《漏洞利用及渗透测试基础》实验报告

姓名：孙蕗 学号：2112060 班级：信安1班

**实验名称：**

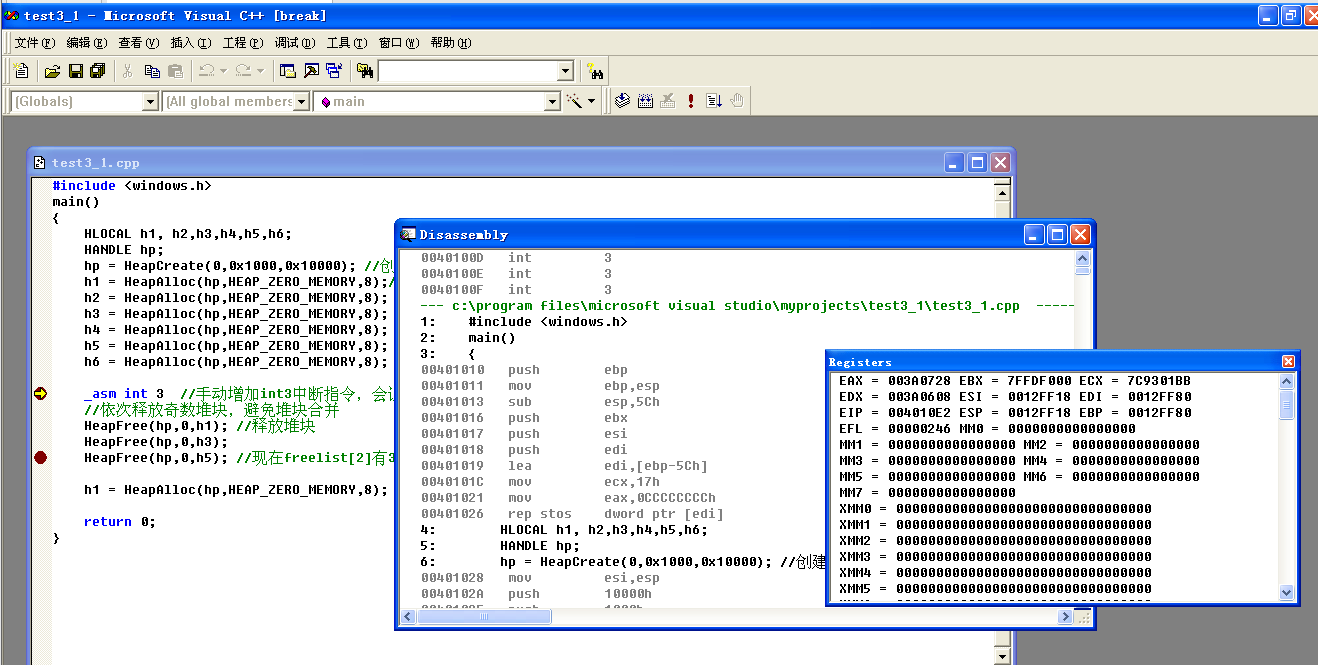
堆溢出Dword Shoot模拟实验

