**IDA Pro快速指南**

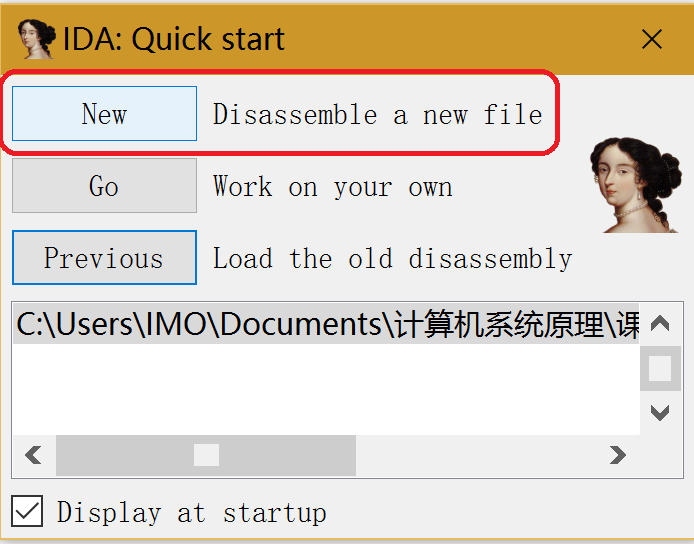
IDA Pro是目前最为专业的反汇编器，主要用于对二进制代码的反汇编、静态分析、动态调试等，是分析病毒、木马、恶意软件、破解、逆向工程必备利器。

