《实验二》

上机实验指导书

**(最新版)**

# 实验二：软件的破解实验

**实验项目名称：软件的破解实验**

**实验项目性质：综合性**

**所属课程名称：逆向工程技术**

**实验计划学时：4学时**

**一、实验目的**

掌握基本的软件破解技术。

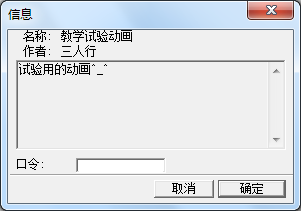
**二、实验内容**

1.认真阅读和掌握本实验的相关知识点；

2.完成实验并将实验结果记录到实验报告中。

**三、实验内容**

1. 上课演示的实例的重现：
   1. 请去除“Nag.exe”程序的警告窗口。
   2. 去除“动画.exe”程序的密码比较，即使输入的密码不正确，点“确定”以后也可以像输入了正确密码一样正常运行。
2. 请找出“动画.exe”的正确口令。提示：该程序会在内存中出现正确的口令。
3. 进一步破解“动画.exe”程序，让其开始运行不弹出下面的对话框而直接像输入了正确密码一样运行：



提示：

* 1. 该程序使用VC++的MFC方式编写，使用的是C++语言。在MFC编程时，也可以使用CDialog类的成员函数DoModal()来弹出对话框。请自行查阅该函数的帮助信息。
  2. 使用call 指令调用一个函数时，该指令的下一条指令的地址会压栈，在栈里可以找到下一条指令的地址。那么上一条指令就是call指令本身了。(本条提示可能在本例用不上)

1. (可选) 请破解adslftpserver.exe程序，让其成为注册版本：

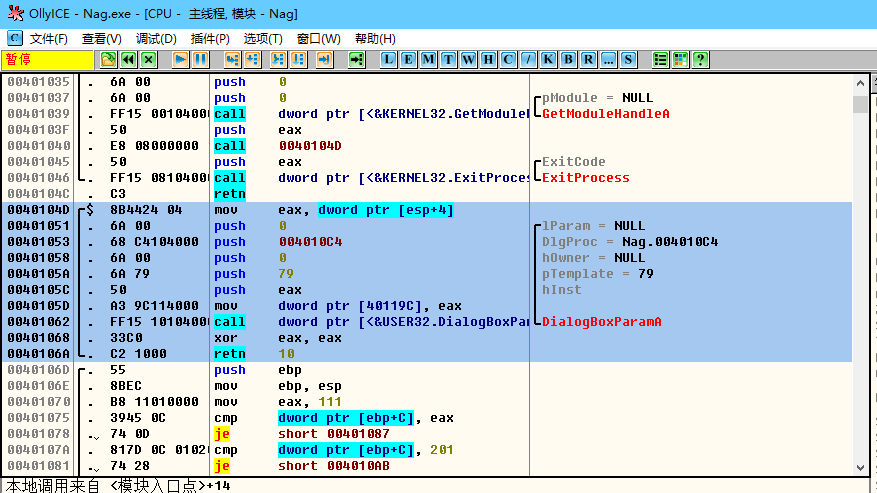


**四、实验步骤**

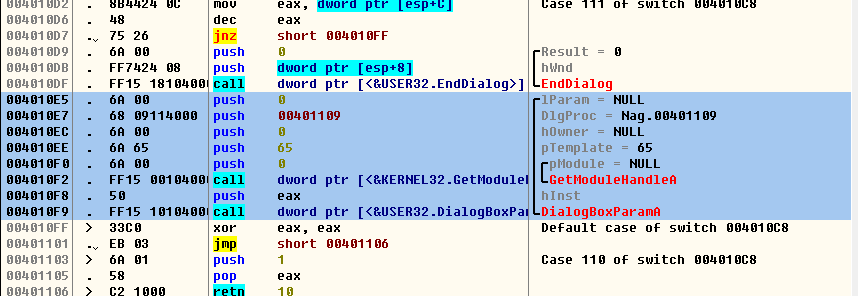
1. 上课演示的实例的重现：
   1. 请去除“Nag.exe”程序的警告窗口。

