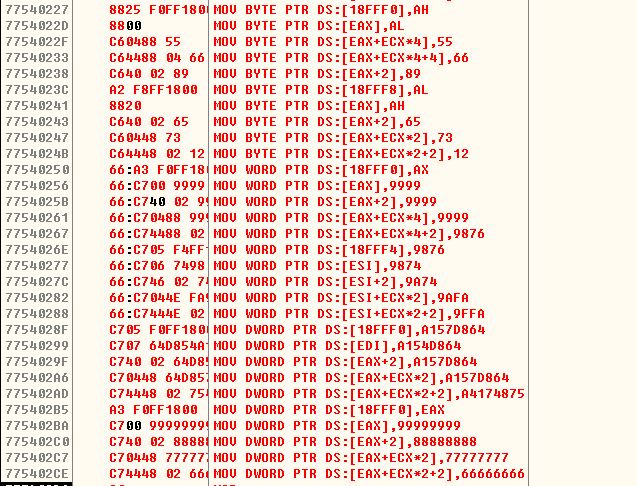
<参考答案见视频源码下载>

分别向内存中存储10个字节、10个字、10个双字 内存地址和内容自己指定,但内存地址要分别使用5种形式来表示。



LEA 指令：

LEA指令和MOV指令一样是移动数据，但是与MOV不同的是，它移动的是“[ ]”里面的内容，例如：LEA EAX,[ECX]中，执行这条指令相当于MOV EAX,ECX。而MOV EAX,[ECX]是把“[ ]”里面的数据当成是内存地址编号，取地址里面的内容。

lea是“load effective address”的缩写，简单的说，lea指令可以用来将一个内存地址直接赋给目的操作数，例如：

lea eax,[ebx+8]就是将ebx+8这个值直接赋给eax，而不是把ebx+8处的内存地址里的数据赋给eax。

而mov指令则恰恰相反，例如：

mov eax,[ebx+8]则是把内存地址为ebx+8处的数据赋给eax。

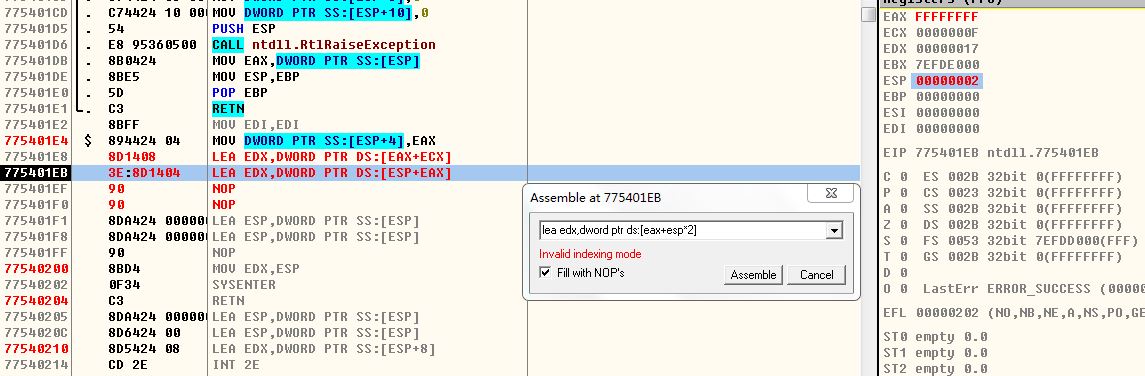
1.已知eax=0x00000008，ecx=0x0000000F，执行下面语句后，求EDX的值

lea edx,dword ptr ds:[eax+ecx]

0x00000017

2.已知eax=0xFFFFFFFF，esp=0x00000002，执行下面语句后，求EDX的值

lea edx,dword ptr ds:[eax+esp\*2]