1. 安装

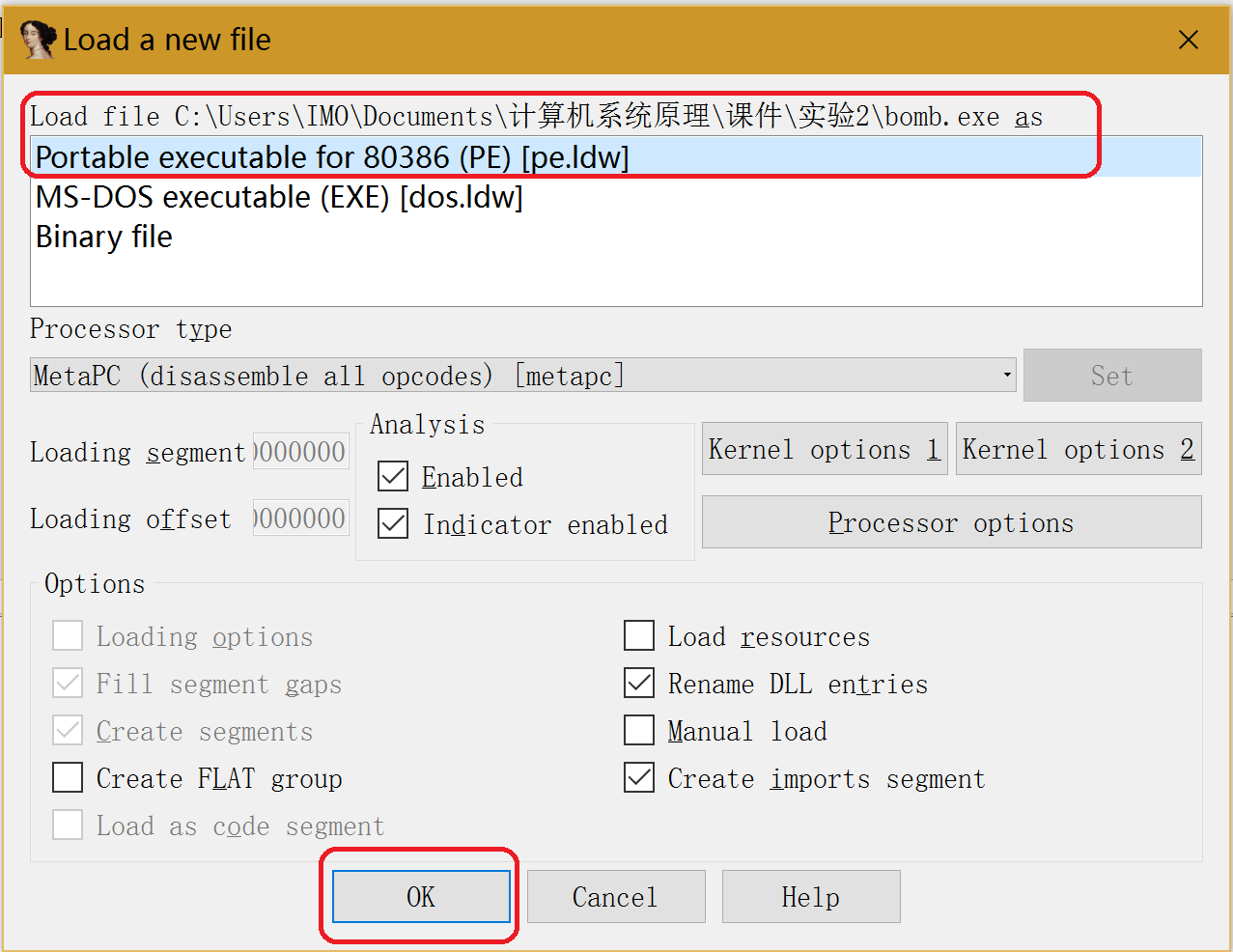
IDA Pro安装，直接一路next即可。

IDA Pro 6.8安装密码

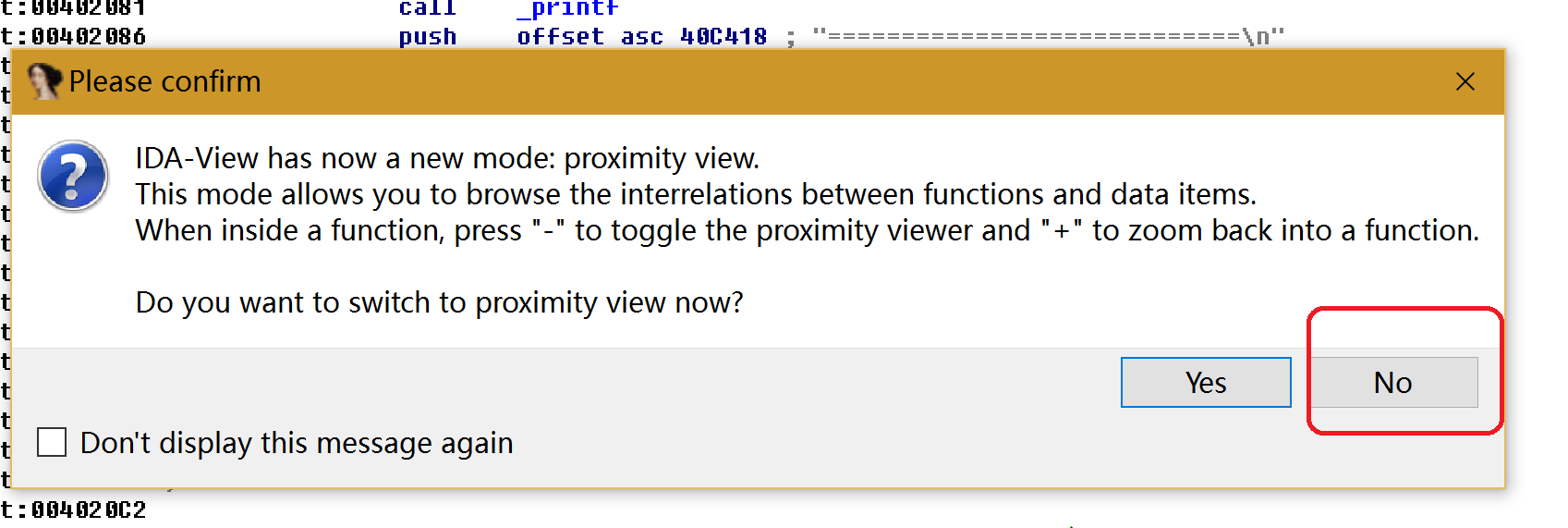
1. 快速使用
2. 启动IDA Pro，选择New；



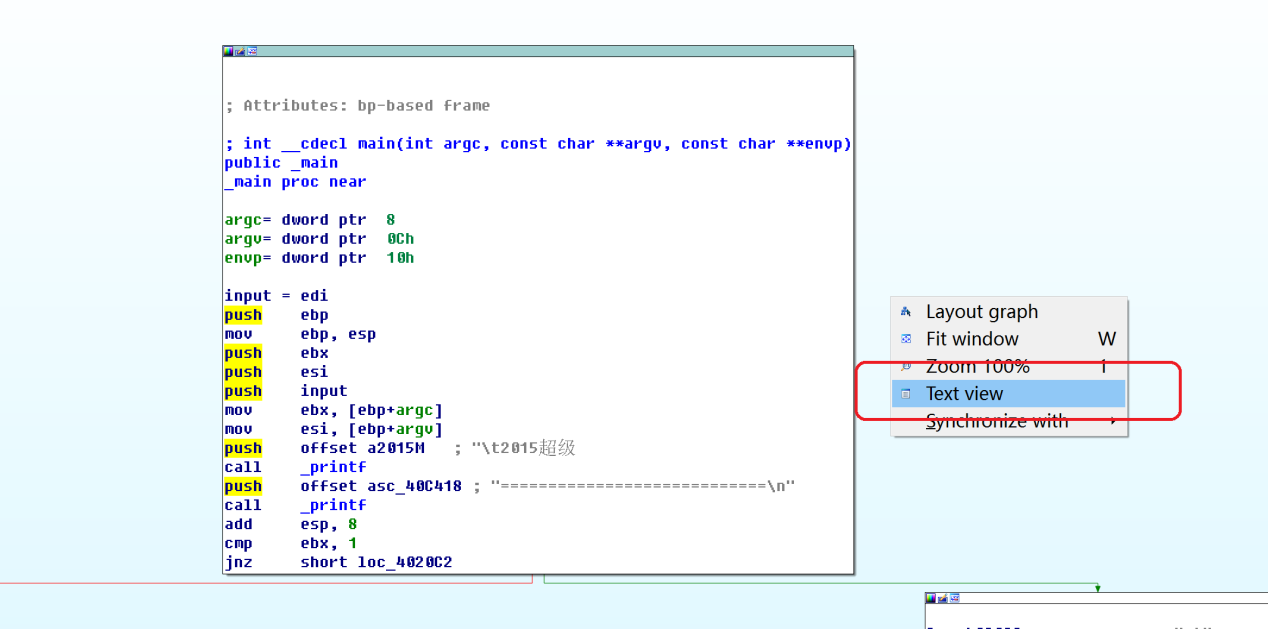
1. 选中待分析的二进制文件（bomb.exe），IDA Pro会自动识别文件类型，为Portable executable for 80386，点击OK



1. IDA Pro会自动开始加载文件并开始分析，然后提示。选择No，不需要相关性视图（后期可以尝试转换到相关性视图，比较容易分析函数之间的关系）

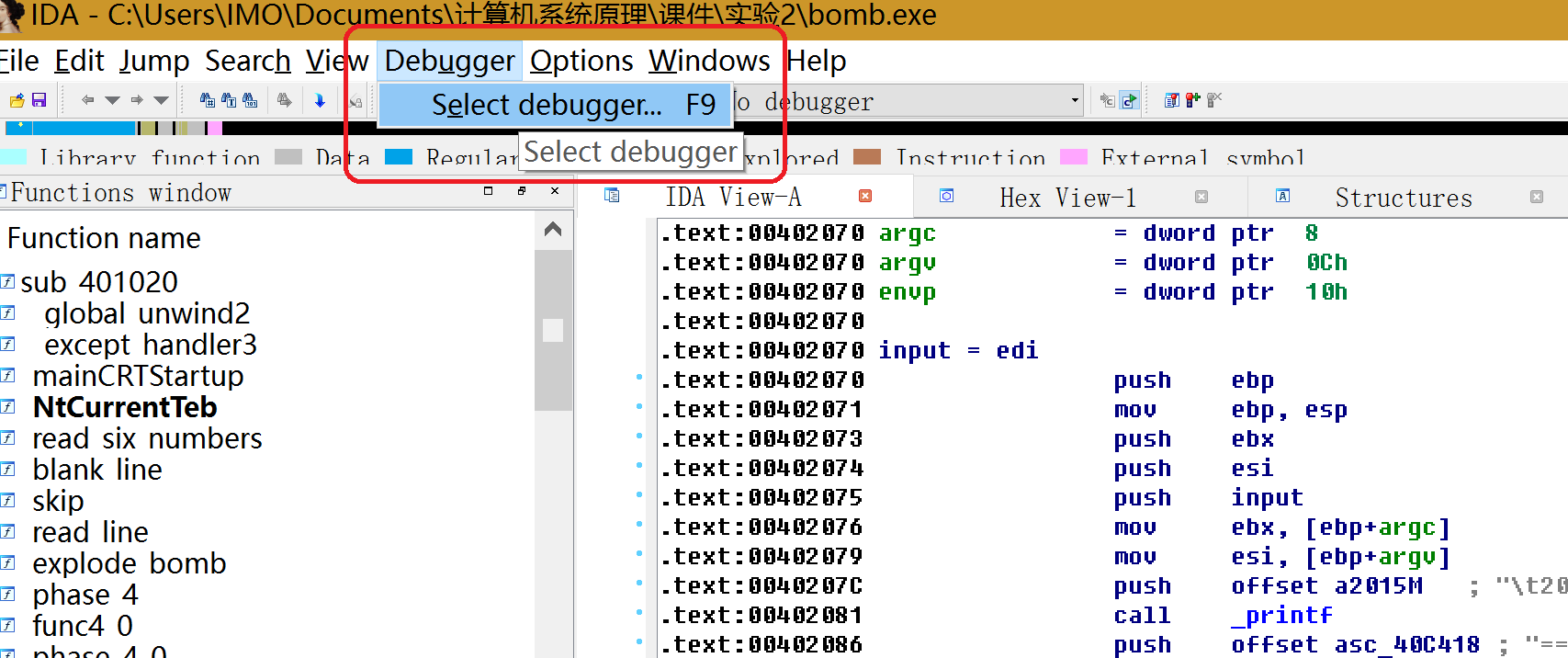


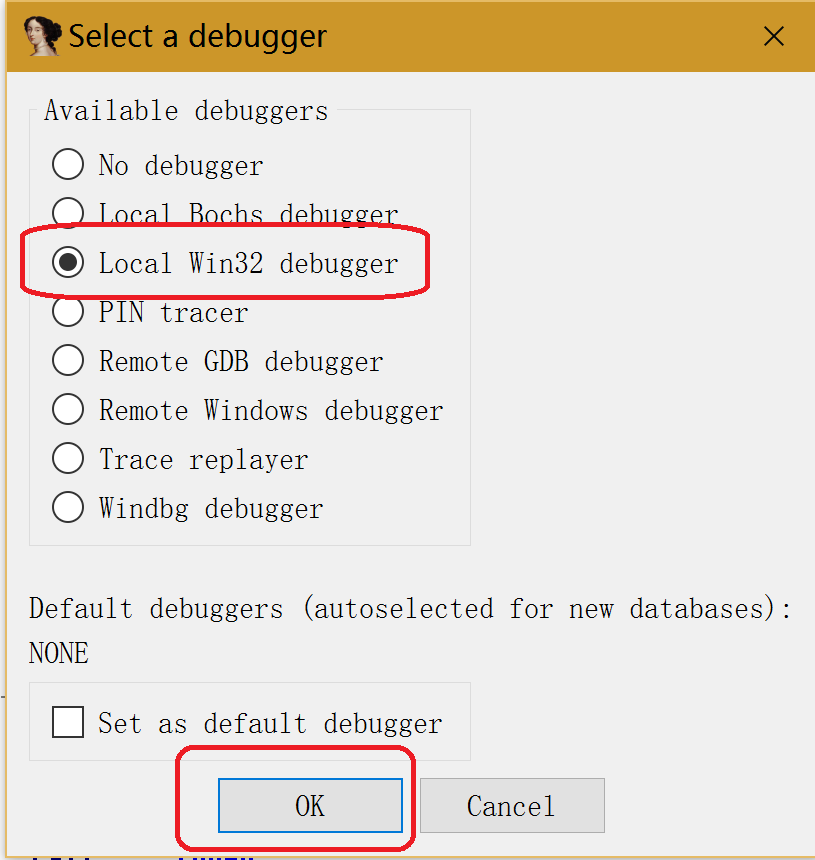
1. 鼠标点击空白出，右键，选择Text View，切换到文本模式，可以以比较线性的方式观察程序。



