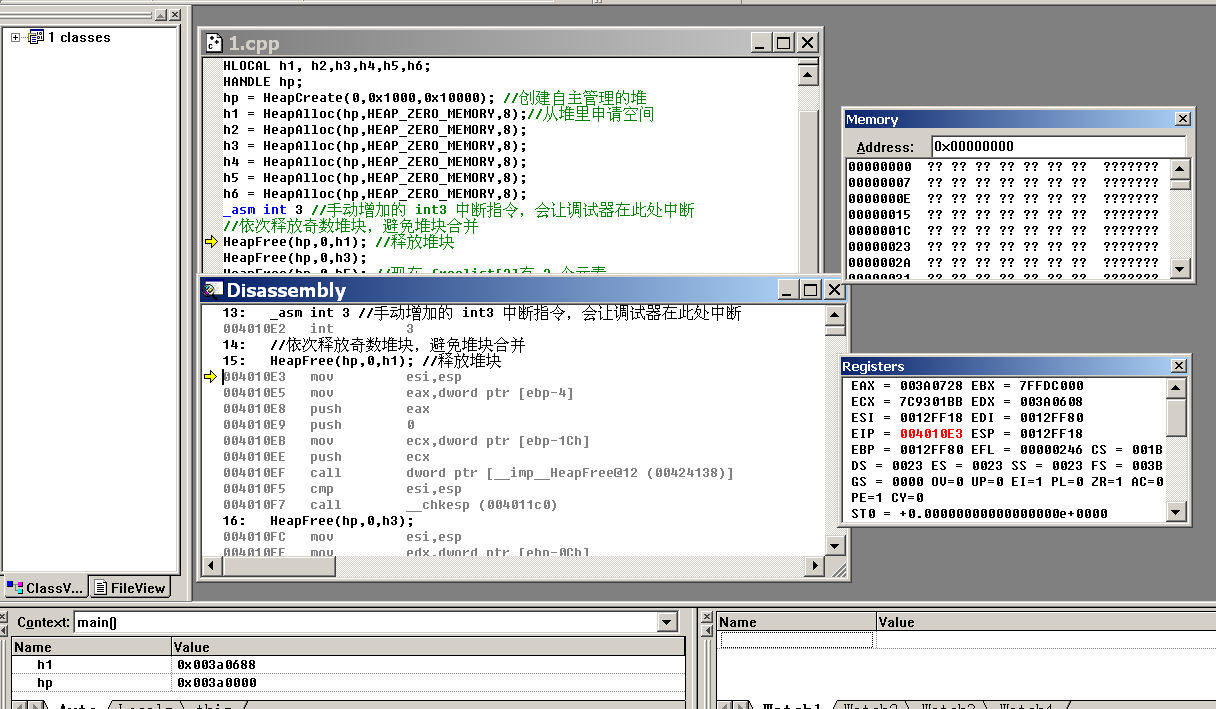
**实验过程：**

1. 进入VC反汇编

打开虚拟机的xp系统后，在vc6中创建win32的控制台文件，添加源文件后将代码复制到源文件中，debug无误后，得到相关的程序。打好断点之后，在窗口中调出相应的反汇编页面，观察相应的断点所在位置。

