《堆溢出Dword Shoot模拟实验》实验报告

姓名：姜志凯 学号：2011937 班级：信安班

**实验名称：**

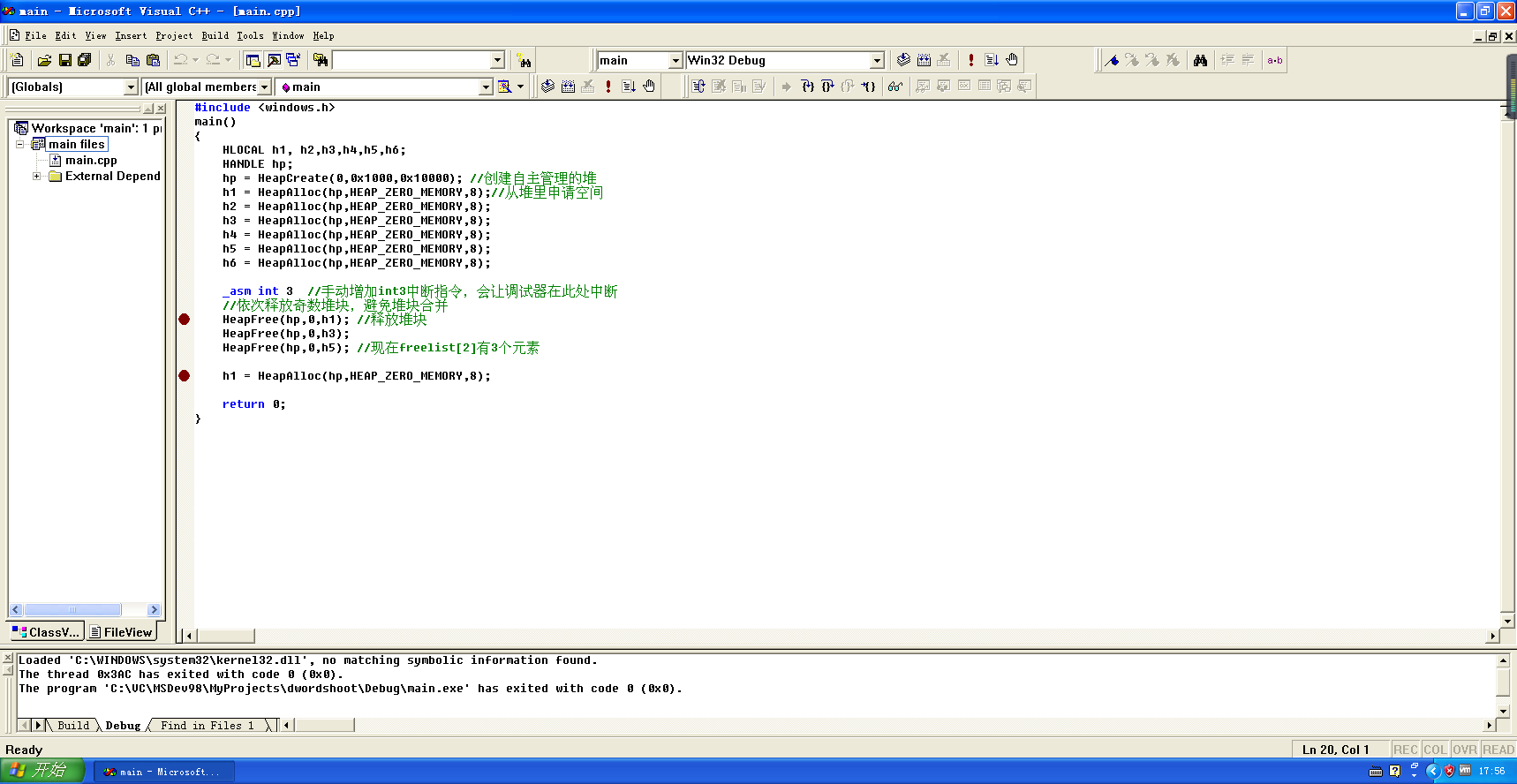
堆溢出Dword Shoot模拟实验