越界

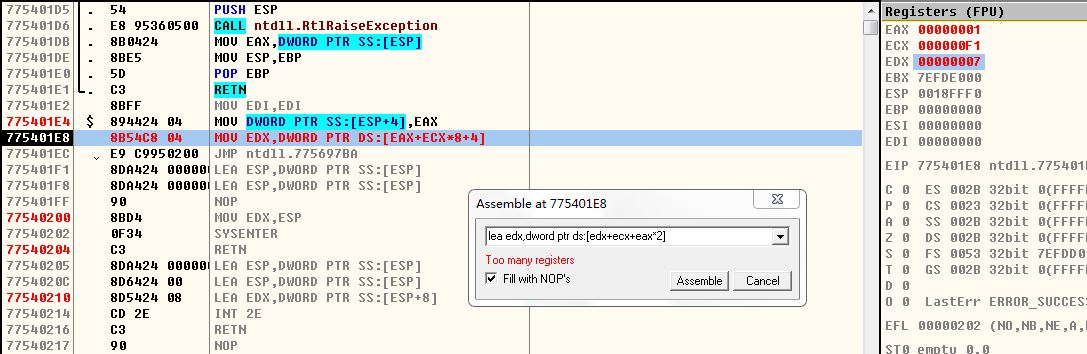
3.已知eax=0x01010101，ecx=0x00000002，执行下面语句后，求EDX的值

mov edx,dword ptr ds:[eax+ecx\*8+4]

内存地址为0x01010115处的数据赋给EDX

4.已知edx=0x00000007，ecx=0x000000F1，eax=0x0000001,执行下面语句后，求EDX的值

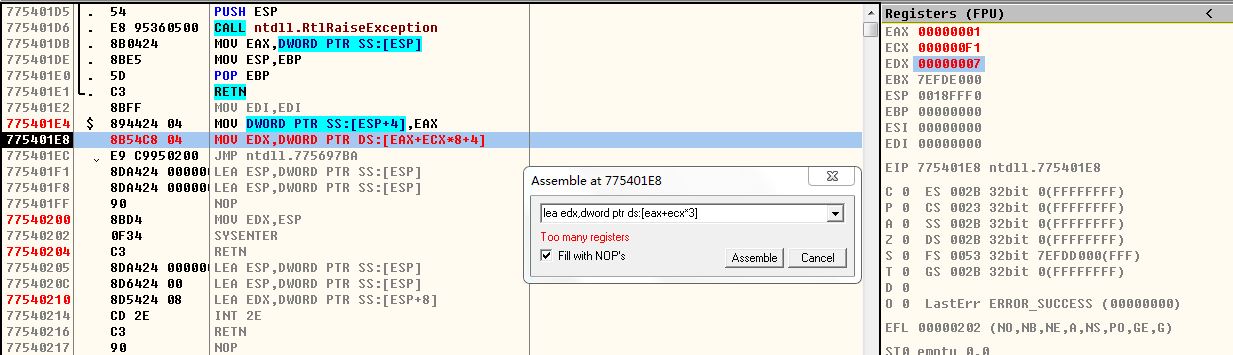
lea edx,dword ptr ds:[edx+ecx+eax\*2]



没有这种储存形式

5.已知eax=0x00000008,ecx=0x00000009执行下面语句后，求EDX的值

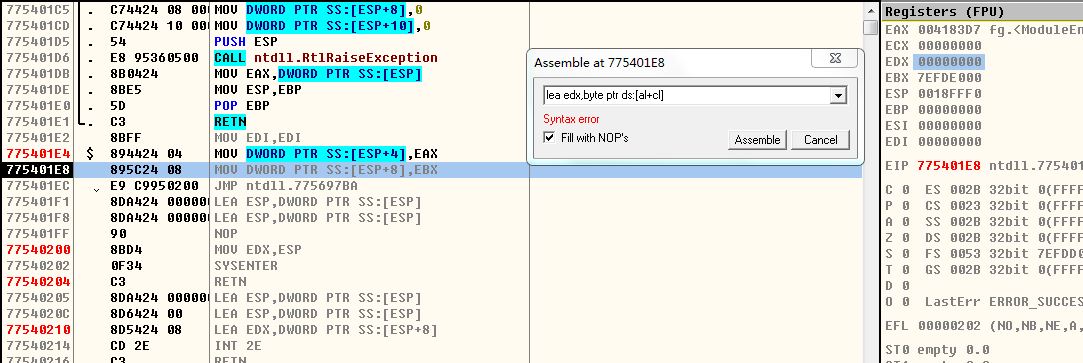
lea edx,dword ptr ds:[eax+ecx\*3]