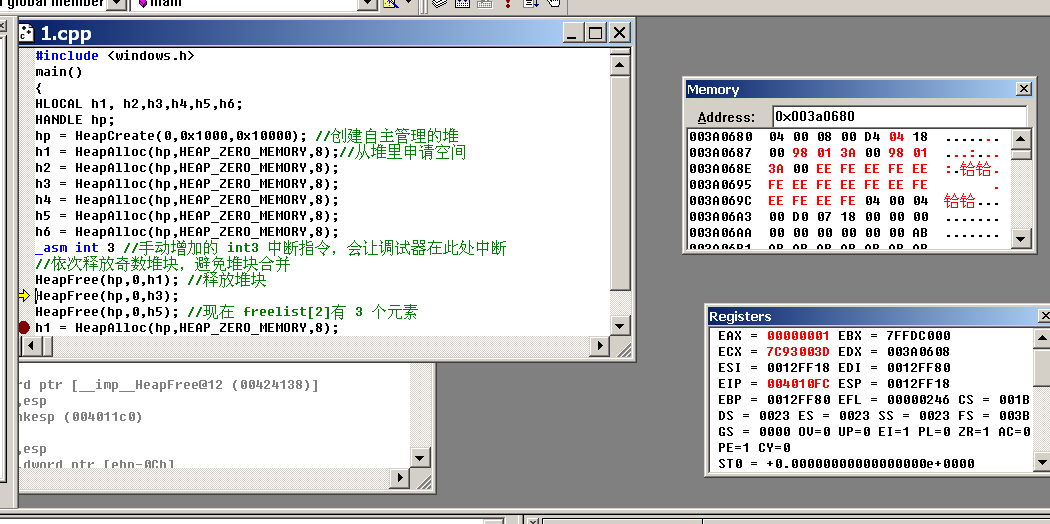


1. 参数入堆

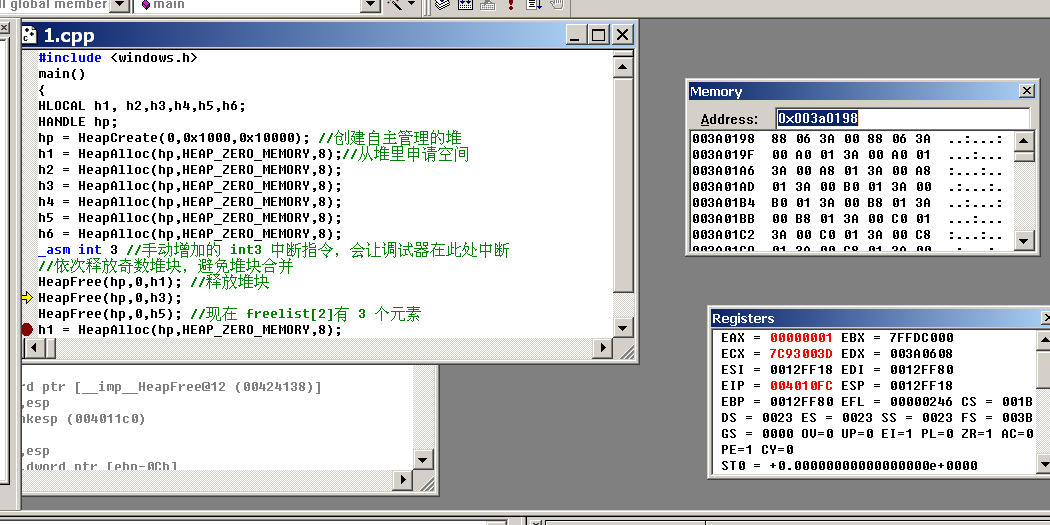
继续按下f10，我们可以看到程序继续运行，释放堆块的同时堆指针发生变化。  


1. 我们调回source页面，并且找到h1的地址，在memory里找到h1的状态，在h1的状态下，我们继续按f10执行堆释放的操作，最终达到了发现块首和块身都被释放后的状态。

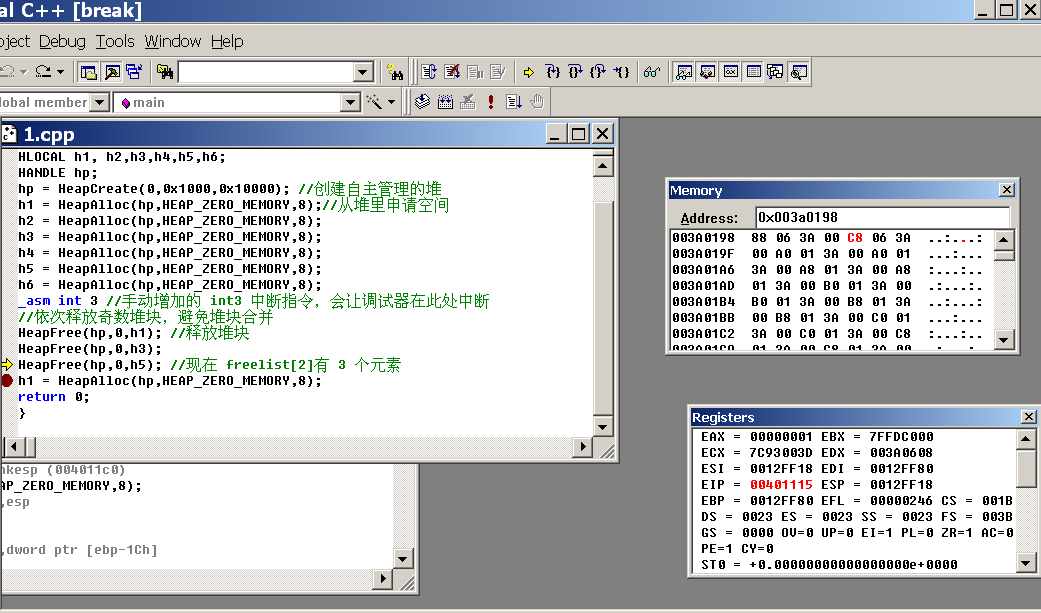
可以观察到前项和后向指针都变成了 00 3a 01 98，（因为freelist[1]没有东西放进来，所以前向和后向指针全部都是freelist[2]