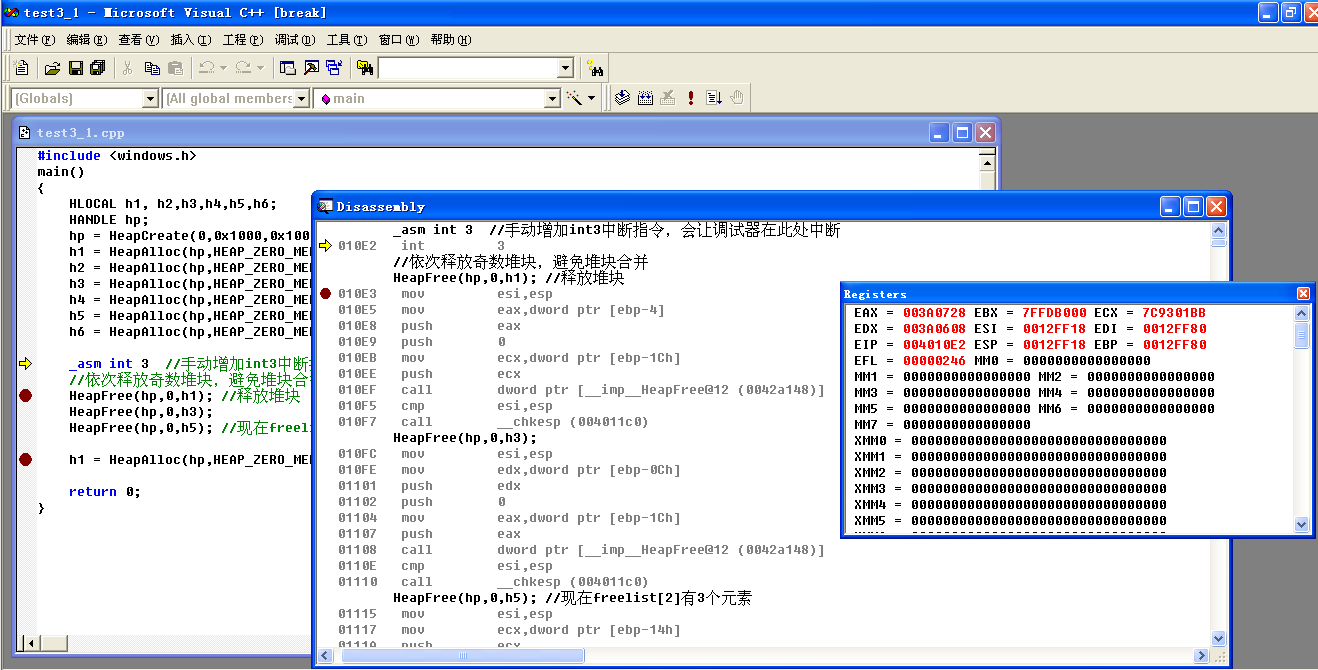
**实验要求：**

以第四章示例4-4代码为准，在VC IDE中进行调试，观察堆的管理结构，记录Unlink节点时的双向空闲链表的状态变化，了解堆溢出漏洞下的Dword Shoot攻击。