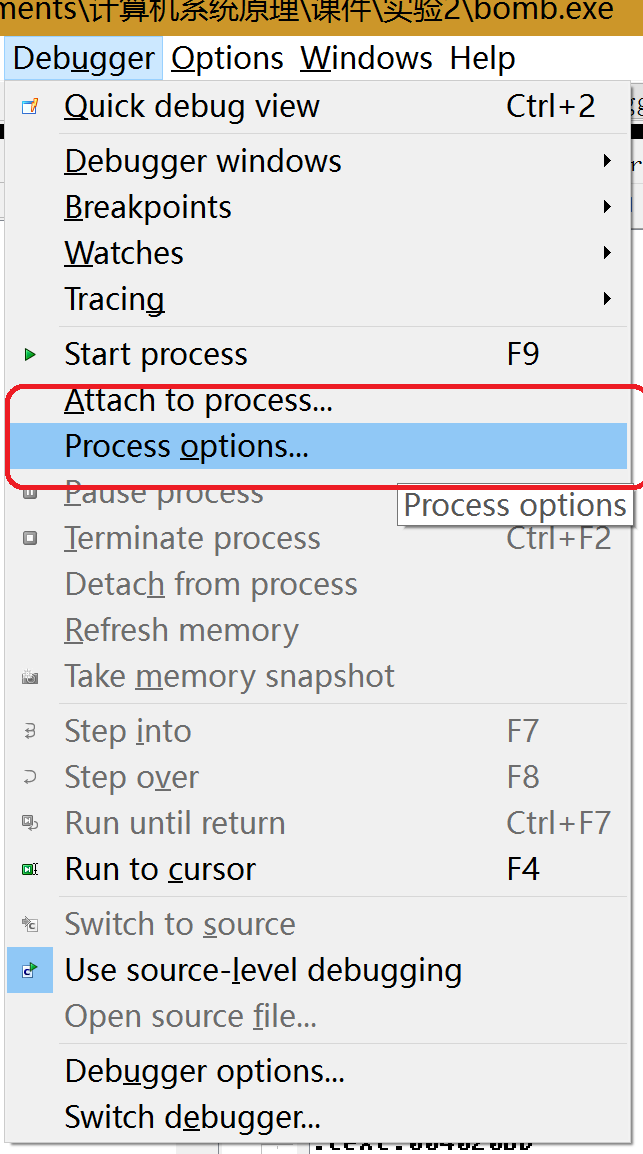


1. 此时，IDA Pro会自动定位到程序运行的入口点main函数上。后面将进行调试。
2. 调试前，需要设定调试器。从Debugger菜单中，点击Select debugger，然后选择Local Win32 debugger。OK

