《漏洞利用及渗透测试基础》实验报告

姓名：姜志凯 学号：2011937 班级：信安班

**实验名称：**

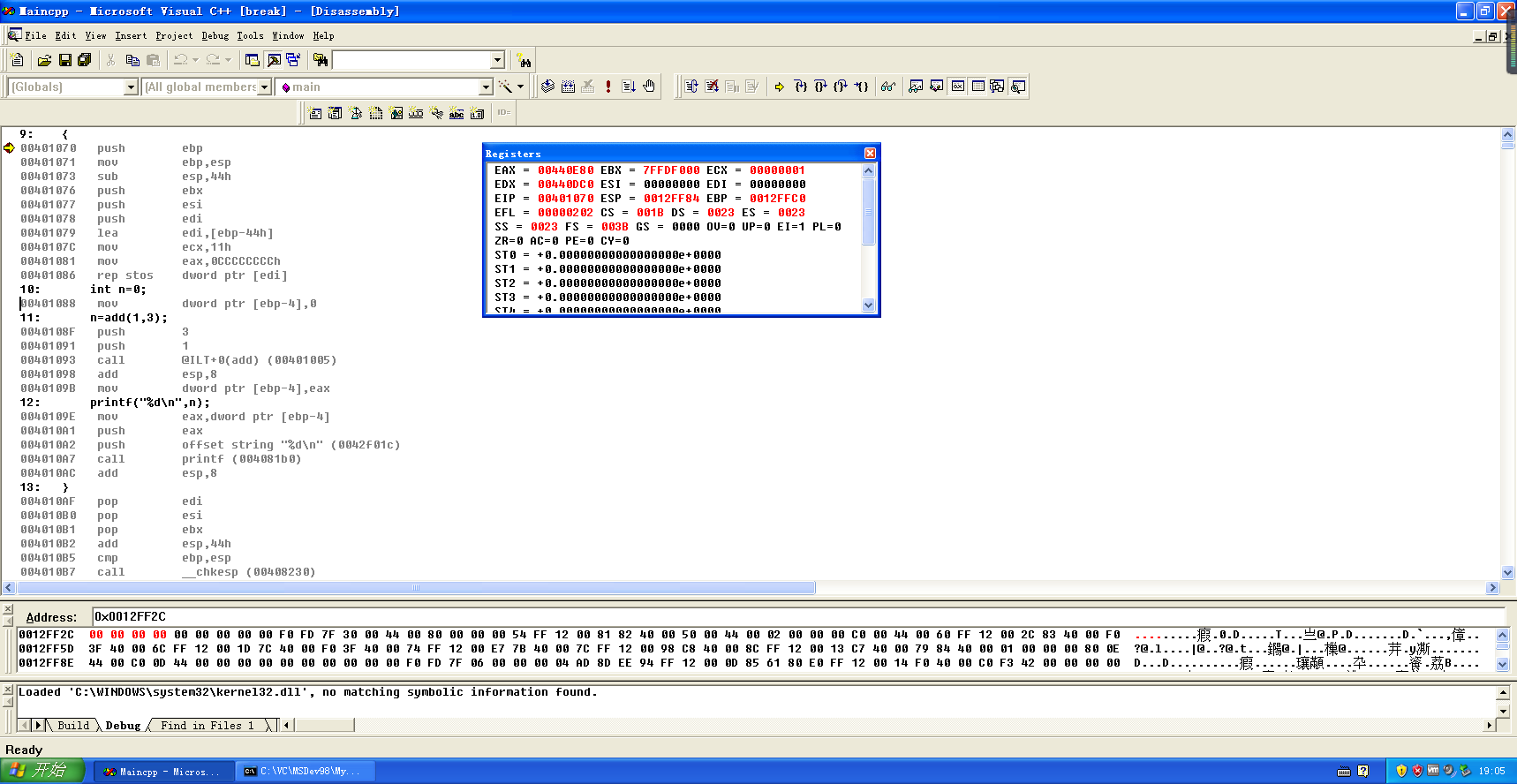
IDE反汇编实验

**实验要求：**

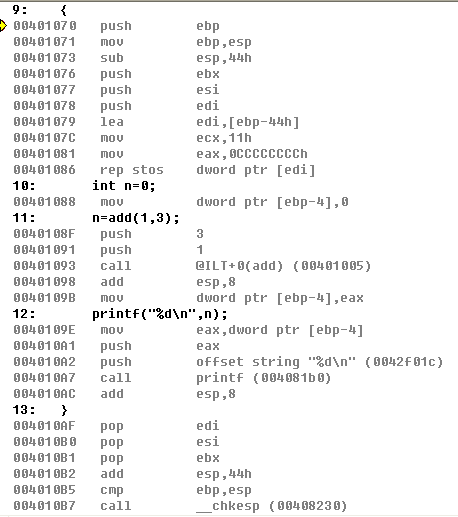
根据第二章示例2-1，在XP环境下进行VC6反汇编调试，熟悉函数调用、栈帧切换、CALL和RET指令等汇编语言实现，将call语句执行过程中的EIP变化、ESP、EBP变化等状态进行记录，解释变化的主要原因。

**实验过程：**

1. 进入VC反汇编

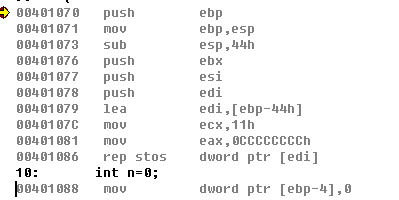


1. 观察add函数调用前后语句



3. add函数内部栈帧切换等关键汇编代码

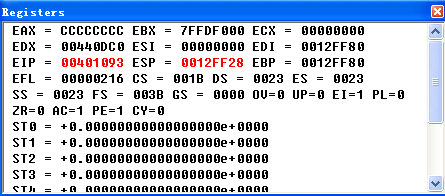
**Add函数前：**

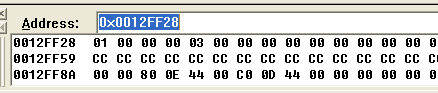


首先为主函数开辟栈帧，将EBP入栈，再将ESP值赋给EBP，再将ESP抬高，然后esi、edi保存现场，再循环将栈帧空间填充，接着通过ebp为n赋值0，接下来将调用函数。