打开OD调试“Nag.exe”程序，通过search for找到调用对话框的三个模板，分别下断点依次调试

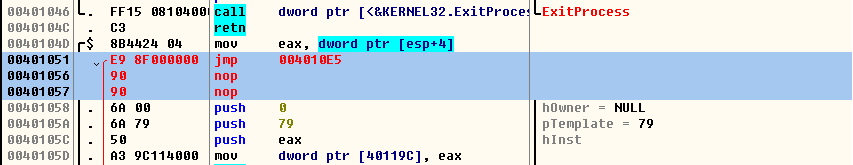
经过单步调试，发现下列代码就是弹出警告对话框的代码：



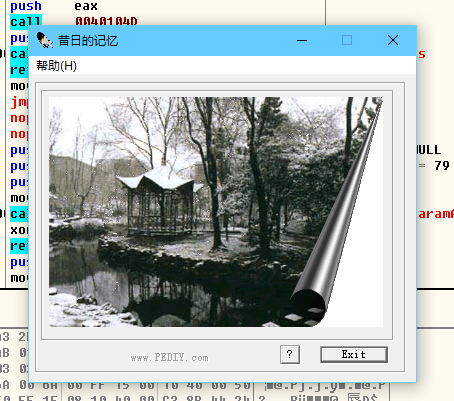
而下面的代码是弹出原始窗口的代码：



故只需要在弹出警告对话框之前跳转到原始对话框即可。于是修改弹出警告对话框的代码，把push 0改成jmp 004010E5：

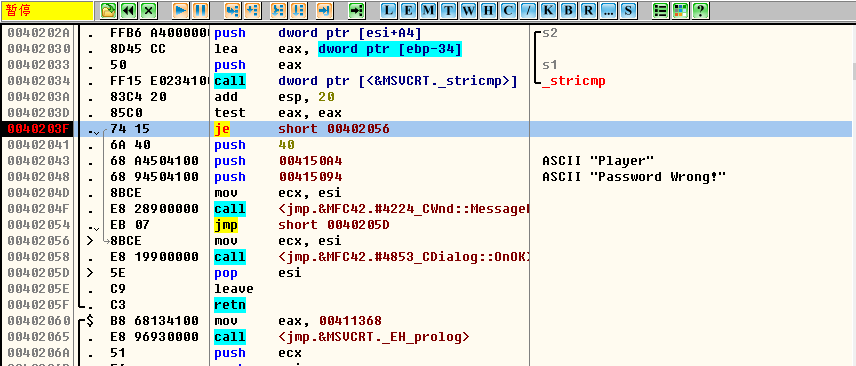


然后运行程序，发现没有原来的警告窗口了：

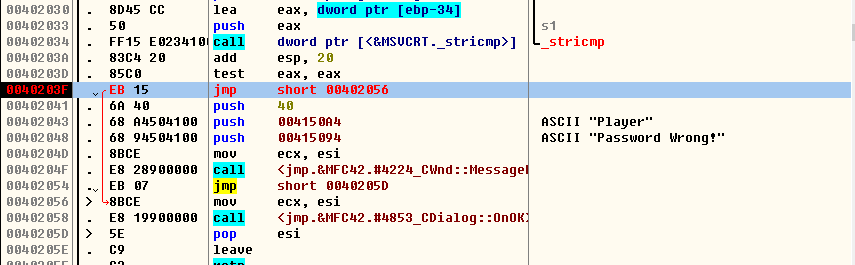


* 1. 去除“动画.exe”程序的密码比较，即使输入的密码不正确，点“确定”以后也可以像输入了正确密码一样正常运行。

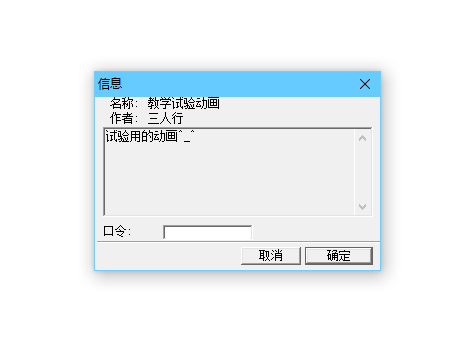
用OD打开“动画.exe”，经过单步调试，发现此次跳转是用来判断密码是否正确，若密码正确则跳转：

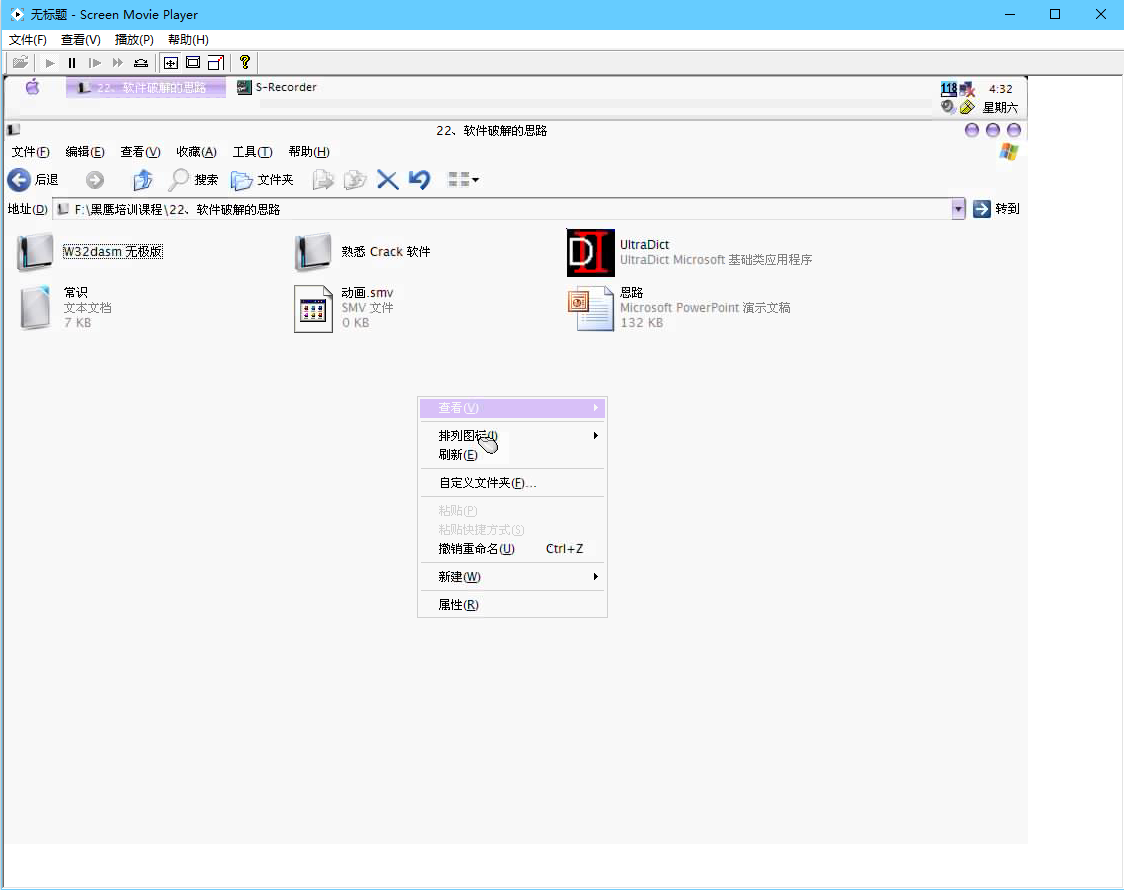


于是将其改成无条件跳转：

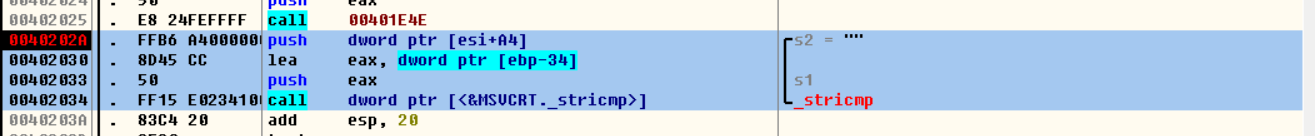


运行程序，发现即使输入的密码不正确，点“确定”以后也可以像输入了正确密码一样正常运行：

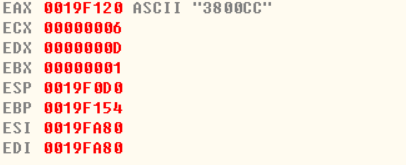




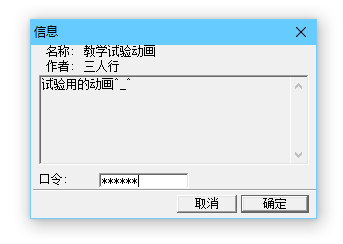
1. 请找出“动画.exe”的正确口令。提示：该程序会在内存中出现正确的口令。通过在上面的实验的基础上向上搜索，发现程序调用了一个字符串比较指令。

