《漏洞利用及渗透测试基础》实验报告

姓名：王亦辉 学号：2310764 班级：计科1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。

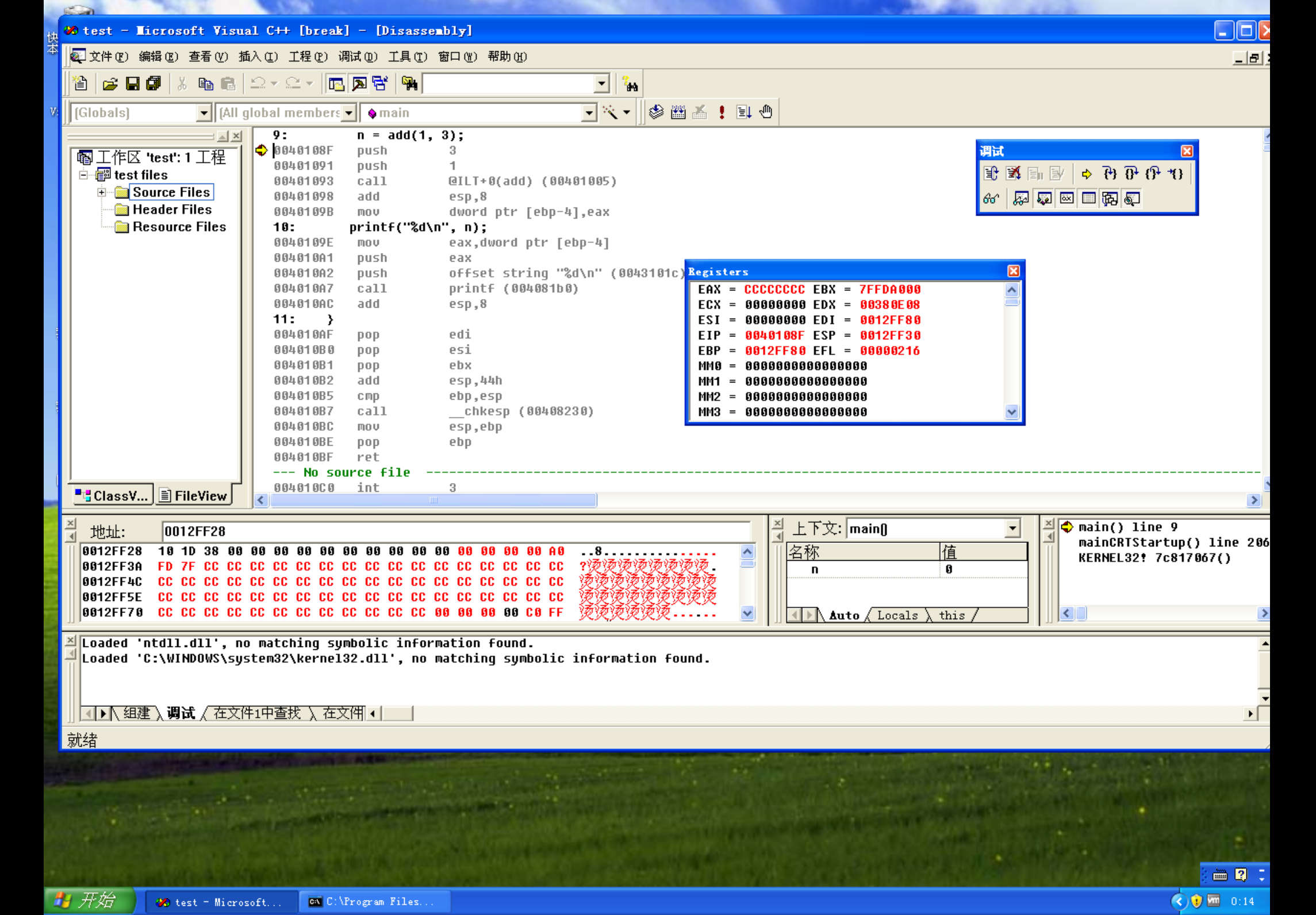
**实验过程：**

1. 进入VC反汇编

1. F9打断点，然后启动调试

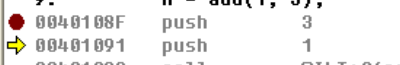
2. 在相应语句处右键，点击go to disassembly

3. 查看汇编代码



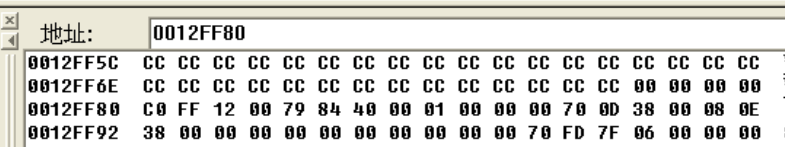
2. 观察add函数调用前后语句

1.调用之前，先将参数按从右往左的顺序入栈。

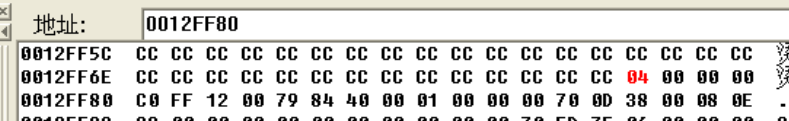


2. 在add函数调用之前，初始化了一个变量n。在add调用后，先把ESP 恢复，然后将eax寄存器的值，存储到前面初始化的变量n里面。从内存里也可以看到，mov dword ptr [ebp-4], eax 之后，内存有变化。



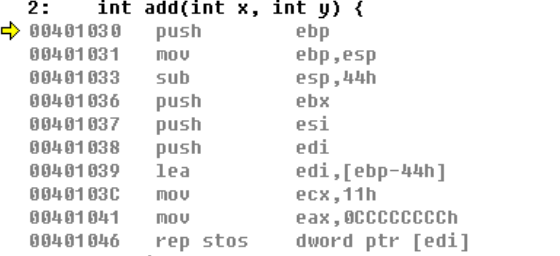




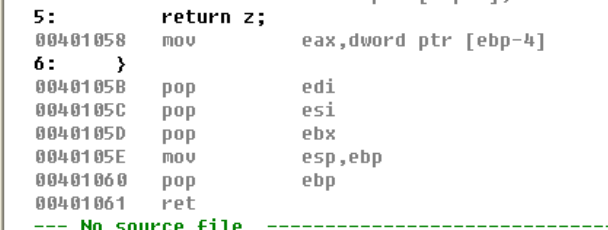


3. add函数内部栈帧切换等关键汇编代码

1. 进入add函数内部之后，首先保存寄存器ebp的值。然后移动ebp喝esp来创建新栈帧。保存了一些寄存器的值，然后初始化栈帧内的内存。



2. 函数返回的时候，是存到eax里了，所以add函数调用完之后，是从eax里拿结果的。返回之后，恢复栈帧，就是取出一些寄存器的值，然后调整esp、ebp。最后返回。



**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

此外，通过本实验，掌握了多个汇编语言的用法

了解了大小端存储，x86是小端存储，所以在内存上读一个4bytes的变量时，地址从低到高，是低字节到高字节。