**调用函数前：**

6f86955392f9285ae987d61727f2aa1

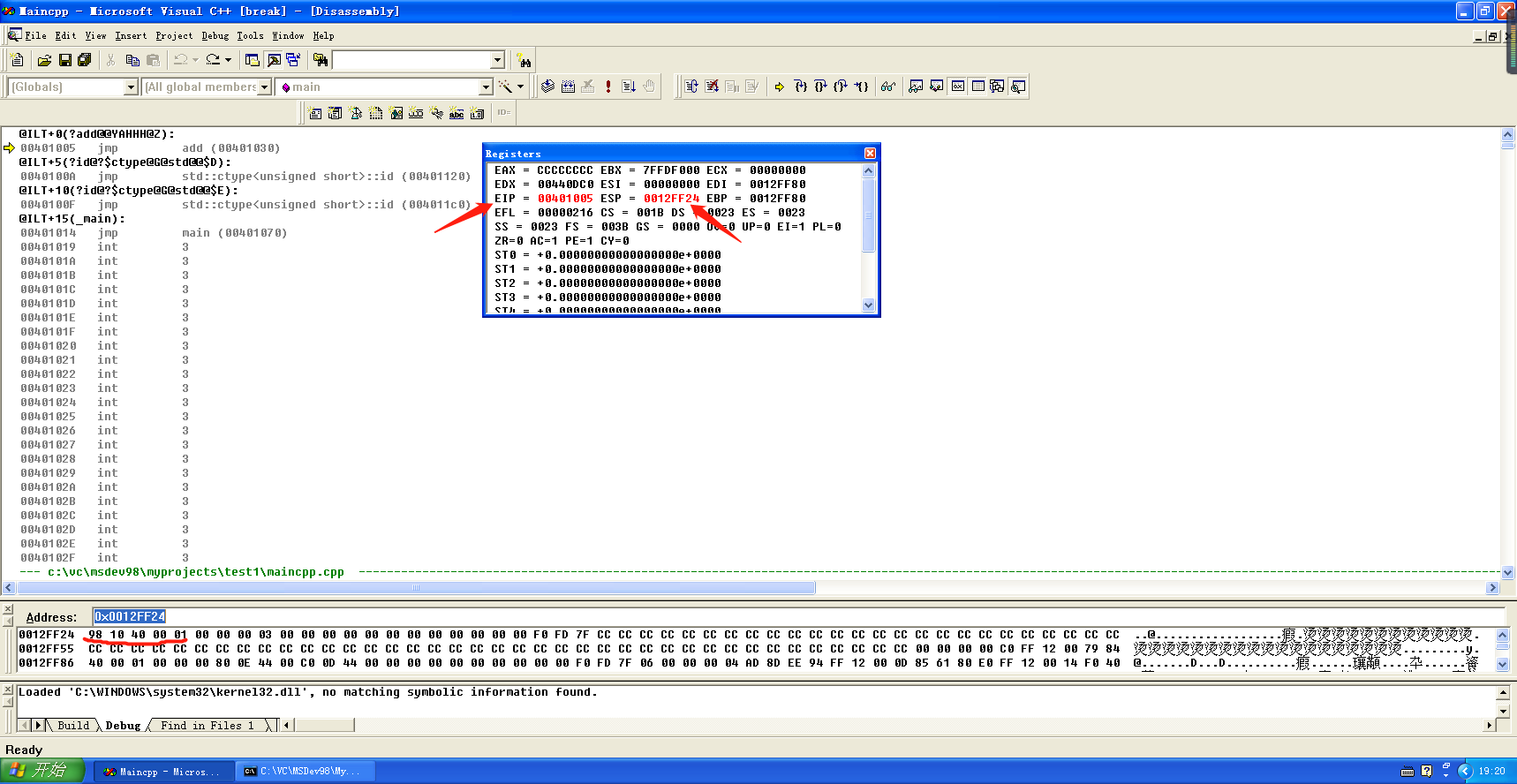




将**参数从右到左入栈**，EIP的值永远是即将执行的指令地址，随着入栈ESP不断变化，3和1依次入栈

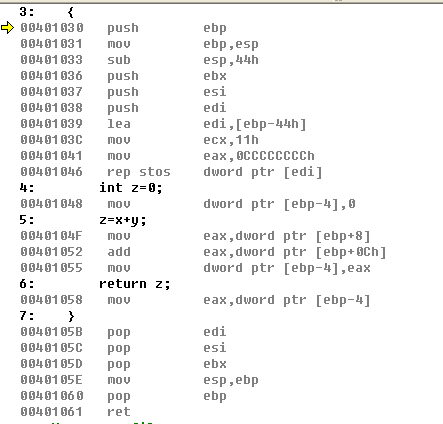
**调用函数时：**

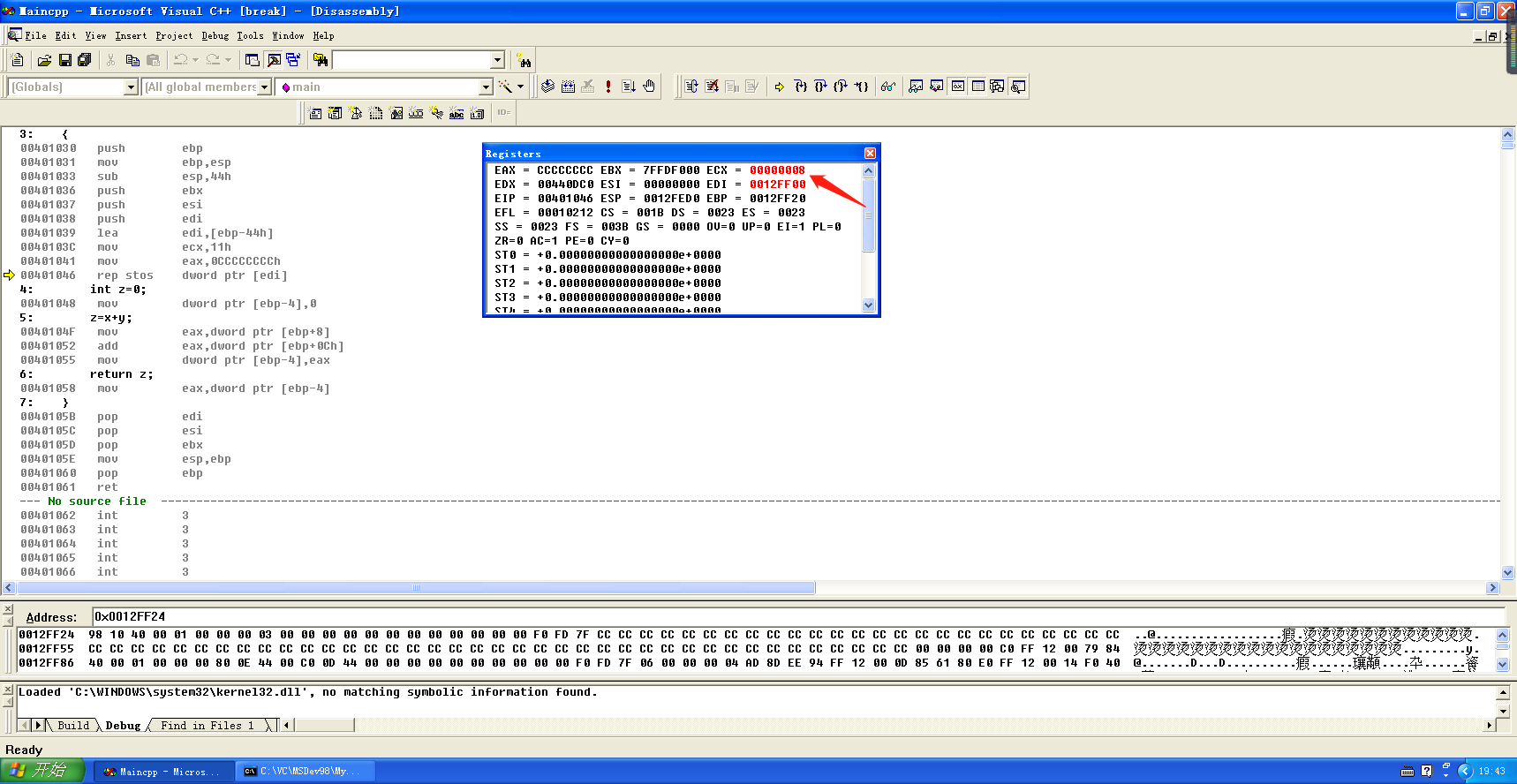