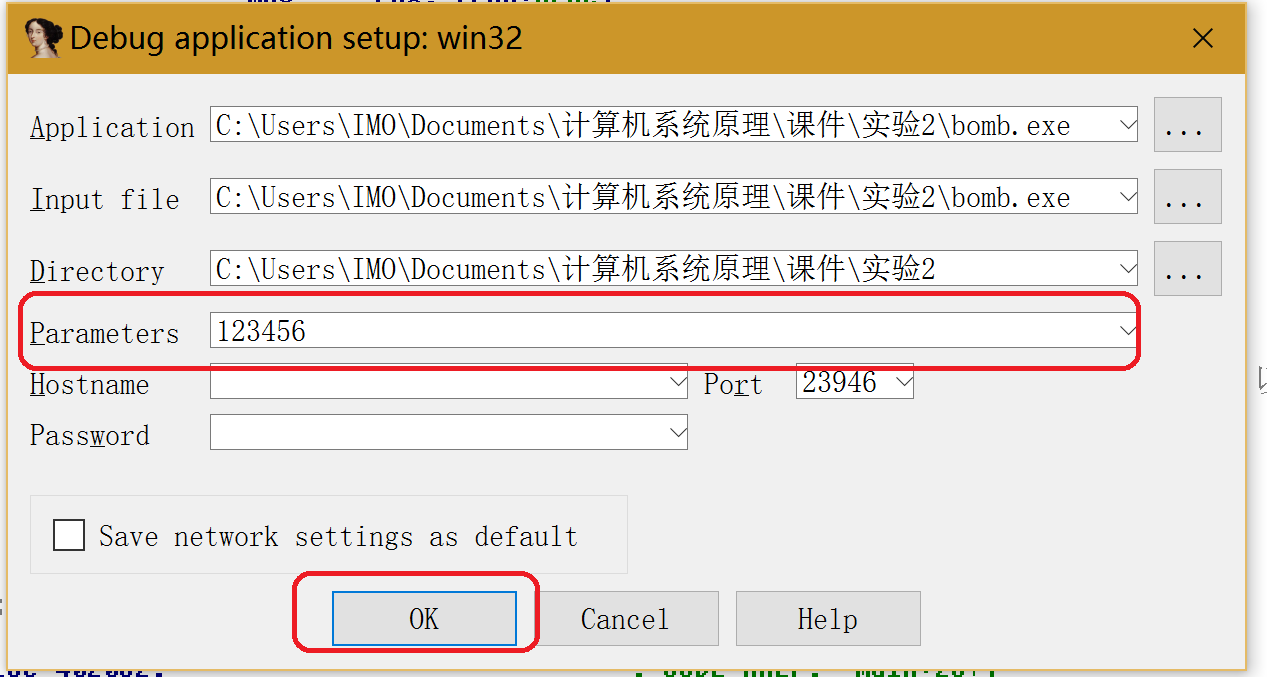




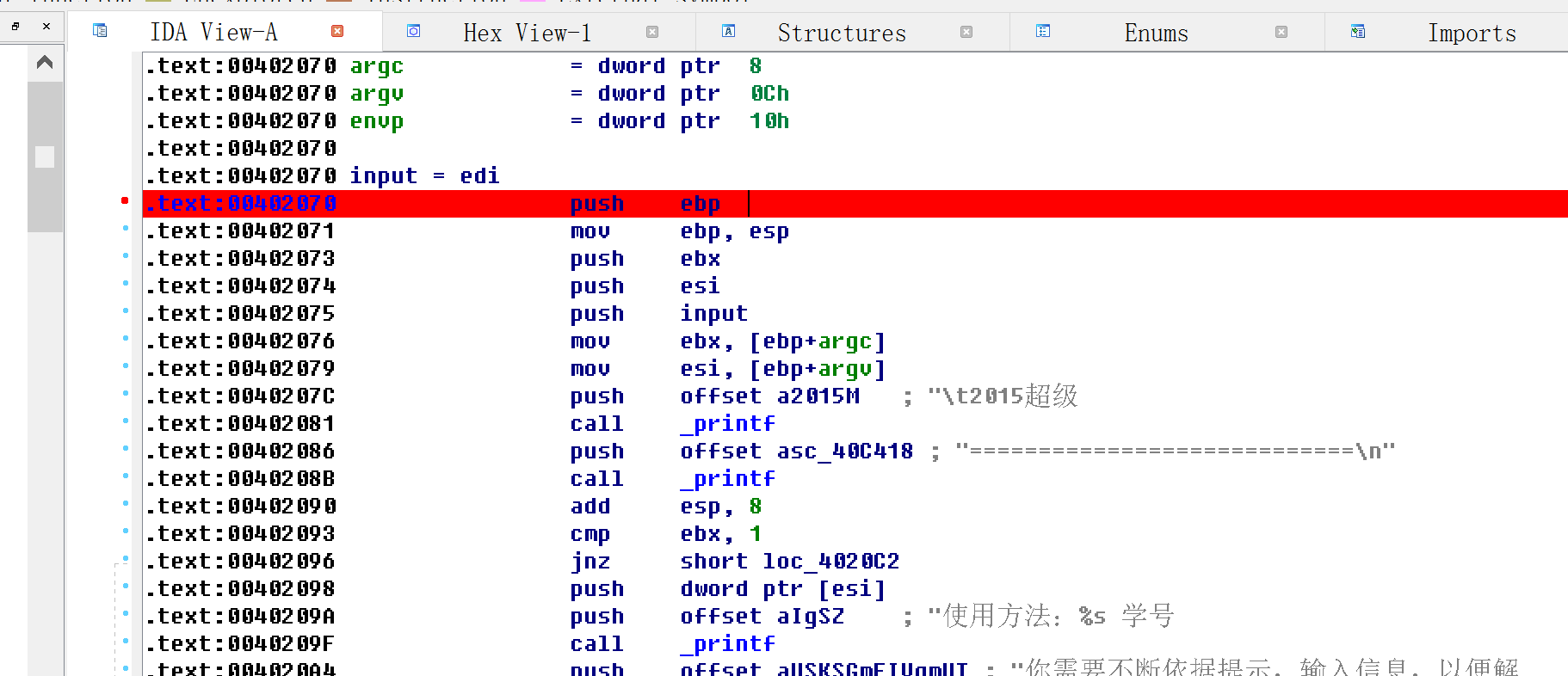
1. 由于我们的程序（bomb.exe）运行需要一个6位数字的学号参数，所以先进行设定。在Debugger菜单中，选择Process Option。



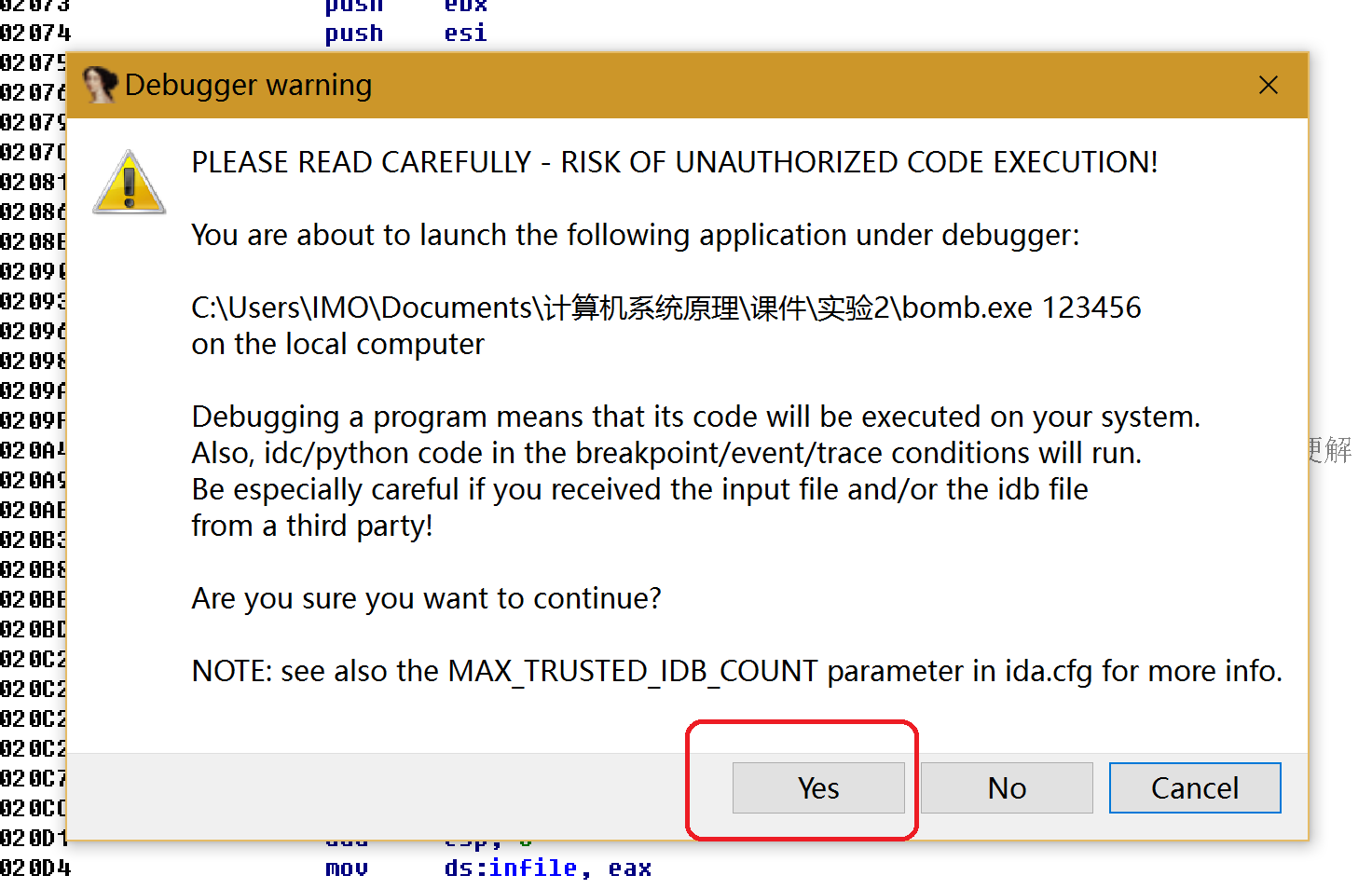
1. 在Parameter栏中，填入你的学号后6位数字（比如此处是123456），然后OK