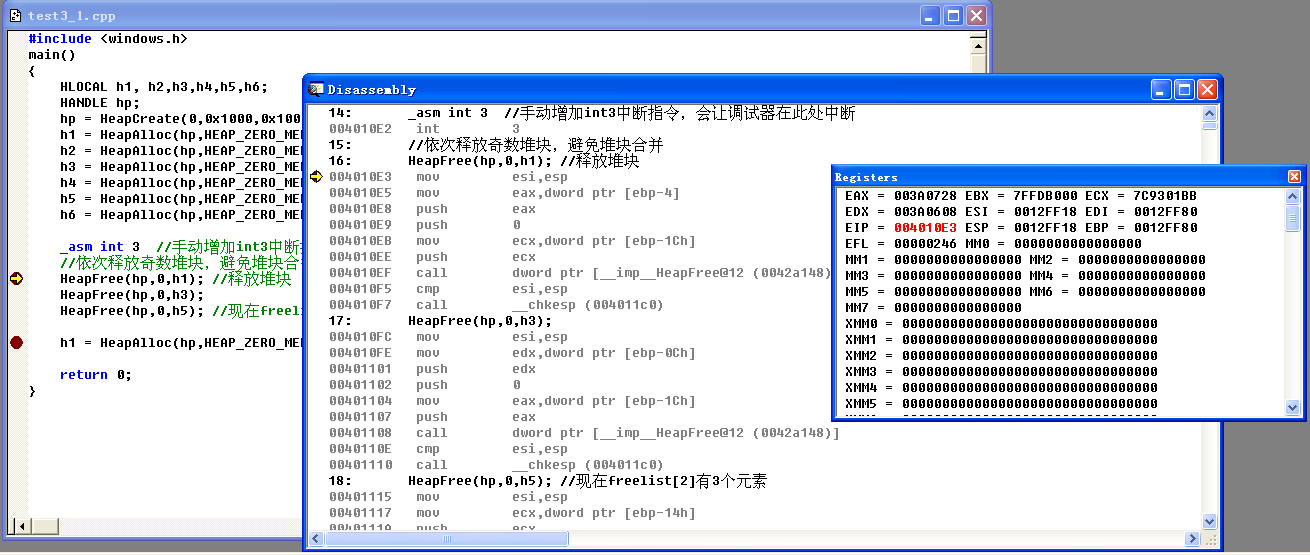
**实验过程：**

添加断点，debug反汇编代码

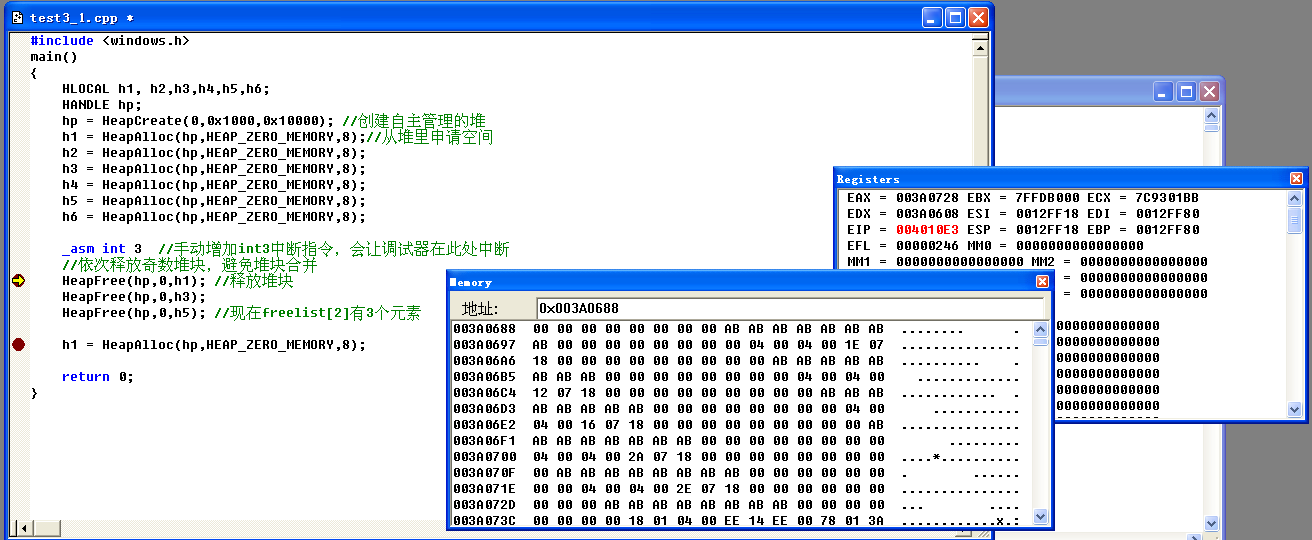
****

****

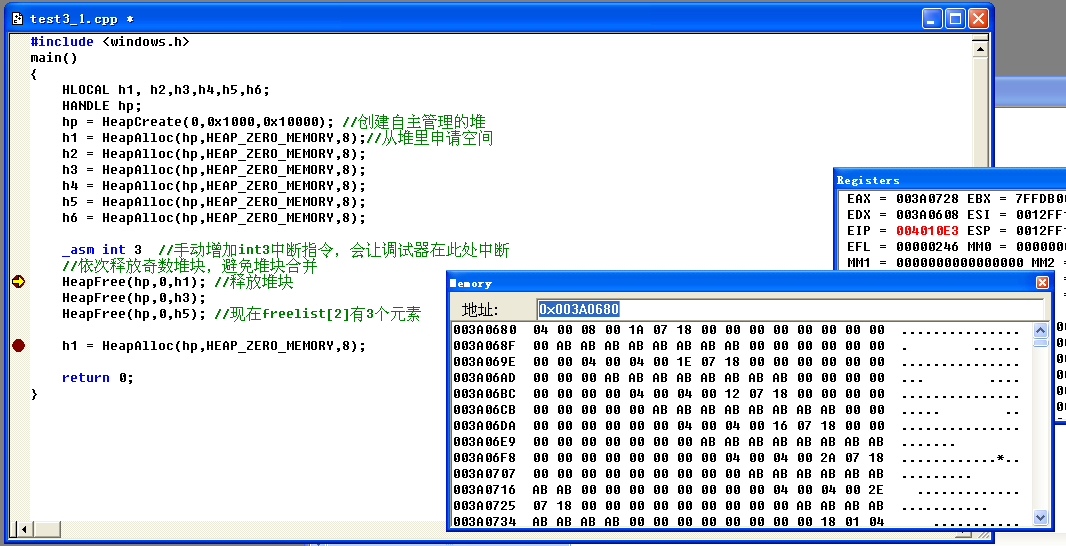
F10 单步执行



hp为0x003A0000，h1为0x003A0688,h3为0x003A06C8,h5为0x003A0708