**00411419 call add (411096h)**

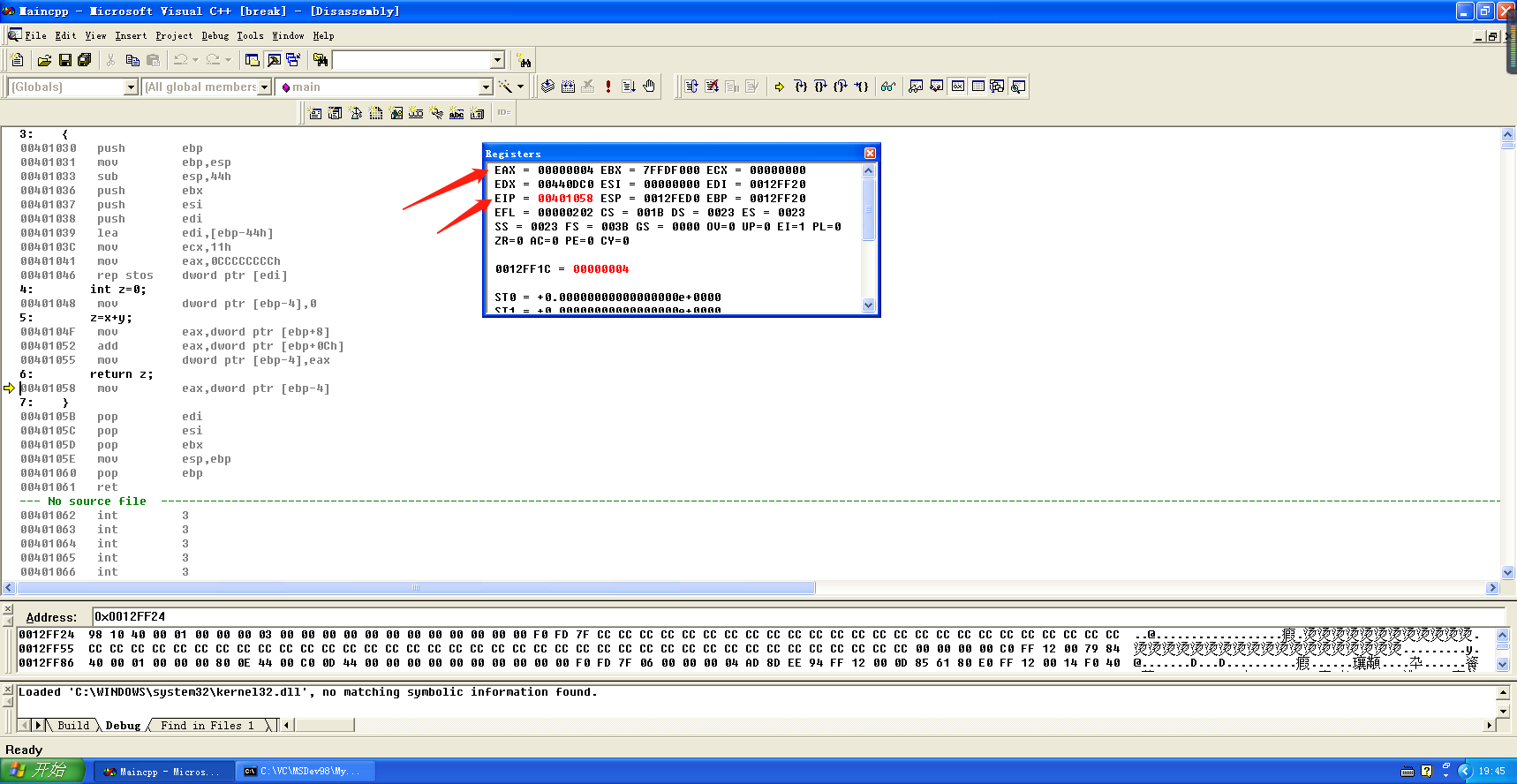


将call指令的下一条地址入栈，即**返回地址入栈**，EIP没有直接变成函数的地址，而是变成跳到函数的地址的指令的地址，执行该跳转指令后，EIP变成函数的入口地址

Add函数的汇编代码：



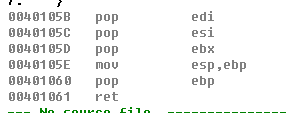