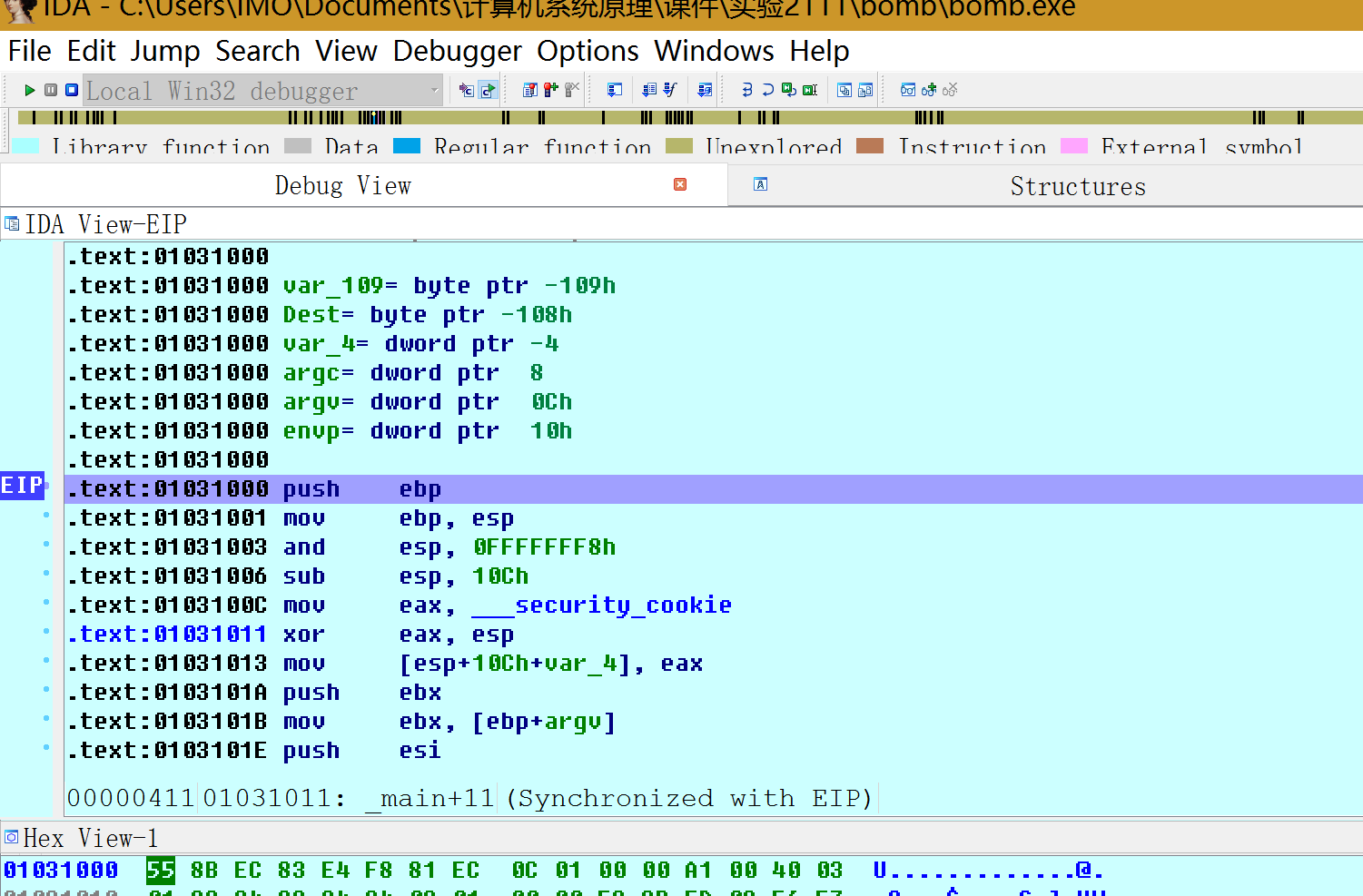
1. 此时，参数就设置好了，后面就可以开始调试程序了。调试器一个很强大的功能，就是可以设置断点，然后单步执行程序等等。。。。。我们先要设置一个断点，让程序运行到断点，停下来。首先在main函数处设置断点。鼠标点击main函数中的一行代码，然后按F2，就会在光标行，加入一个断点。此行变成红色。



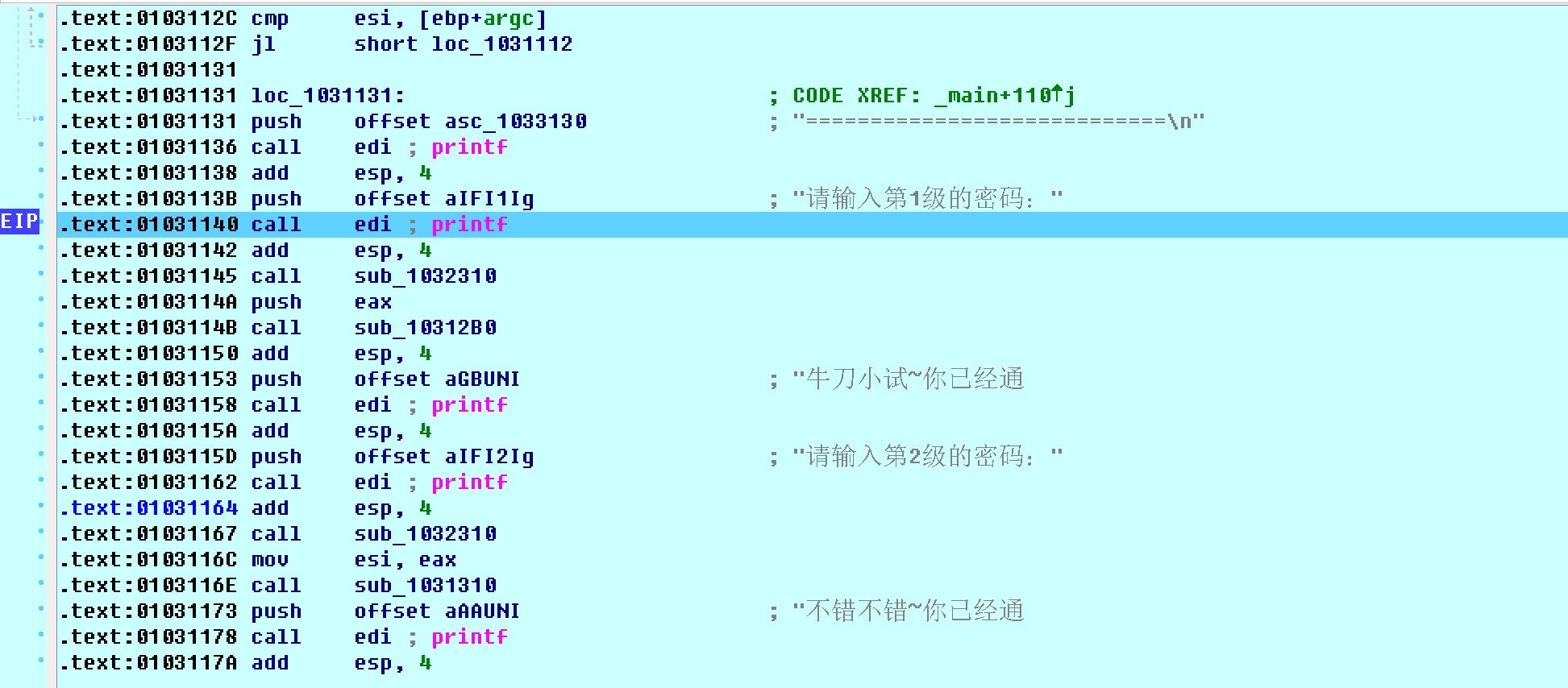
1. 然后我们可以按F9运行程序。此时，会警告被调试的程序将会被执行，如果是病毒木马，可能有中毒的危险，巴拉巴拉……不理他，YES



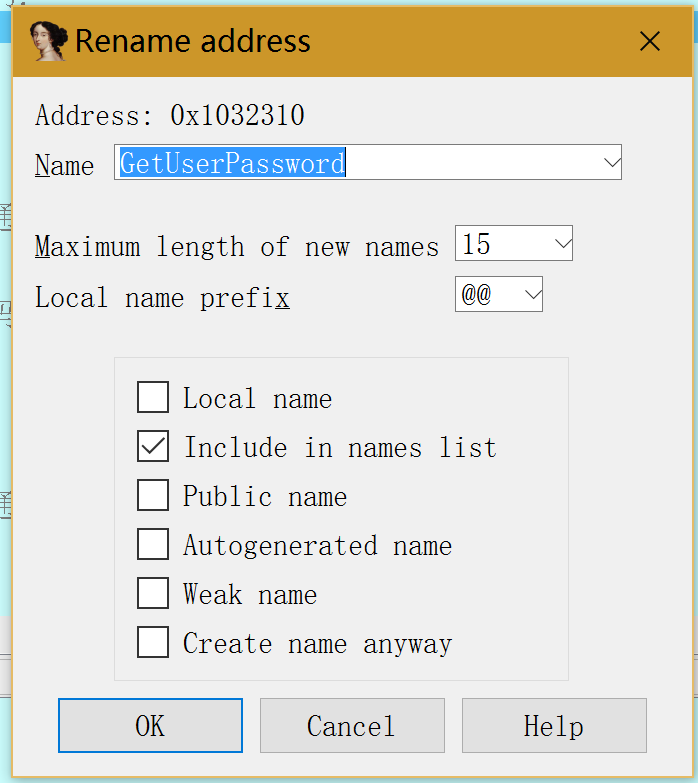
1. 好了，程序执行了。当然他很快会停在我们设置的断点地方。注意到EIP此时指向的正是我们设置了断点的地方。