**实验要求：**

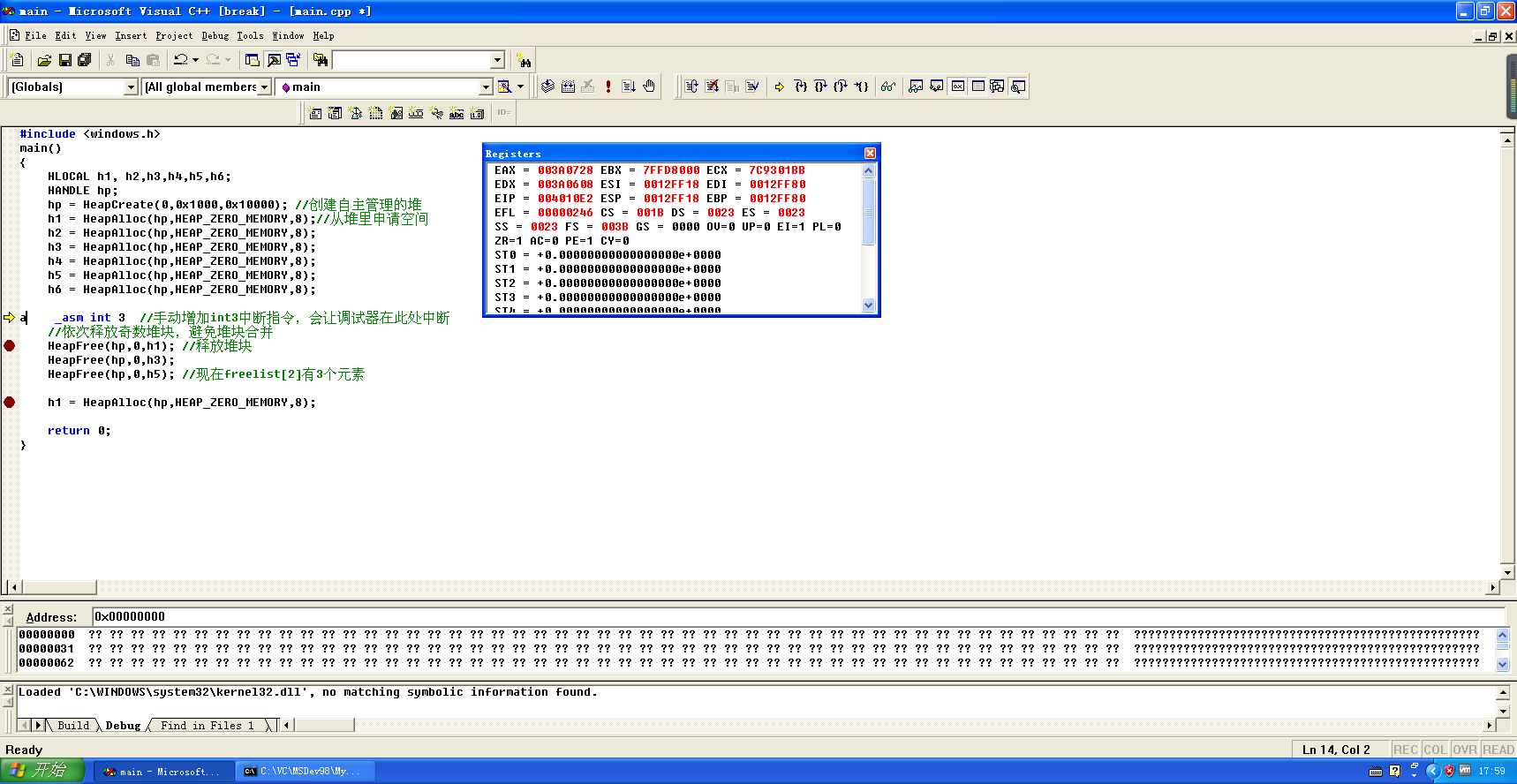
以第四章示例4-4代码为准，在VC IDE中进行调试，观察堆管理结构，记录Unlink节点时的双向空闲链表的状态变化，了解堆溢出漏洞下的Dword Shoot攻击。