0x003A0688加上块首8个字节，查看0x003A0680。

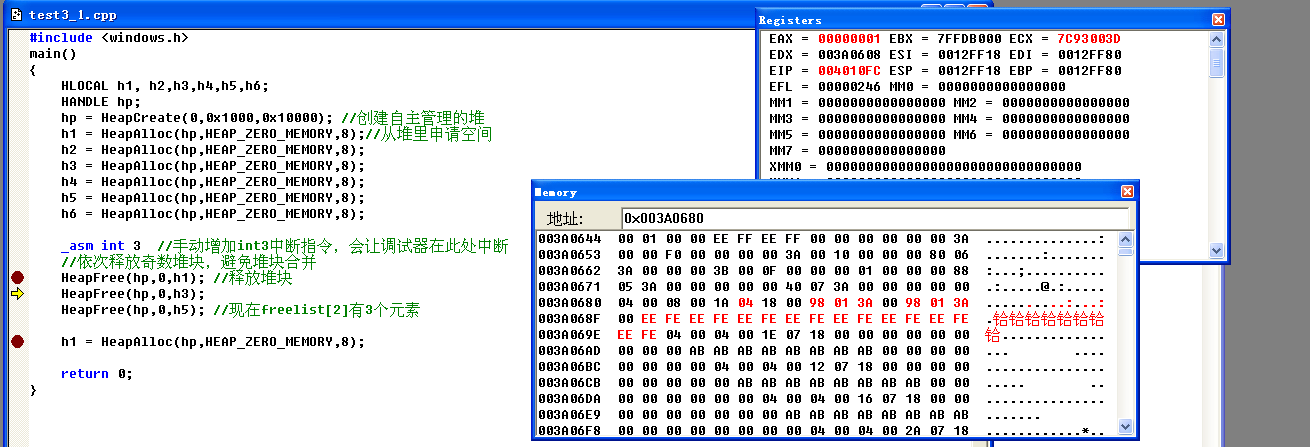


F11，当前块的一些状态会被改写，块首，flink，blink发生改变。flink=0x003A0198，blink=0x003A0198。0x003A0198也就是freelist[2]的地址。