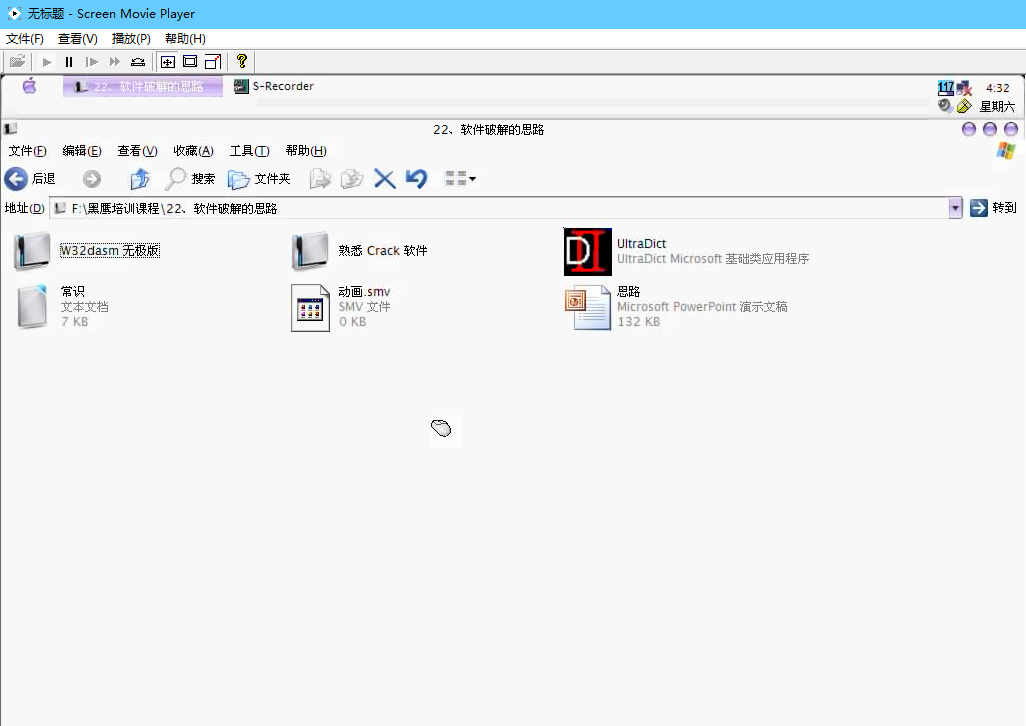


而此时EAX寄存器的值为3800CC

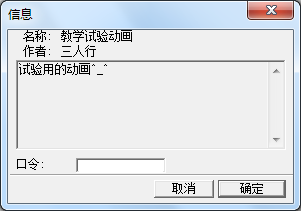


故正确口令为3800CC，经过测试，在修改前的程序输入密码3800CC，发现可以进入程序：