**实验过程：**

1. 进入VC并设置断点



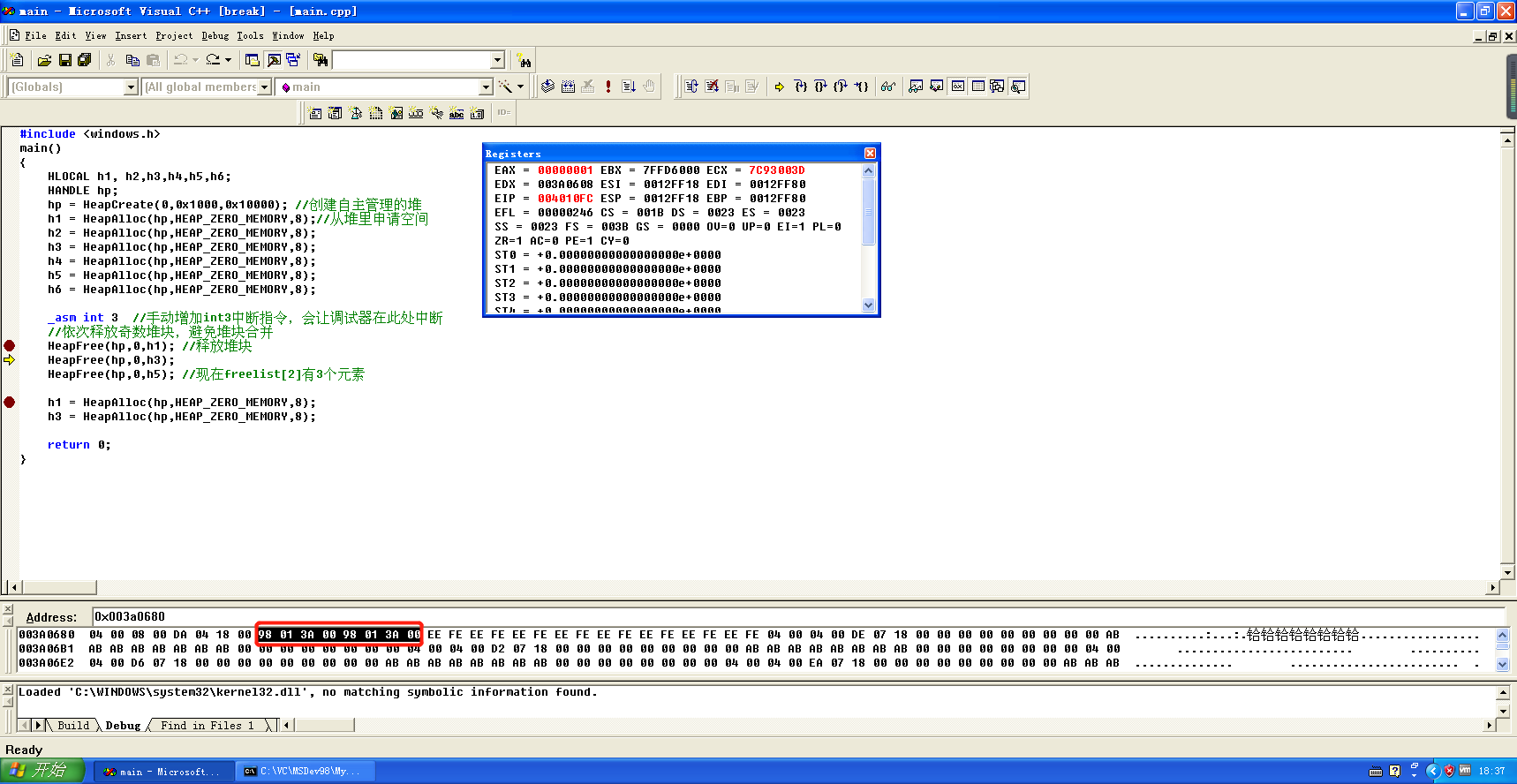
1. f5进入调试模式

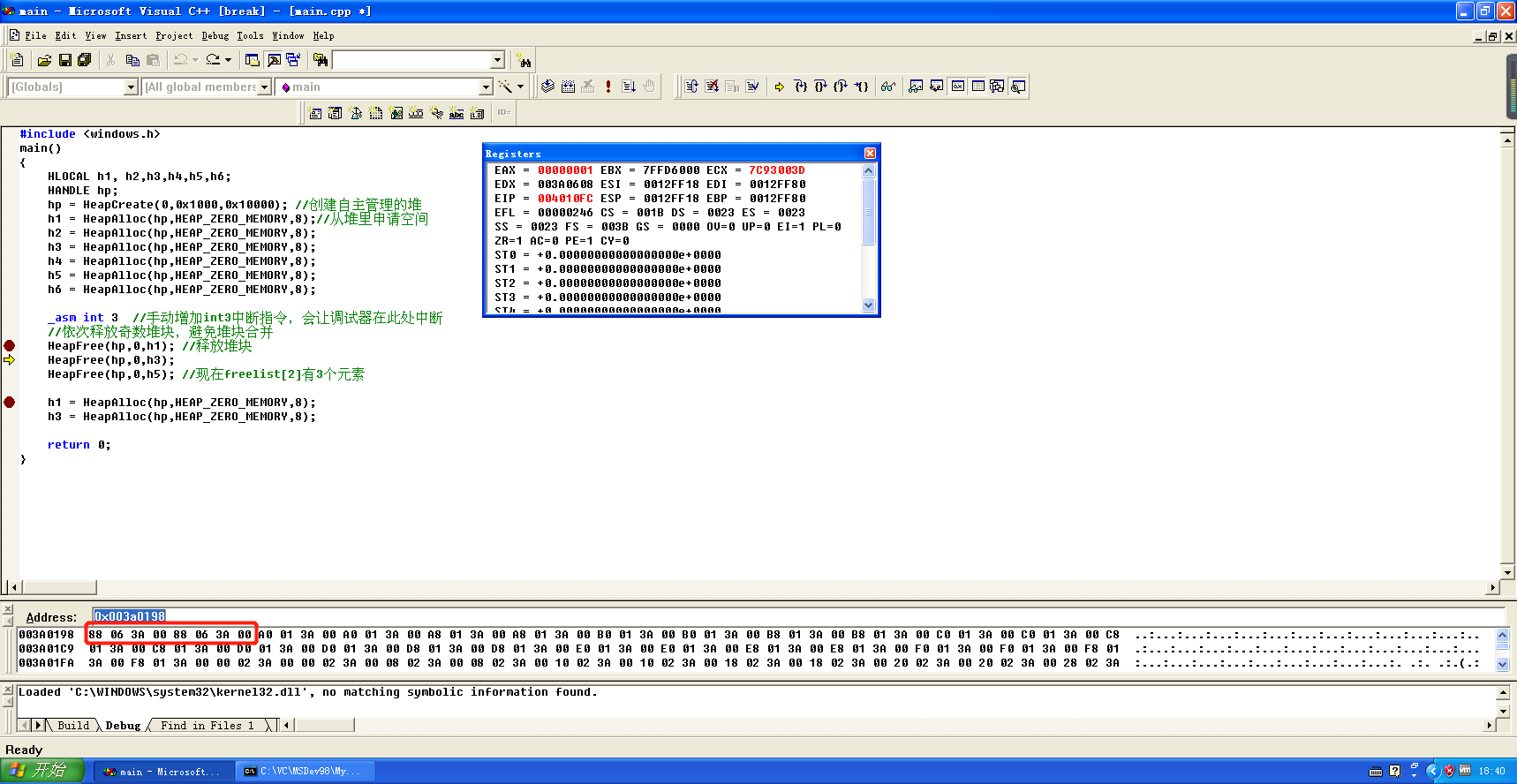


创建的自主管理的堆hp的首地址为0x003a0000，从堆中申请6个堆块（都是从freelist[0]中卸下的），第一个堆块的快首地址为0x003a0680，后以此类推，可用鼠标做停留动作查看（这六个堆块现均为占有态）。

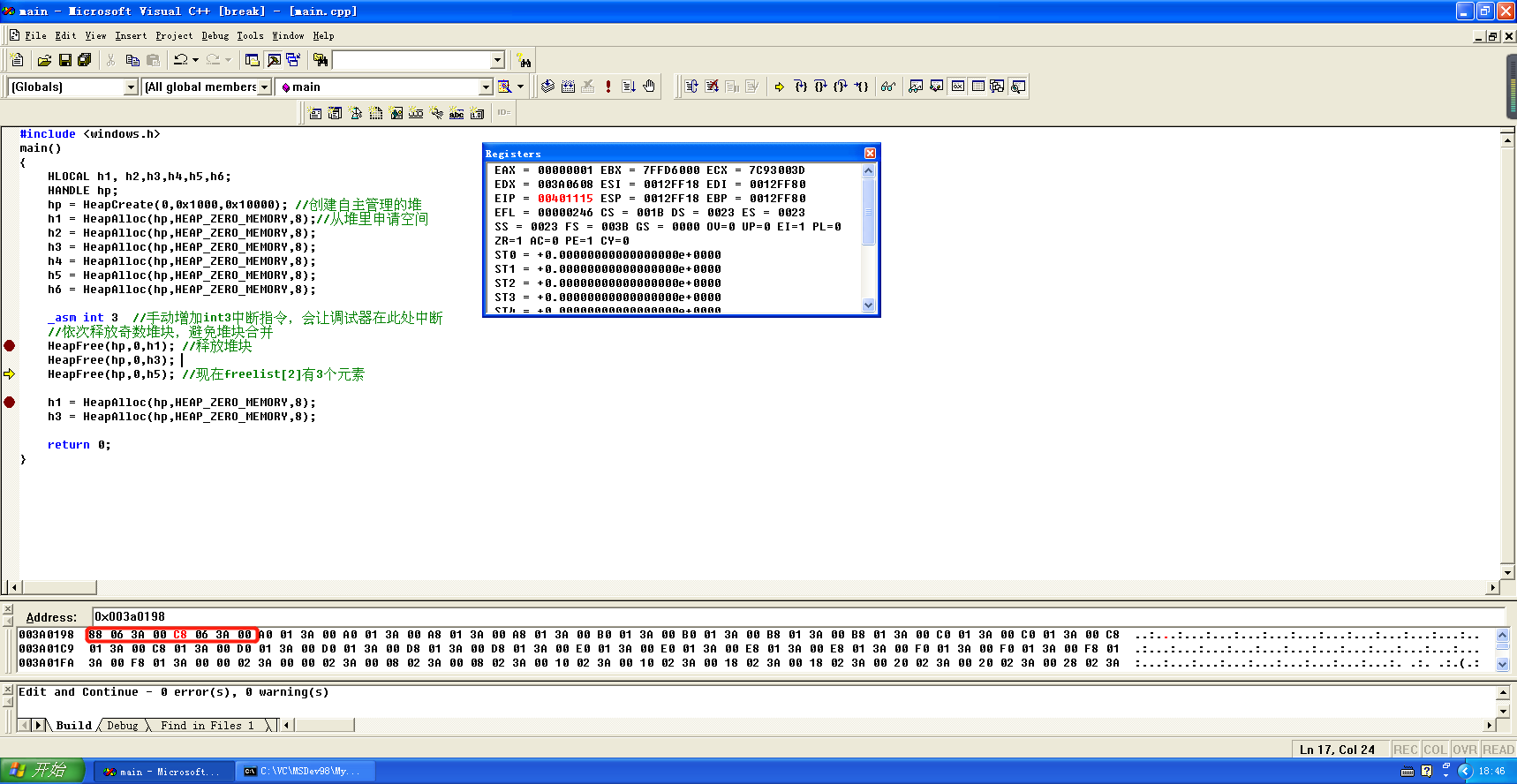
1. 然后依次释放堆块h1、h3、h5，由于堆块的大小均为16，所以卸下的堆块全部链入freelist[2]后边

