《漏洞利用及渗透测试基础》实验报告

姓名：管昀玫 学号：2013750 班级：计科一班

**实验名称：**

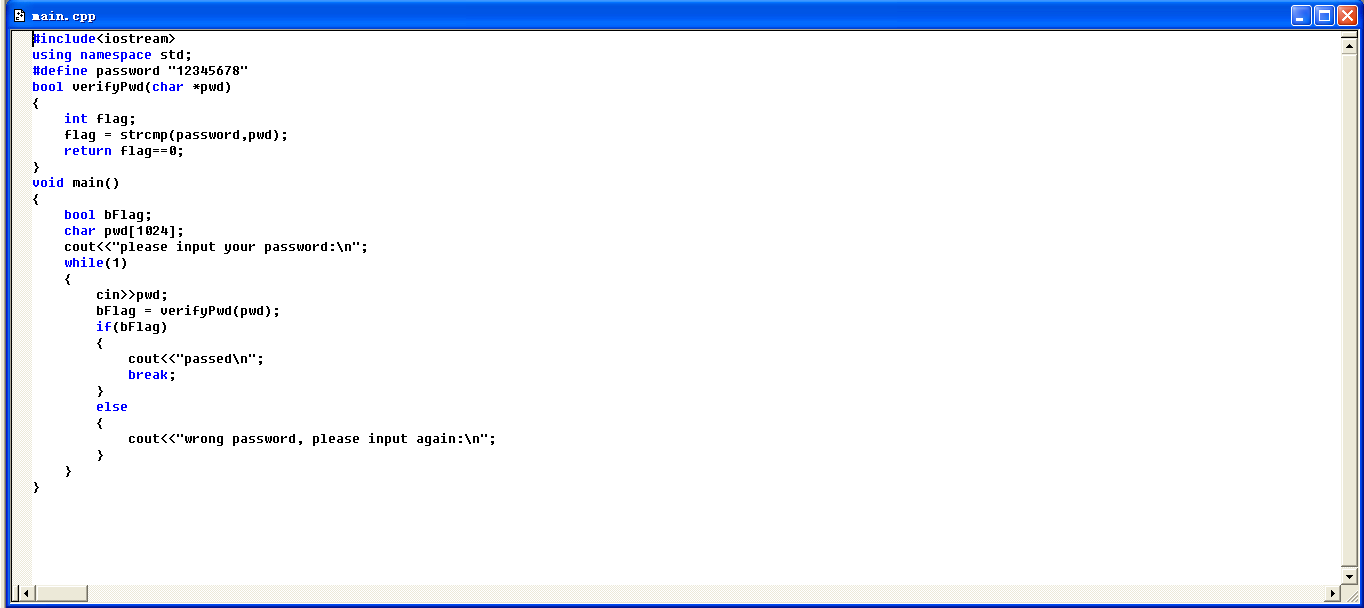
OLLYDBG软件破解

**实验要求：**

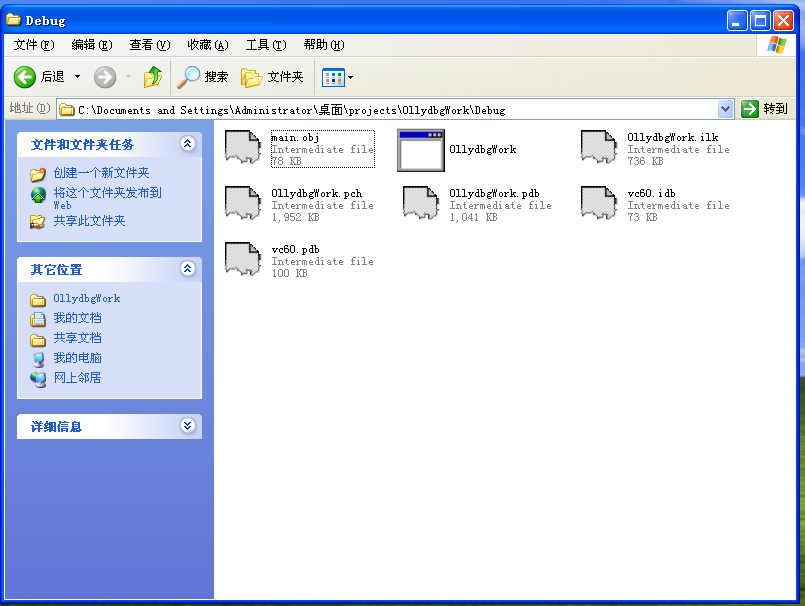
1. 请在XP VC6生成课本第三章软件破解的案例(DEBUG模式，示例3-1)。进而，使用OllyDBG进行单步调试，获取verifyPWD函数对应flag==0的汇编代码，并对这些汇编代码进行解释。
2. 对生成的DEBUG程序进行破解，复现课本上提供的两种破解方法。

