《漏洞利用及渗透测试基础》实验报告

姓名：申宗尚　学号：2213924　班级：信息安全

实验名称：

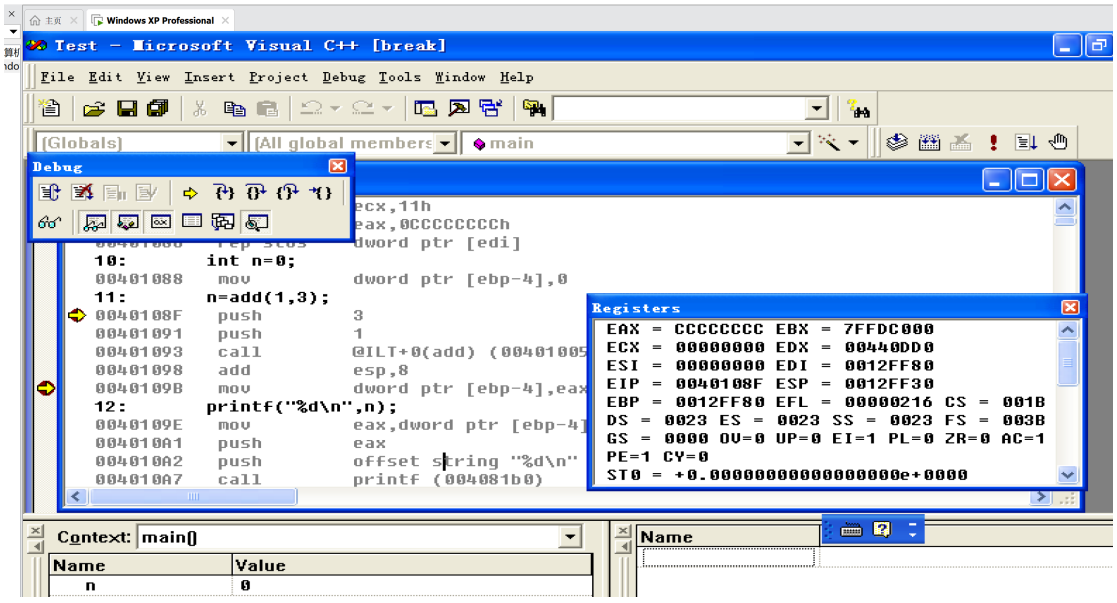
IDE 反汇编实验

实验要求：

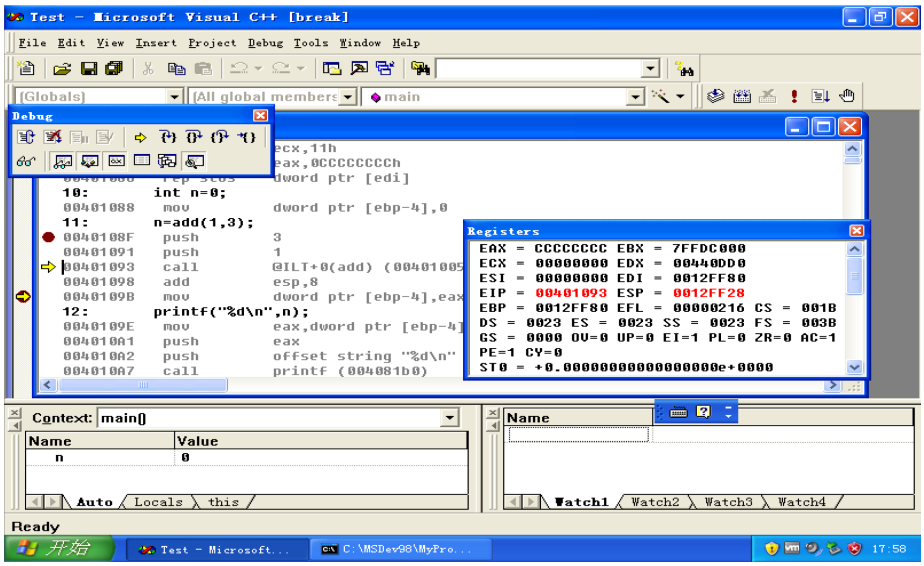
根据第二章示例 2-1，在 XP 环境下进行 VC6 反汇编调试，熟悉函数调用、栈帧切换、CALL 和 RET 指令等汇编语言实现，将 call 语句执行过程中的 EIP 变化、ESP、EBP 变化等状态进行记录，解释变化的主要原因。