块首

flink

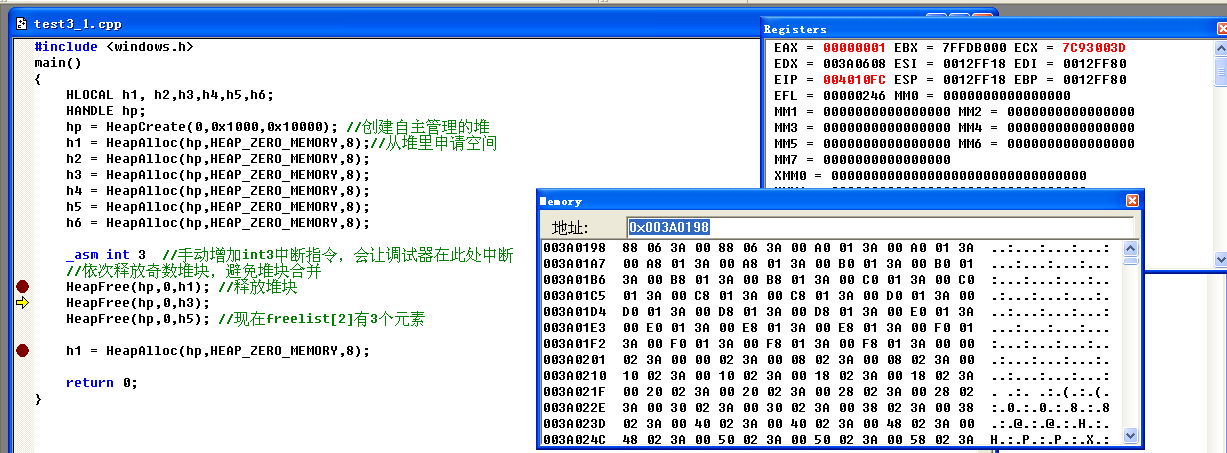
blink



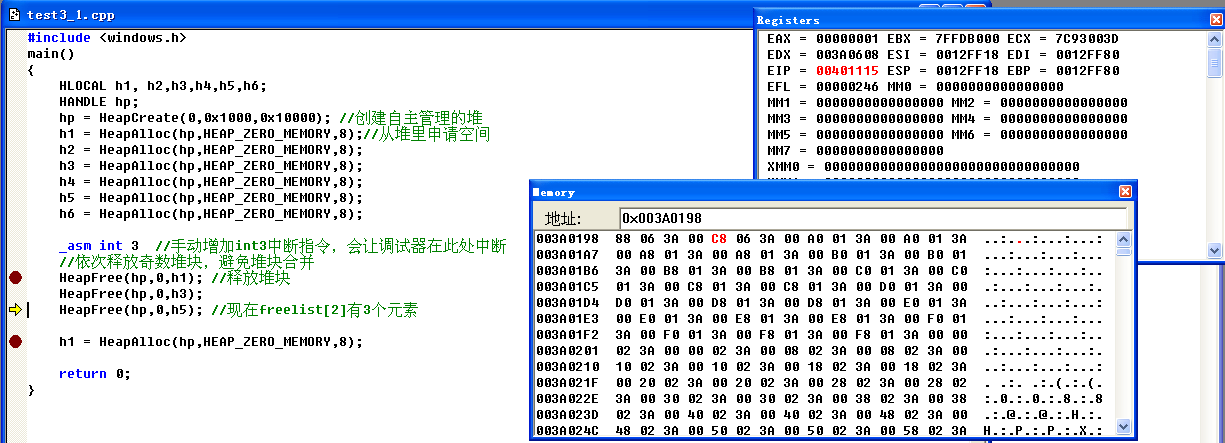
flink

blink

查看freelist[2]地址0x003A0198，flink,blink都是0x003A0688(h1)



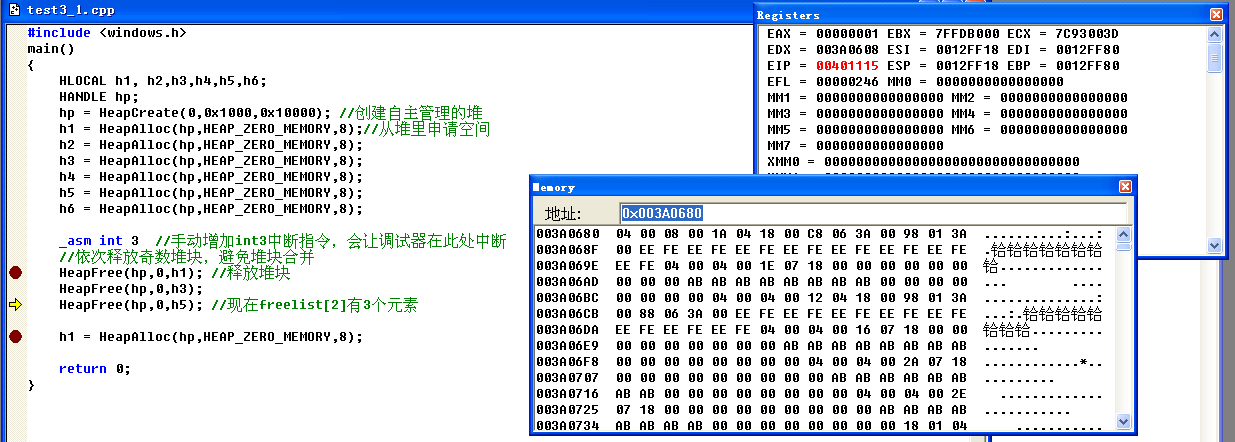
F11，查看freelist[2]0x003A0198,flink=0x003A0688(h1),blink=0x003A06C8(h3)