只能\*1 2 4 8

6.已知eax=0x00000018,ecx=0x00000002执行下面语句后，求EDX的值

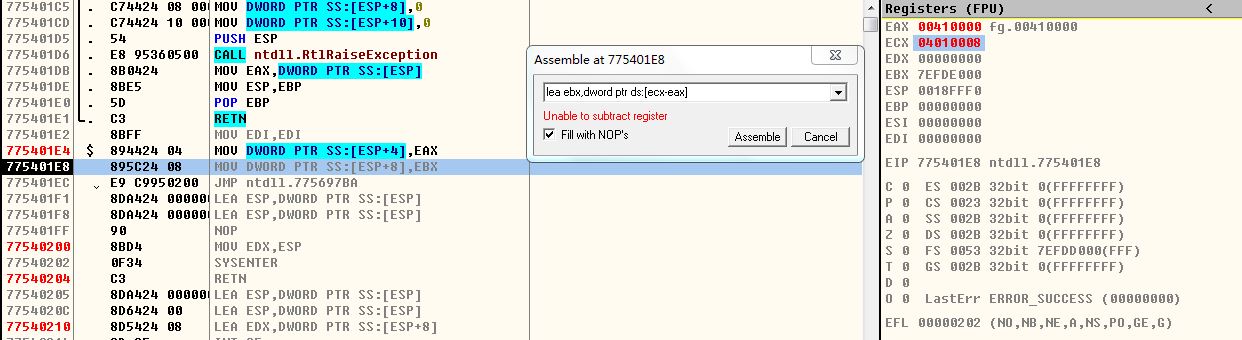
lea edx,byte ptr ds:[al+cl]



语法错误dword 和 byte

7.已知eax=0x00410000,ecx=0x004010008,执行下面语句后，求EDX的值

lea ebx,dword ptr ds:[ecx-eax]



无法减去寄存器

lea edx,dword ptr ds:[ebx]

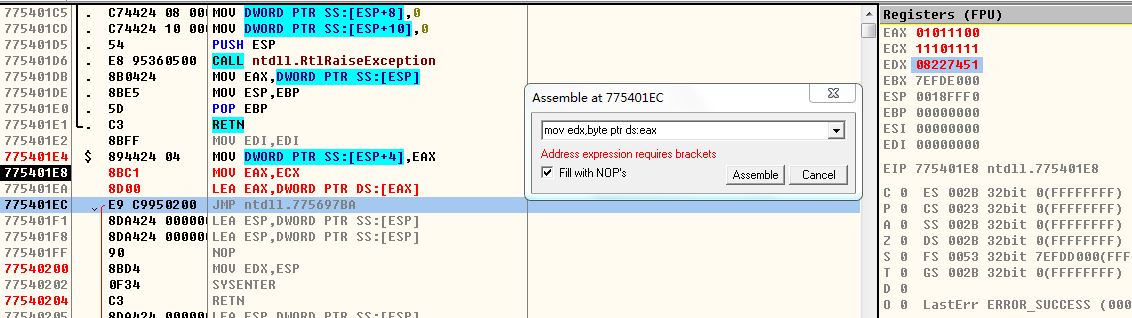
edx值等于ebx

8.已知eax=01011100 ecx=11101111，edx=0x08227451执行下面语句后，求EDX的值

mov eax,ecx

lea eax,dword ptr ds:[eax]

mov edx,byte ptr ds:eax



地址表达式没带括号,byte不能到dword

8.已知eax=01011100 ecx=11101111，edx=11101110,执行下面语句后，求EDX的值

lea edx,word ptr ds:[eax+ecx\*2]

EDX的值 = 23213322

9.已知eax=01011100 ecx=11101111,执行下面语句后，求EDX的值

lea edx,dword ptr ds:[eax+ecx\*2]

EDX的值 = 23213322