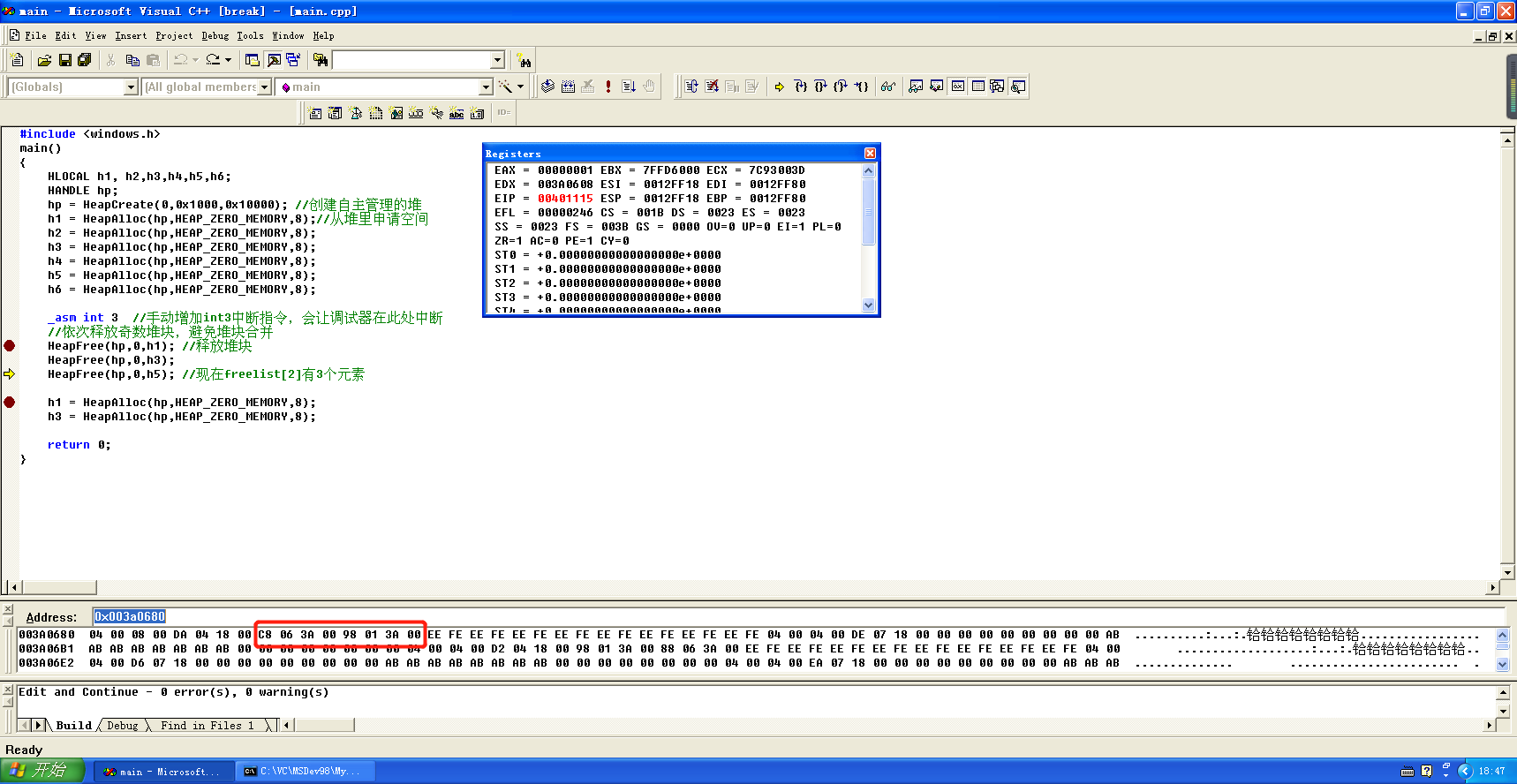
先释放h1，可以通过h1的地址0x003a0680查看该堆块的前后向指针，发现均为0x003a0198，因为freelist[2]原先为空的，所以0x003a0198即为freelist[2]的地址，此时freelist[2]和h1的前后向指针均指向对方（这里指针指向的是块身）