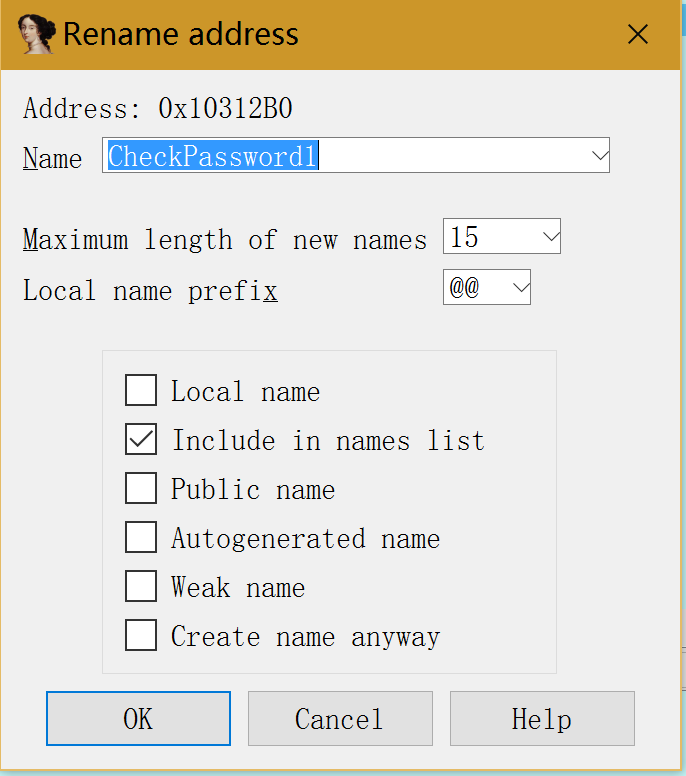


1. 然后我们可以单步执行程序，注意观察EIP光标的变化情况。单步执行有两种，一种称为Step Into（F7），一种称为Step Over（F8），它们都是单步执行，区别在于，Step Into碰到call指令，会“进去”执行被调用的程序。而Step Over碰到call指令，就会快速执行完被调用的程序，而不会停到被调用的程序。
2. 此时按F8，直到"请输入第1级的密码："处：

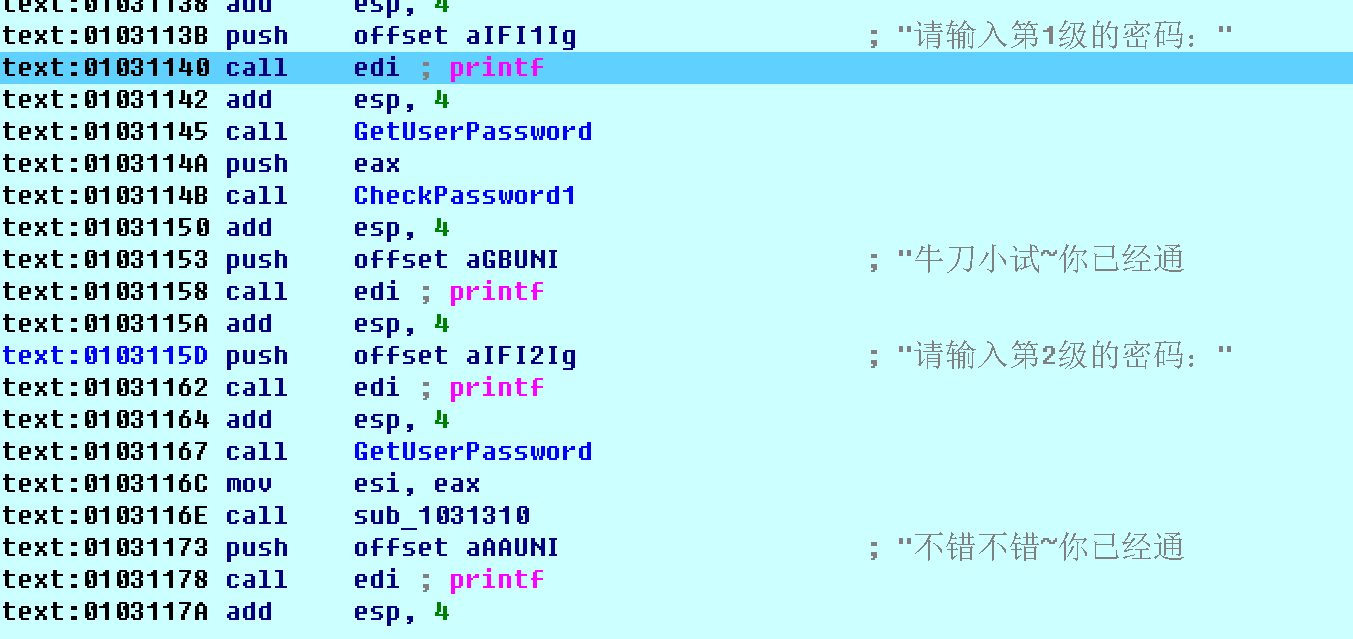


1. 注意到，当前停在call printf语句，输出"请输入第1级的密码："这句话。下面有两个函数，一个call sub\_1032310，显然是读取用户的输入密码。而第二个call sub\_10312B0显然是检测用户输入的密码对不对。给它们改个名字吧。鼠标右键点击call sub\_1032310，rename，输入GetUserPassword；鼠标右键点击call sub\_10312B0，rename，输入CheckPassword1