flink

blink

查看0x003A0680,也发生了变化，flink=0x003A06C8（h3）,blink=0x003A0198（freelist[2]）



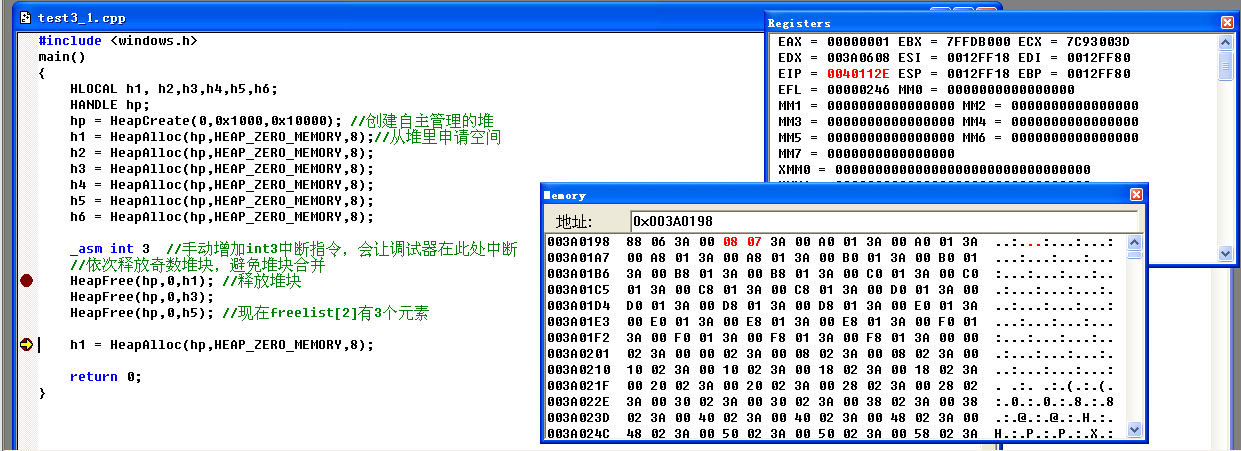
flink

blink

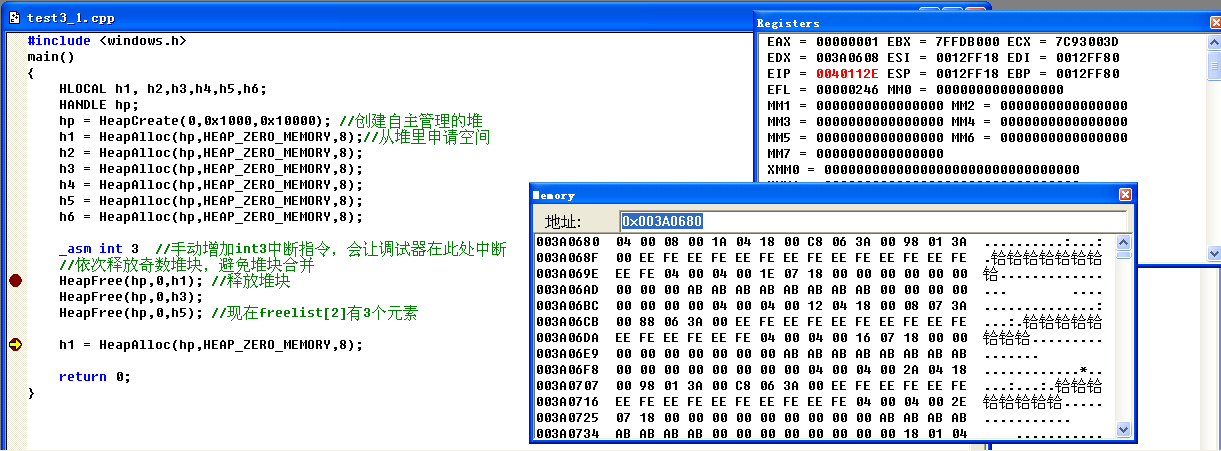
F11,查看freelist[2]0x003A0198,flink=0x003A0688(h1),blink=0x003A0708(h5)