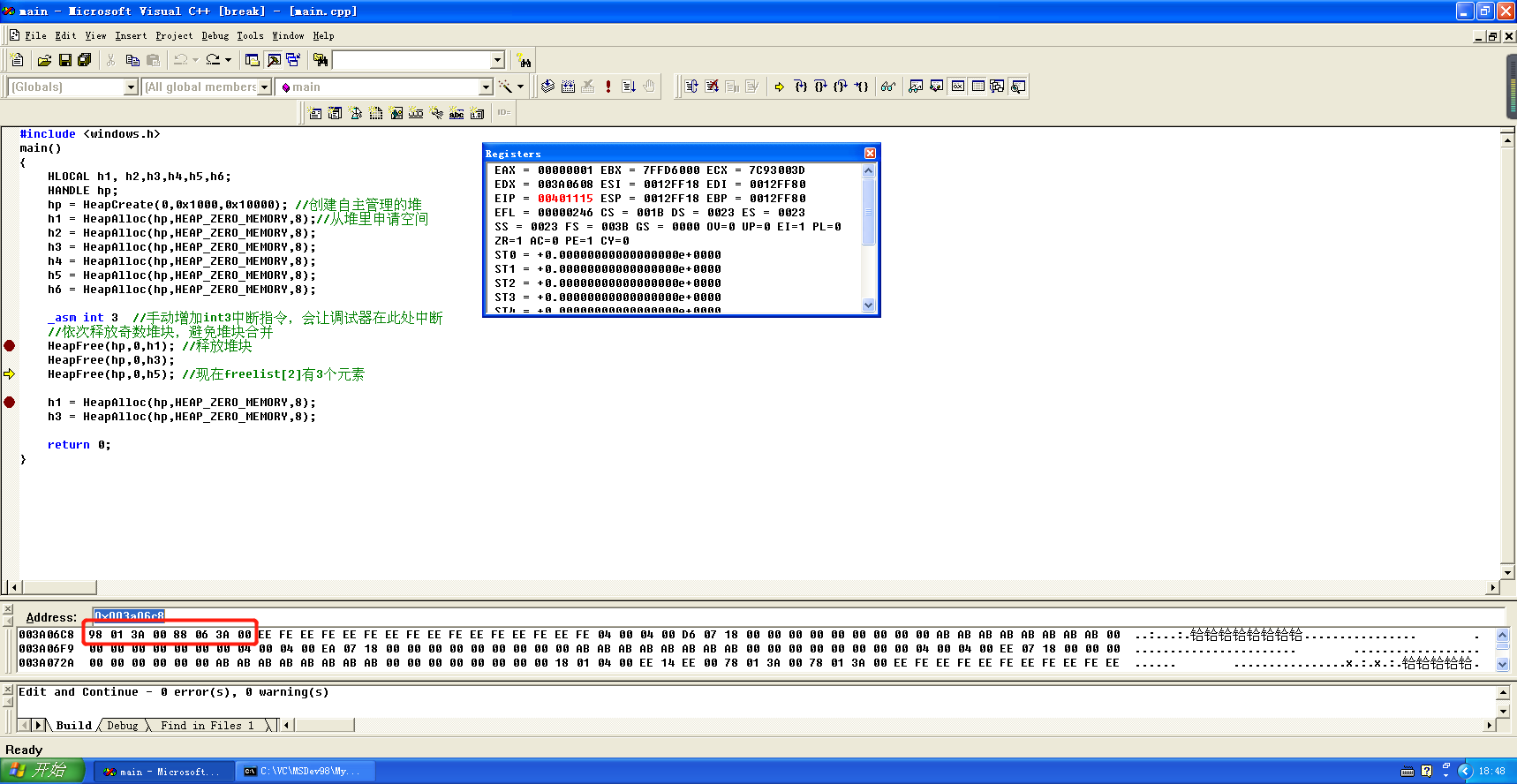




然后释放h3，发现freelist[2]的flink和h1的blink指向了h2地址0x003a06c8，而h2的flink指向h1，blink指向freelink[2]，即h2链入了后边