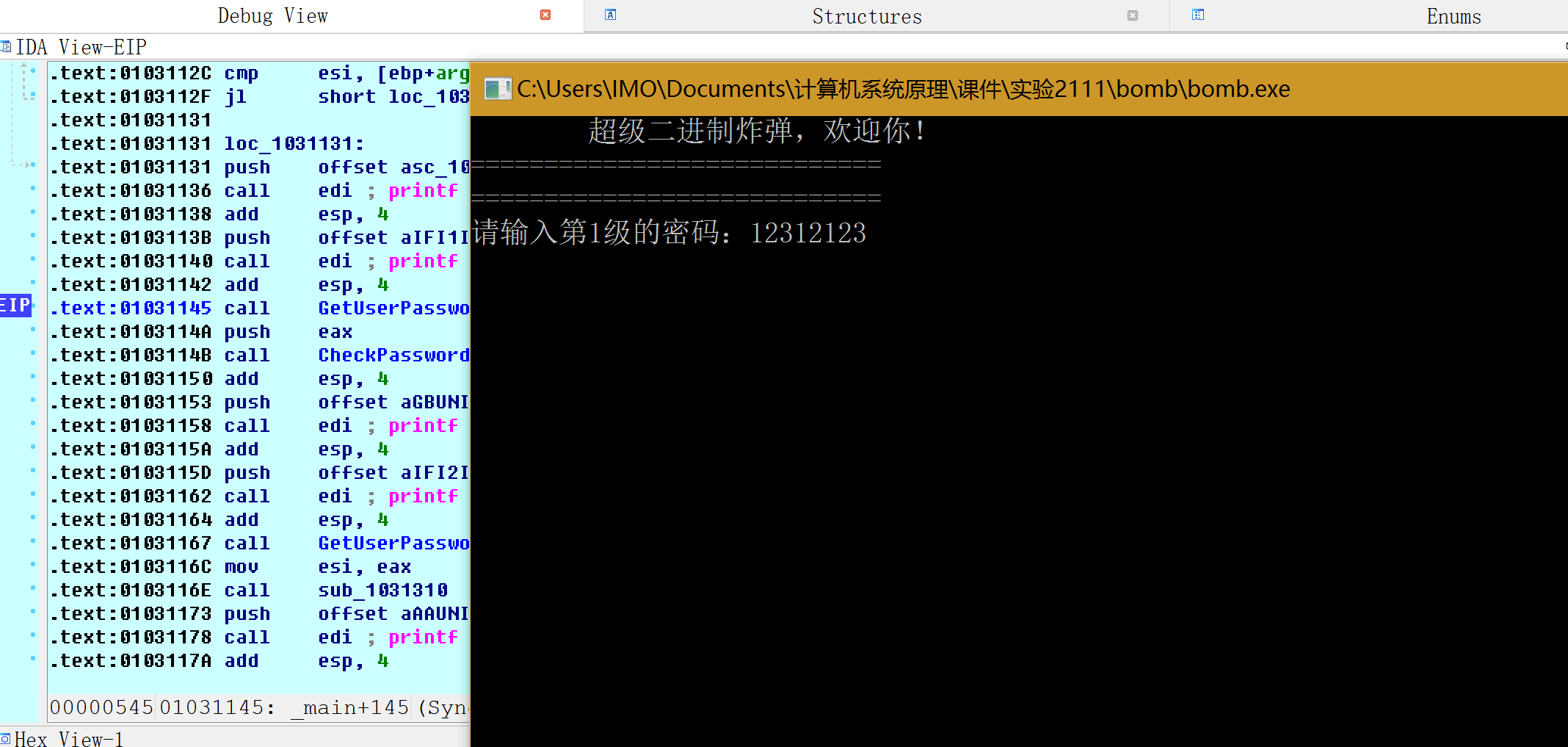




回到代码看看，有改变，比较容易看懂了~~~



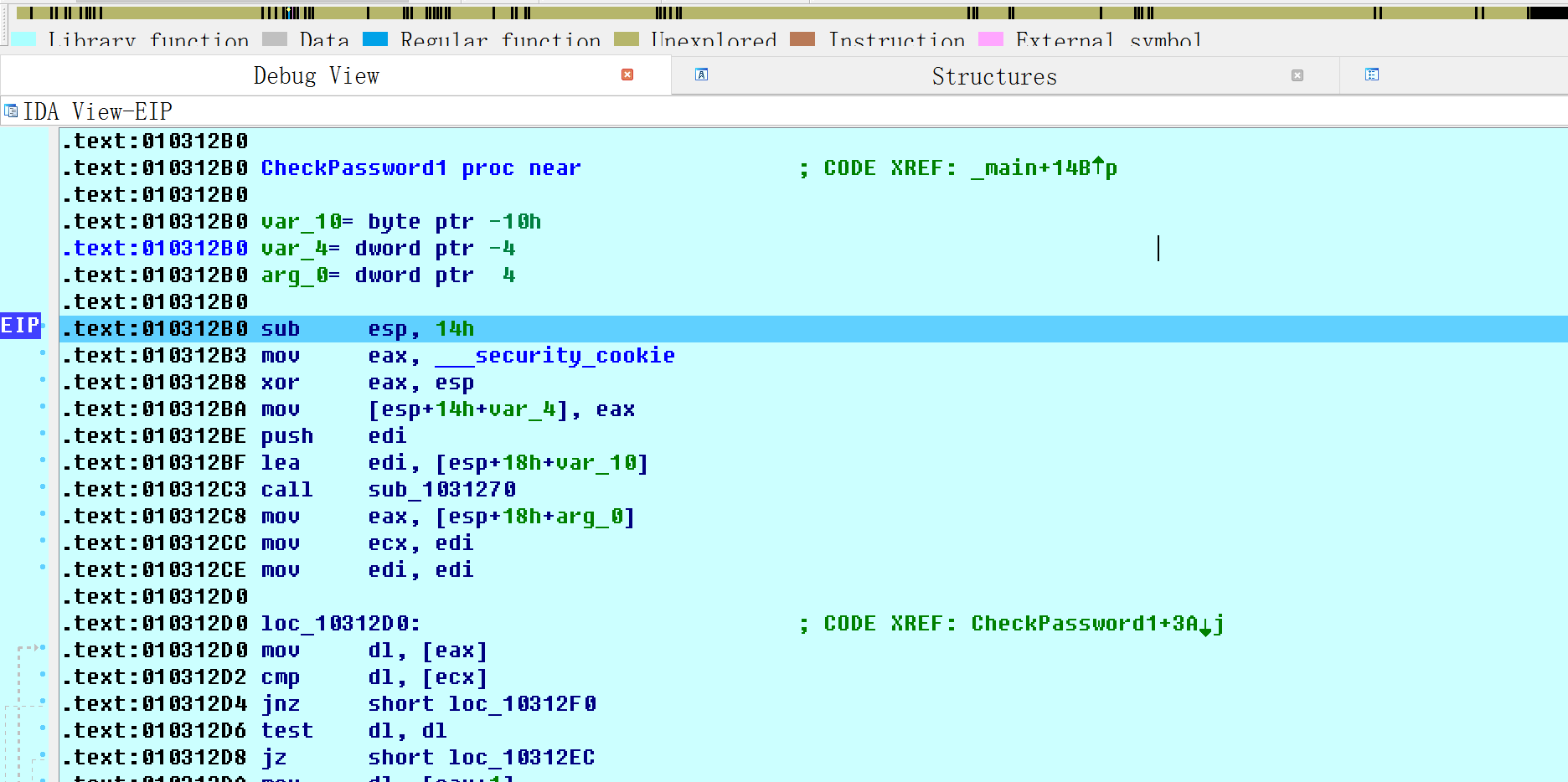
1. 我们破解bomb.exe，需要输入正确的口令。显然一开始你是不知道什么是正确的口令的，因此当继续单步执行后，程序会首先让你输入一个口令，切换到运行窗口，随便输入一个字符串，然后回车。