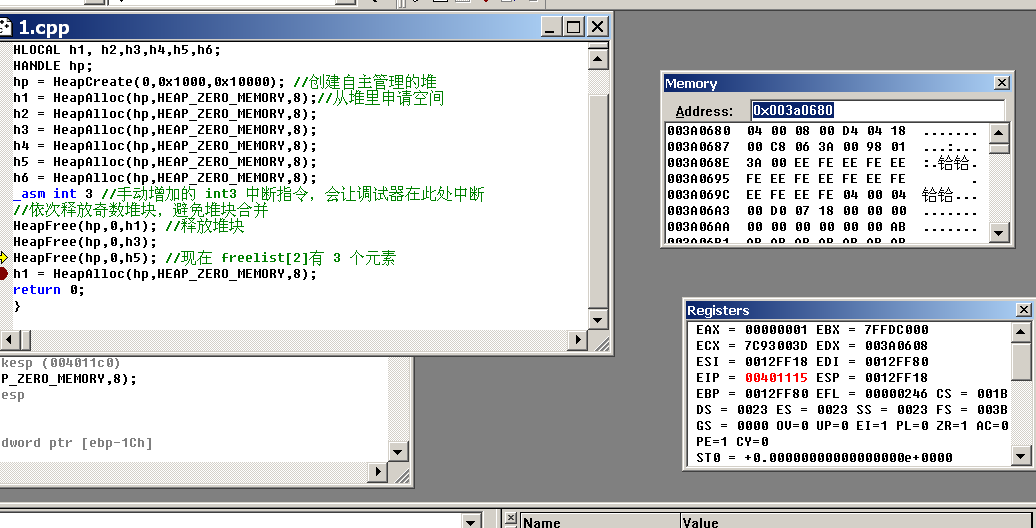
转到00 3 a 01 98观察栈指针的变化，前后向指针都是00 3a 06 88。由此我们可以确定freelist[2]里面就是h1的值。