最后释放h5，效果同上，h5被链入后边



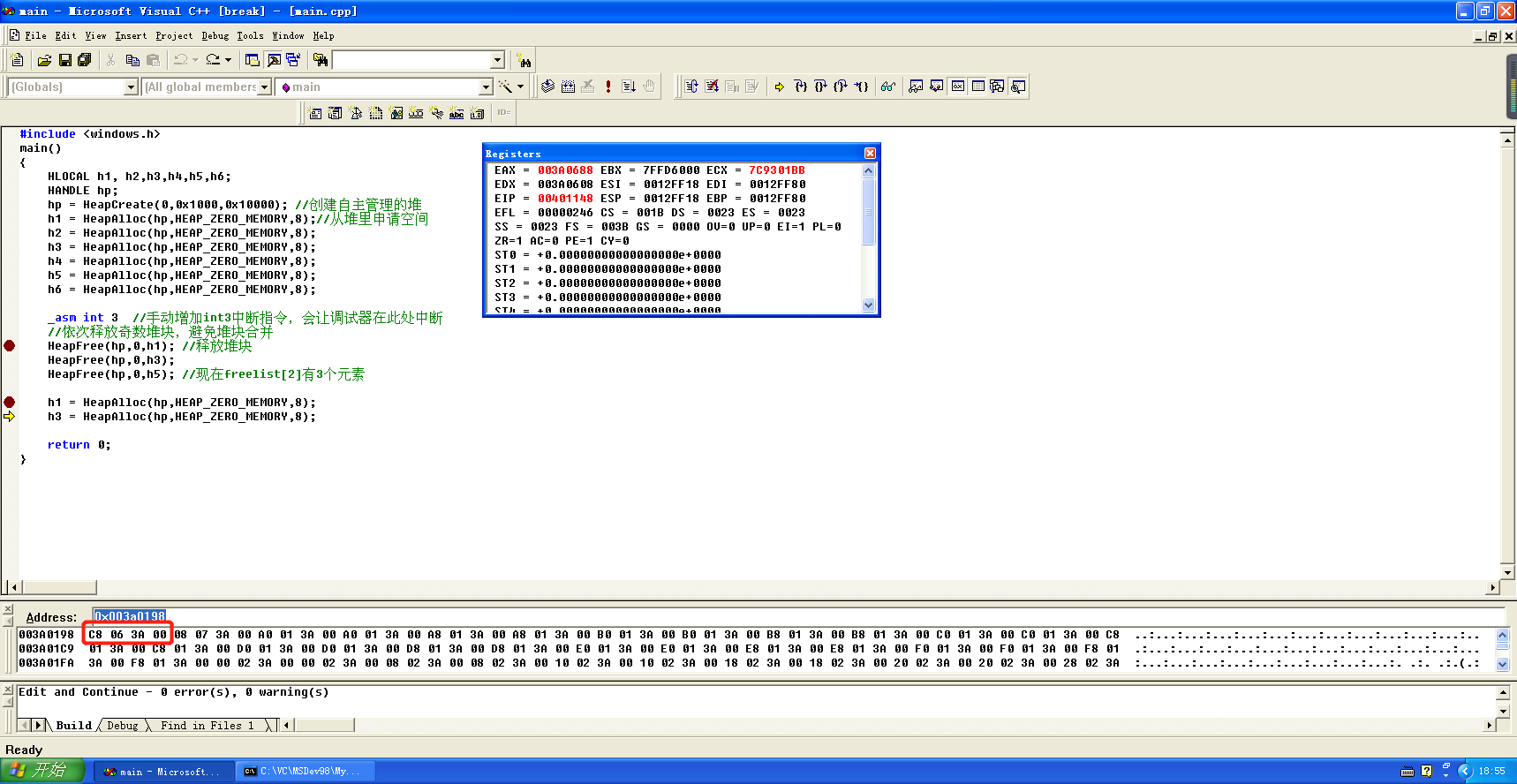


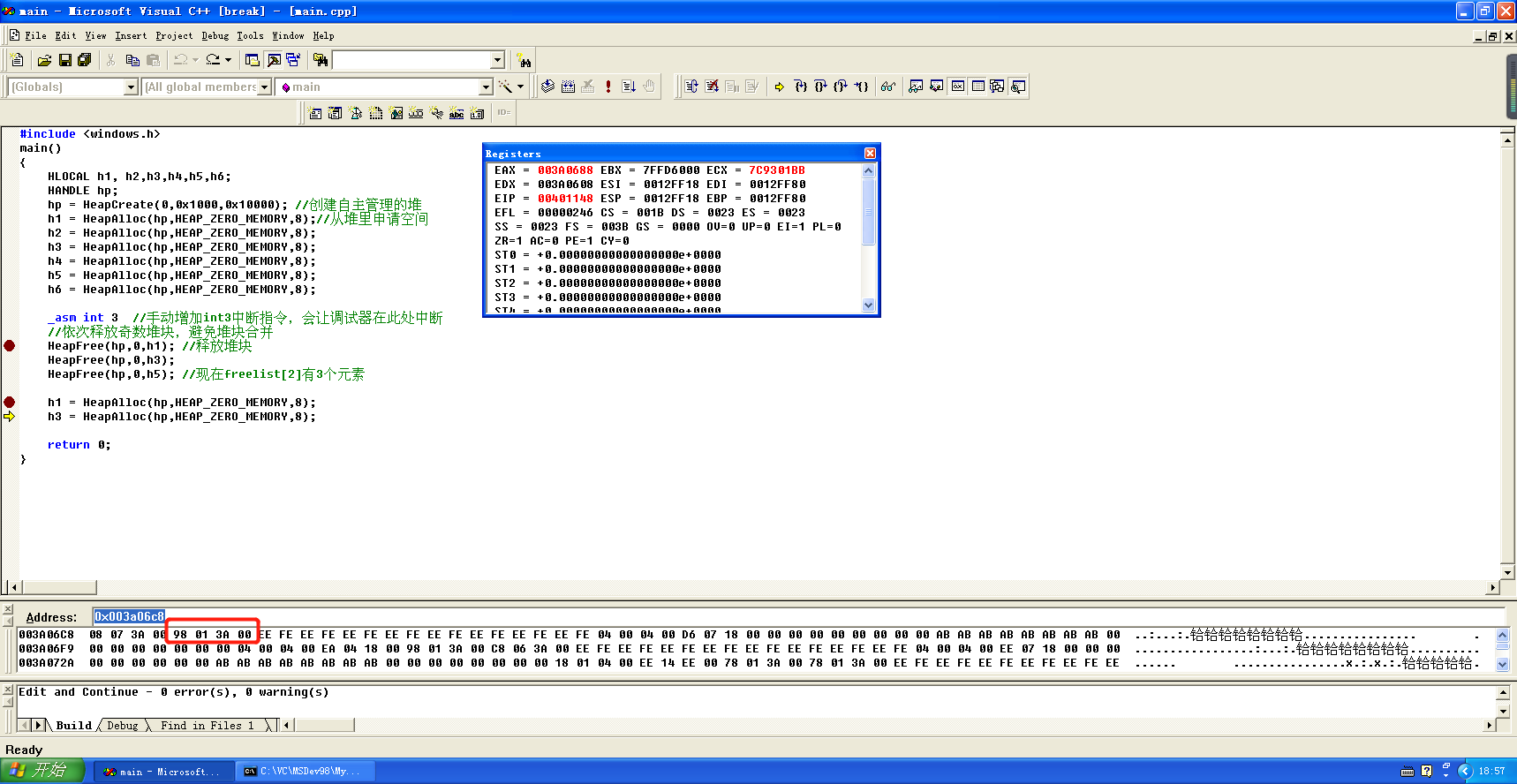




1. Unlink

先h1 Unlink





可以发现freelink[2]的blink指向了h2 0x003a06c8，h2的flink也指向了freelink[2]，即h1被卸下，h1变成了占有态，可以向块内处写入数据。

再回到卸下h1之前，h1的前后项指针为



在卸下h1之前可以将其地址修改，这样的话在卸下h1时，就发生了一次**Dword Shoot攻击。**

实现向任意地址写入任意数据。

再卸下h3也是如此，也是一次Dword Shoot攻击。

**心得体会：**

通过实验，掌握了堆的结构，堆块的存储及索引形式，申请空间和释放空间的操作，堆块的占有态和空闲态的储存形式

了解了Dword Shoot攻击的实现方法

此外，通过本实验，掌握了多个汇编语言的用法