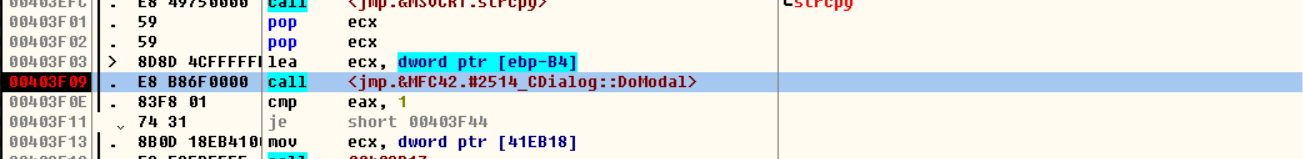
****



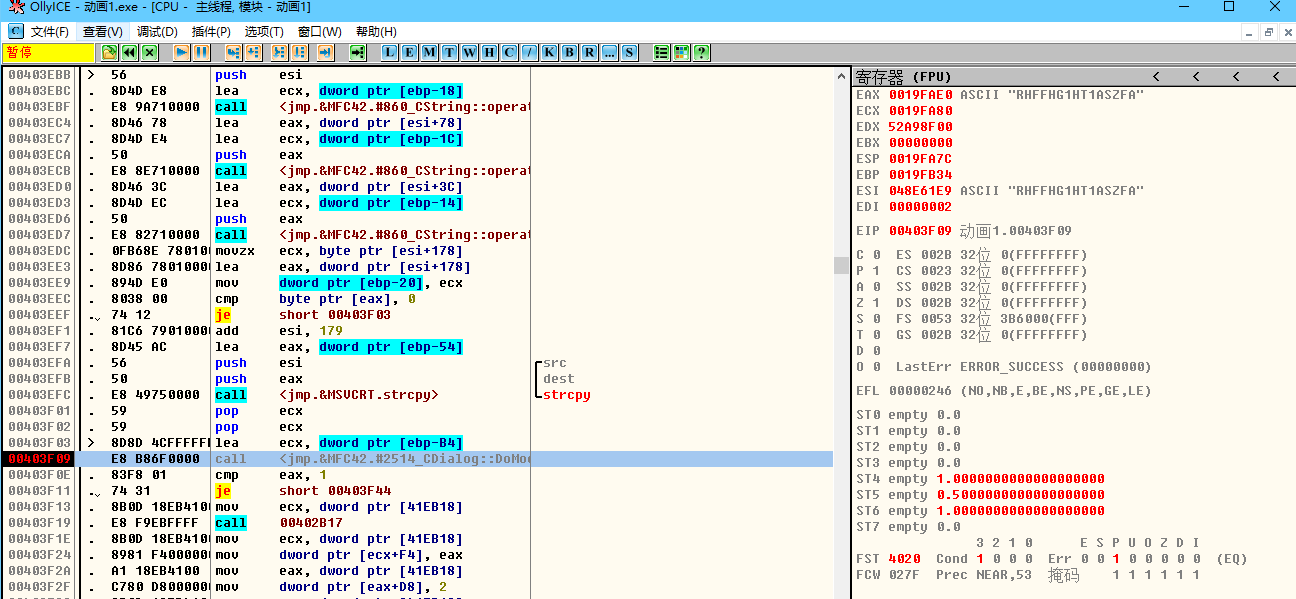
1. 进一步破解“动画.exe”程序，让其开始运行不弹出下面的对话框而直接像输入了正确密码一样运行：