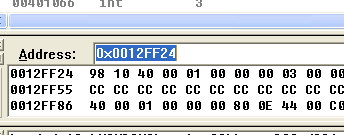
将EBP入栈，ESP的值赋给EBP，再将ESP抬高，为函数栈帧开辟空间。然后EDI、ESI用于保存现场，再循环将栈区都初始化为CCh，rep指令的目的是重复其上面的指令，ECX的值是重复的次数，STOS指令的作用是将eax中的值拷贝到ES:EDI指向的地址

然后依次将z初始化为0，然后将EAX设置为形参x的值，然后把形参y再加到eax上，EAX赋值给z。

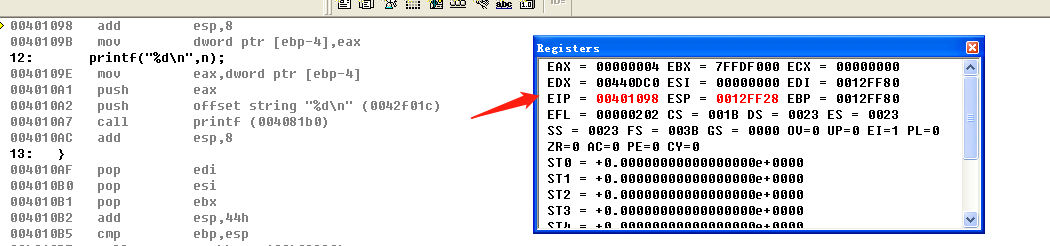
函数返回时将z值存储到EAX中。



接下来恢复寄存器的值，将EBP赋值给ESP，之后pop ebp，ebp恢复



此时ESP指向的是返回地址



EIP又回到了调用函数的地方，函数调用完成。

**心得体会：**

通过实验，掌握了RET指令的用法；

RET指令实际就是执行了Pop EIP

EIP存放的永远是下一条指令的地址

此外，通过本实验，掌握了多个汇编语言的用法