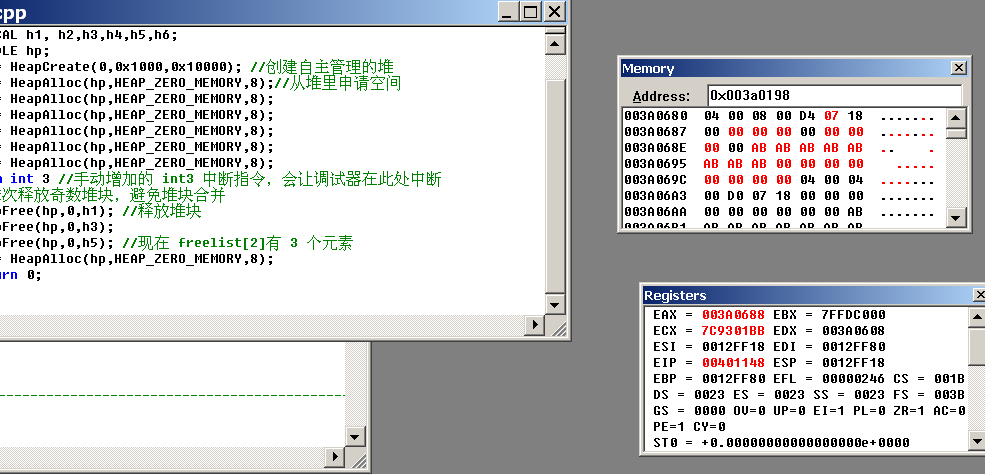
再按f10进入下一步，我们可以发现后向指针发生了变化，变成了h3。

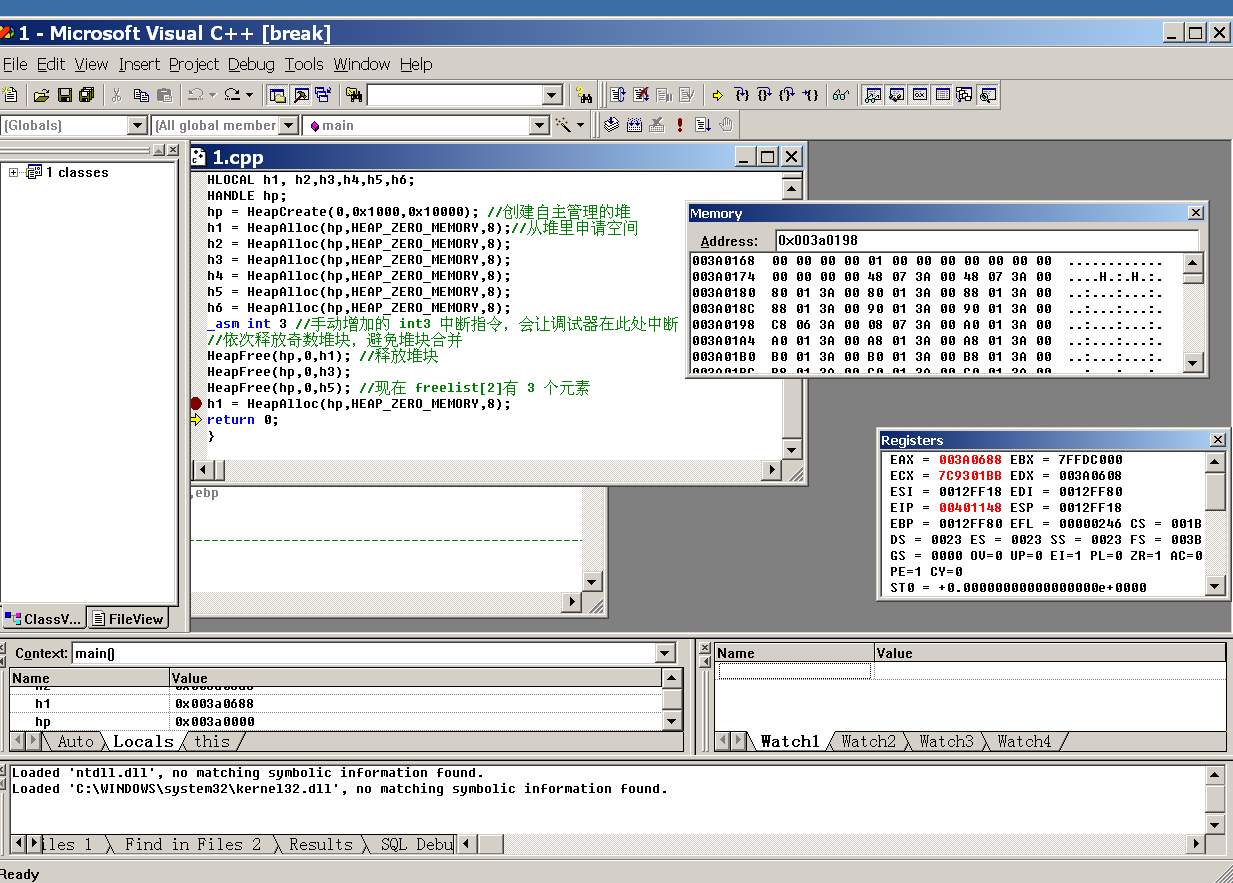


转回00 3a 06 80，可以发现前向指针发生了变化。

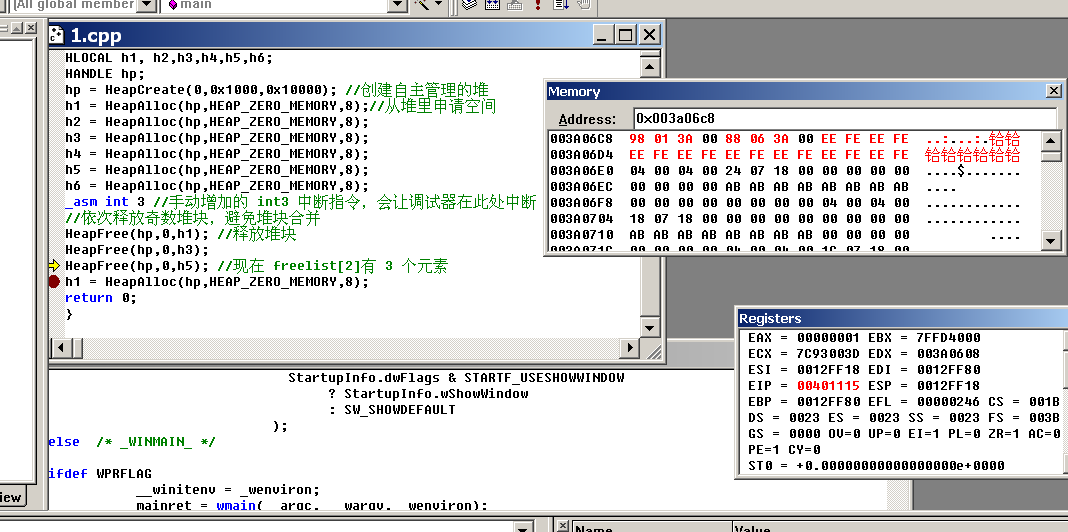


再次进行下一步操作，我们可以发现变成了之前指向h3的指针变成了指向h5





00 3a 06 c8的地方也发生了变化，前后向指针分别为003a0198和003a0688



把h1处相应的地址用ollydbg进行更改，就成功的进行了堆溢出dword shoot攻击。

**心得体会：**

通过实验，我掌握了堆的生成和释放的方法，通过堆的生成和释放的过程，利用memory地址栏的查询功能，我们可以看到堆的前向指针和后向指针在进行进入和释放的过程，其值不断随着h1,h3,h5的进入和出堆而变化。

同时，我们可以利用ollydbg对实验当中的堆的地址进行改变，从而进行堆溢出的dword shoot 攻击。