flink

blink



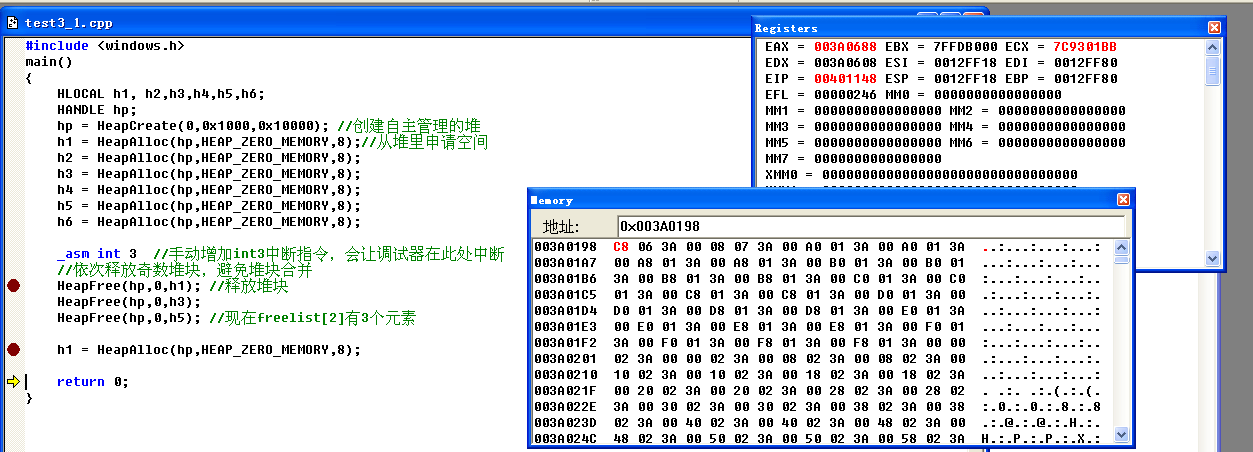
查看0x003A0680,如果将h1的flink和blink改写成特定地址和数值，那么将完成一次dword shoot攻击。



此时freelist[2]链表状态为freelist[2]<=>h1<=>h3<=>h5

F11,查看freelist[2]0x003A0198

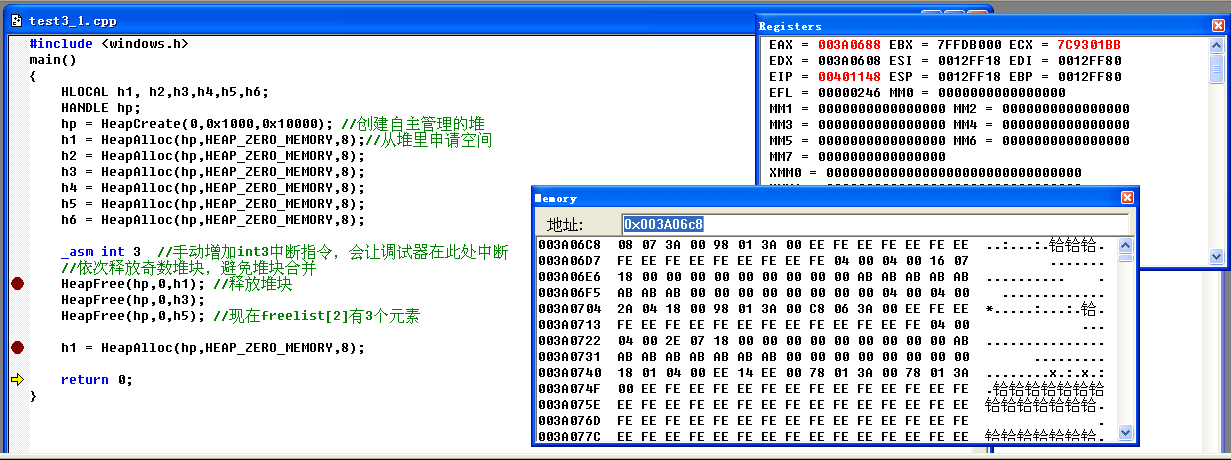
摘掉h1后，flink=0x003A06C8(h3),blink=0x003A0708(h5)



flink

blink

查看0x003A06C8（h3）,flink=0x003A0708(h5),blink=0x003A0198(freelist[2])



flink

blink

**心得体会：**

如果通过堆溢出覆写一个空闲堆块的块首前向指针flink和后向指针blink，可以精心构造一个地址和数据，当这个空闲堆块从链表里卸下的时候，就获得一次向内存构造的任意地址写入一个任意数据的机会。基于dword shoot攻击，攻击者甚至可以劫持进程，运行植入的恶意代码。例如，当构造的地址为重要函数调用地址、栈帧中函数返回地址、栈帧中SEH句柄等时，写入的任意数据可能就是恶意代码的入口地址。