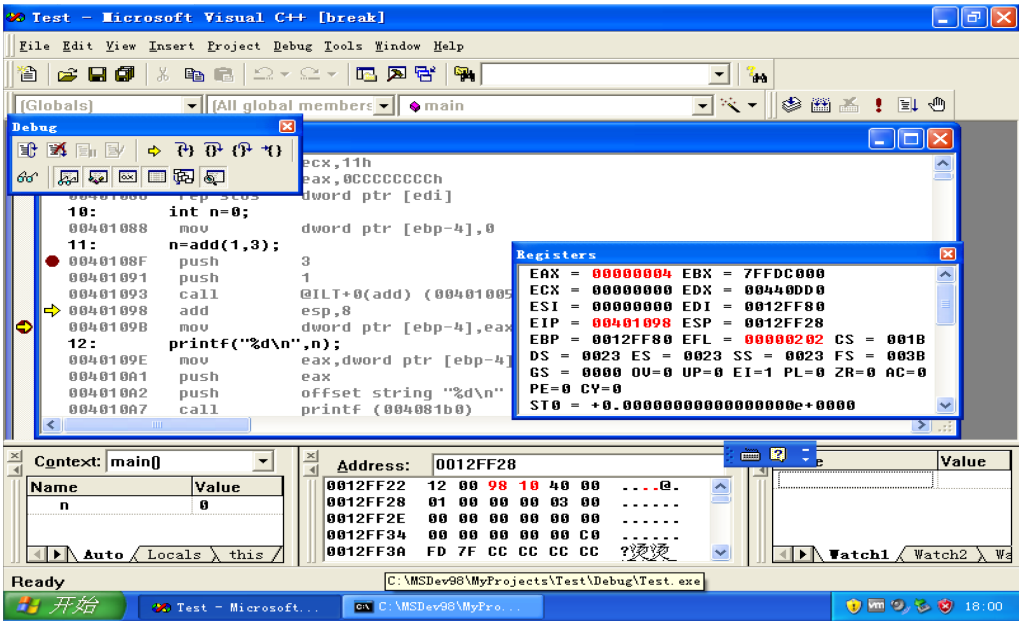
实验过程：

1. 进入 VC 反汇编



2. 观察 add 函数调用前后语句



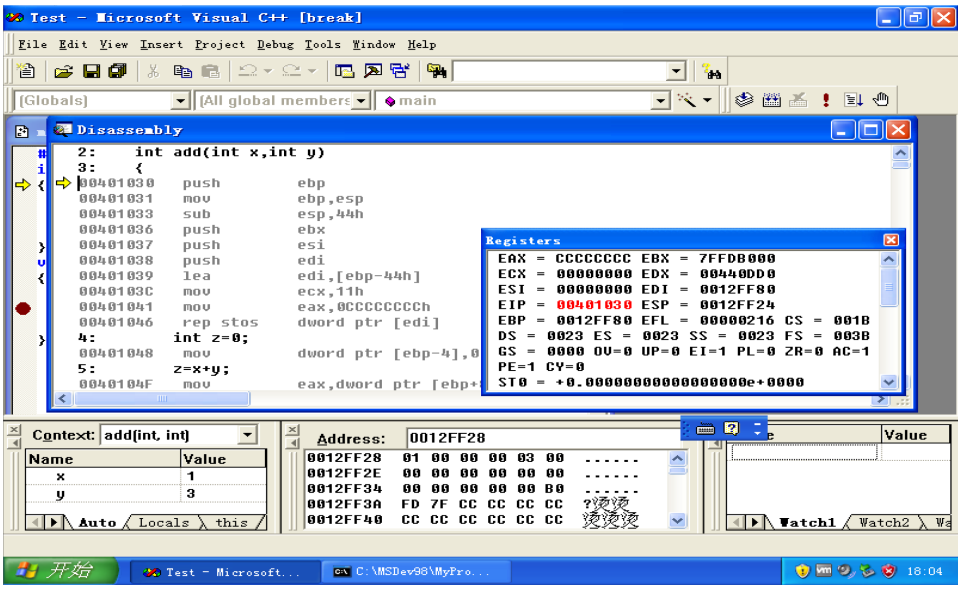


Call 指令执行前，通过从右往左传递参数，将参数 3，1 依次推入栈内，随后进行函数调用。

Call 指令执行后，eax 寄存器变为 00000004，EIP 变为下一步 add 指令的地址。

最后将 eax（即 add 函数的返回结果）储存至变量 n 的存储地址

3. add 函数内部栈帧切换等关键汇编代码



在 add 函数内部，先将 ebp（main 函数栈帧地址）存入栈顶，再将栈顶指针 esp 值赋给 ebp，成为 add 函数新的栈帧底部，再将 esp 进行-44h 的操作，预留出新的栈空间，随后进行栈内部空间的初始化、清零操作。

在进行完运算后，将原本存入的 main 函数栈帧取出，并给 ebp 赋值，从而切换栈，再更改 EIP 位置，使其进行完函数调用后接着运行主函数下一条代码。

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

同时，对于程序中函数的调用底层逻辑、栈空间切换有了更深层的认识，（为何函数调用耗费空间、时间，因为会有很多栈的操作和空间的初始化、清零操作），因此在实际中，除了经常用到的复杂操作，应该以性能为主，尽量写在主函数中。