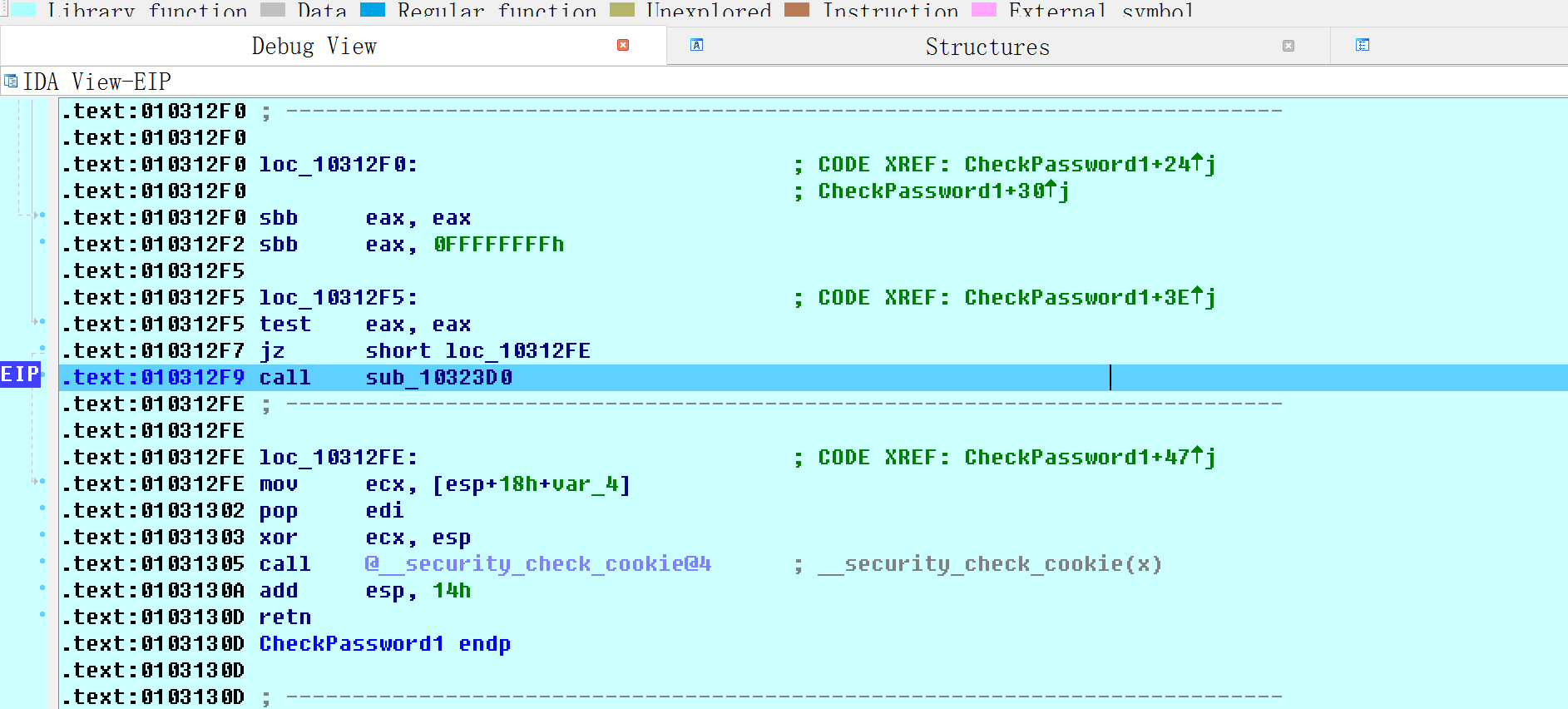
1. 程序会继续单步执行，停在GetUserPassword下面一行（因为刚才我们输入了一个字符串，回车。表明输入结束了，所以GetUserPassword函数执行结束了）。



1. 按F7单步执行（注意，我们要进入CheckPassword1函数里面去看看，所以不能按F8直接跳过，因为我们输入的口令显然不对，会爆炸的~），进入到CheckPassword1函数。



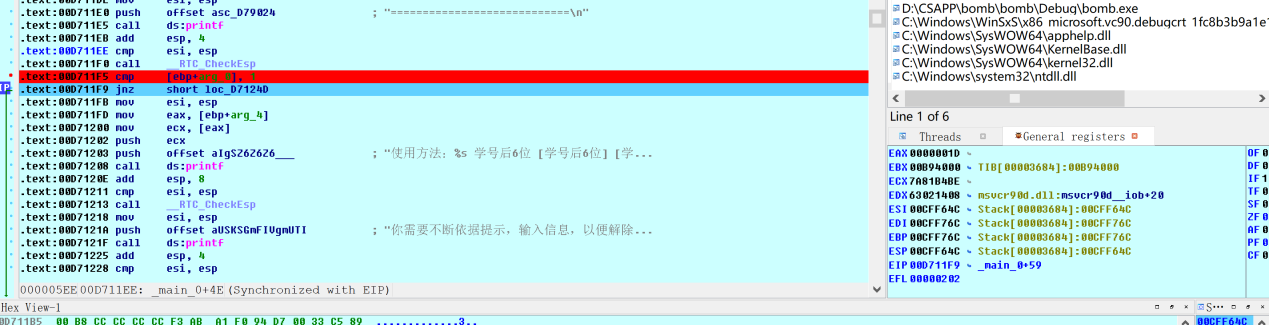
1. 按F8单步运行，快速的看一下CheckPassword1这个函数到底干了什么。。。。（不要按F7单步执行。因为我们的目的是快速的看一下这个函数到底想干什么，了解它大概的意图，不追求细节。）



1. 当执行到call sub\_10323D0的时候，按F8，程序爆了。所以提醒我们，不能让程序运行到这句话。如何让它不执行到这句话呢？上面有两条指令test eax,eax jz short loc\_10312FE。显然，要让这个跳转发生，跳过去。所以需要eax=0。只有eax=0，那么就会跳过call sub\_10323D0。向前面追溯看看，eax是如何计算的？重新启动整个调试过程吧。

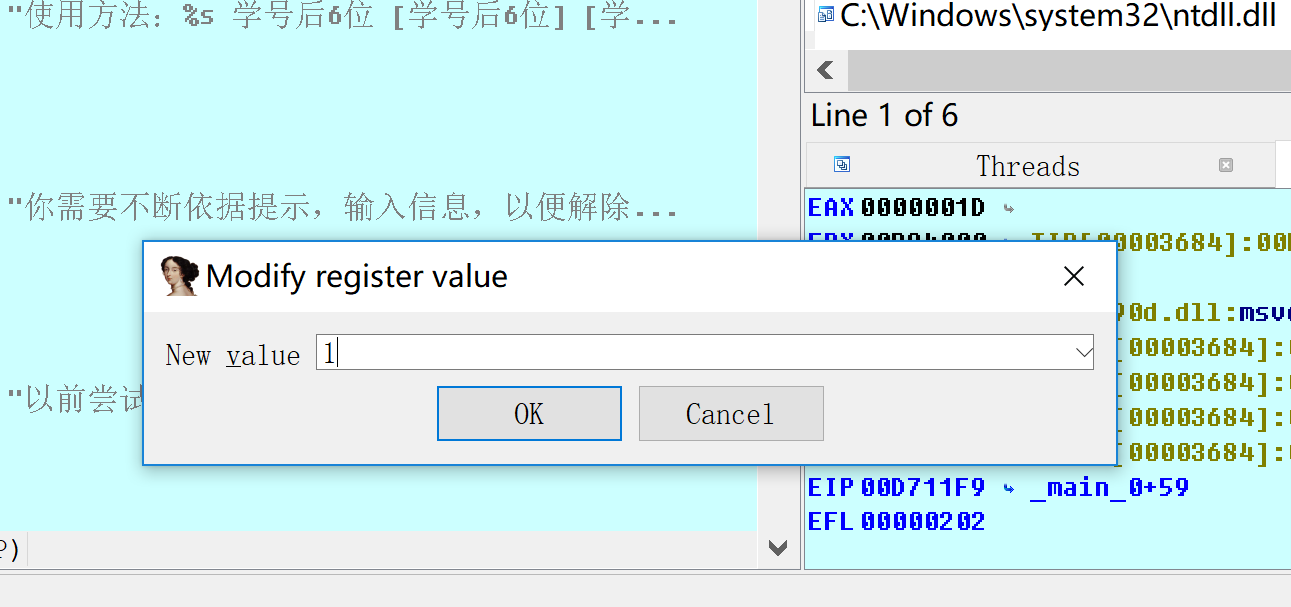
**调试过程中修改寄存器的值**

作为一个调试器，IDA Pro能够在调试程序的时候，动态地修改寄存器的值，这是一个强大的功能。当你在调试过程中，需要暂时修改某些寄存器的值的时候，这个功能非常有用。



比如，在上图中，在断点处，执行完cmp语句后，ZF标志寄存器为0（在最右边的寄存器窗口可以看到），因此后面的jnz语句会跳转的。但是某些时候，我们并不希望执行跳转，此时，可以到最右侧的寄存器窗口中修改ZF标志寄存器，然后再继续执行。

用鼠标双击ZF寄存器边上的数值0，此时，会弹出一个窗口“Modify Register Value”，输入1，确定。



此后，继续执行jnz指令，将不会跳转。

此方法也可以用于修改其他的通用寄存器，比如EAX、EBX等等，甚至可以修改EIP，达到强行设置PC的目的。