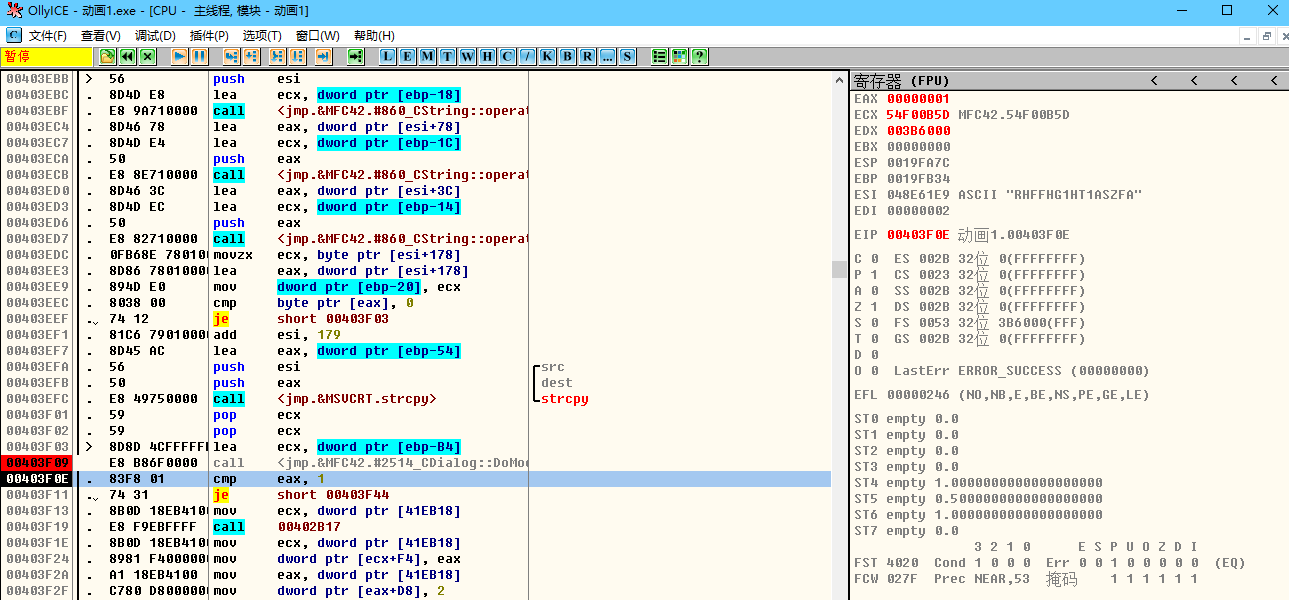
通过不断下断点调试，发现程序会在下面的代码处弹出如上的对话框：



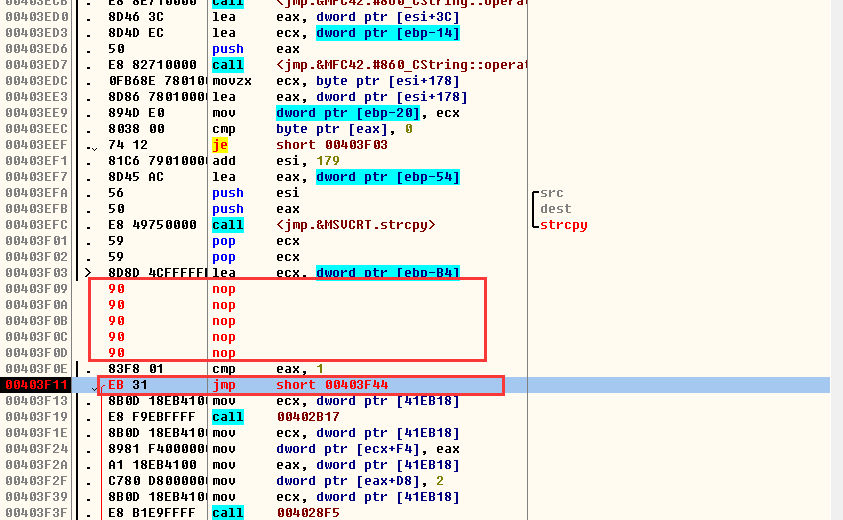
弹出对话框之前寄存器状态：



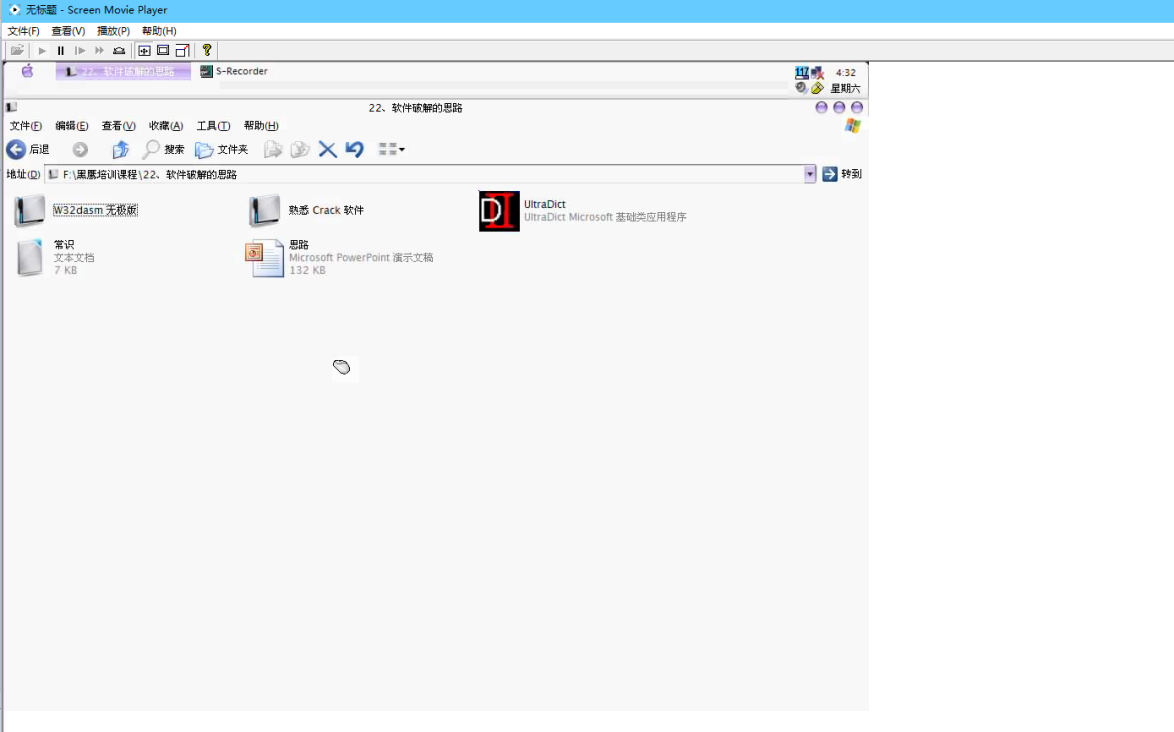
对话框关闭之后寄存器状态：



发现ESP没有变化，于是直接NOP掉该指令：发现程序直接退出。于是进一步观察，发现是由于下面的跳转指令没有实现，于是将该跳转改为强制跳转：



再次运行程序，发现成功去除对话框：



提示：

* 1. 该程序使用VC++的MFC方式编写，使用的是C++语言。在MFC编程时，也可以使用CDialog类的成员函数DoModal()来弹出对话框。请自行查阅该函数的帮助信息。
  2. 使用call 指令调用一个函数时，该指令的下一条指令的地址会压栈，在栈里可以找到下一条指令的地址。那么上一条指令就是call指令本身了。