**实验过程：**

1. **软件准备**
   1. 在VC6中编写相应的代码

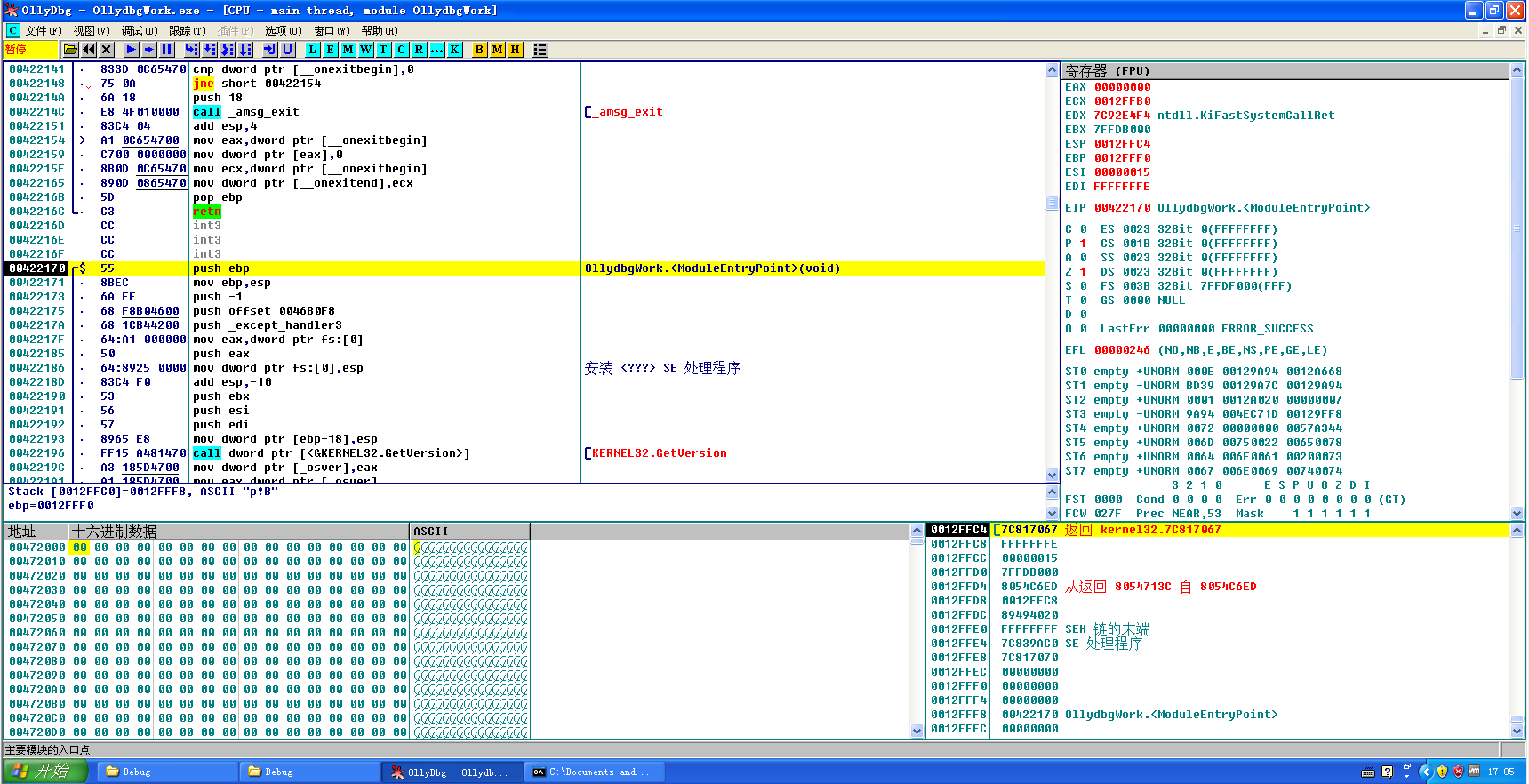


* 1. 将源码在Debug进行编译，生成exe可执行文件



其中OllydbgWork即为生成的可执行文件。

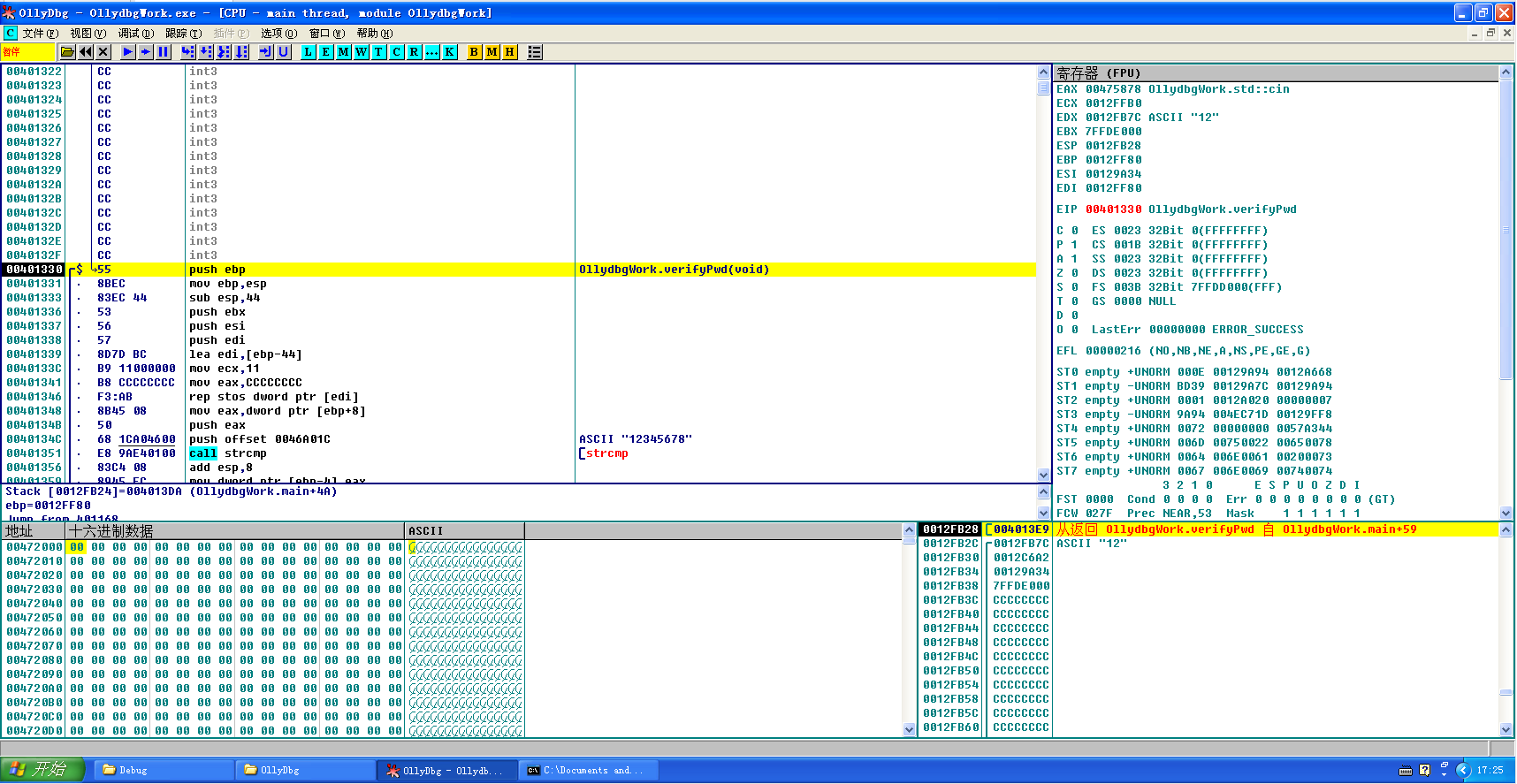
1. **查看反汇编代码**
   1. 使用OllyDBG软件，打开刚刚生成的可执行文件，查看相应反汇编代码



可以看到左上窗口为反汇编窗口，右上为寄存器窗口（查看寄存器当前的状态），左下为内存窗口（查看内存地址上存放的相关数据），右下为栈窗口（查看当前栈中的状态）

* 1. 分析verifyPwd函数的反汇编代码

通过查找引用的“wrong”字符串，查找到引用的位置；根据call语句进行两次跟随，进入到verifyPwd函数体中：



verifyPwd函数反汇编代码如下

CPU Disasm

地址 汇编代码 注释

00401330 push ebp ；将main函数的ebp的值入栈

00401331 mov ebp,esp ；把esp的值赋给ebp，移动栈帧

00401333 sub esp,44 ；esp抬高，开辟空间

00401336 push ebx

00401337 push esi

00401338 push edi ；三个寄存器入栈

00401339 lea edi,[ebp-44] ；把ebp-44这个位置上存的地址赋给edi

0040133C mov ecx,11

00401341 mov eax,CCCCCCCC

00401346 rep stos dword ptr [edi] ;利用循环开辟空间，并赋初值

00401348 mov eax,dword ptr [ebp+8] ; ASCII "1111"，为输入的值

0040134B push eax

0040134C push offset 0046A01C ; ASCII "12345678"

00401351 call strcmp ;调用strcmp

00401356 add esp,8 ；esp下移

00401359 mov dword ptr [ebp-4],eax ；将eax的值（即strcmp比较的结果）保存 到[ebp-4]中

0040135C xor eax,eax ；xor是异或比较，此处将eax设置为0

0040135E cmp dword ptr [ebp-4],0 ；比较[ebp-4]和0，设置标志位。如果[ebp-4] 中存放的是0（即strcmp中比较结果相 同），则ZF设为1，反之则为0

00401362 sete al ；sete是根据状态寄存器的值，如果相等则设置，如果不等，则不设置

00401365 pop edi

00401366 pop esi

00401367 pop ebx ；弹出三个寄存器指针

00401368 add esp,44 ：esp下移

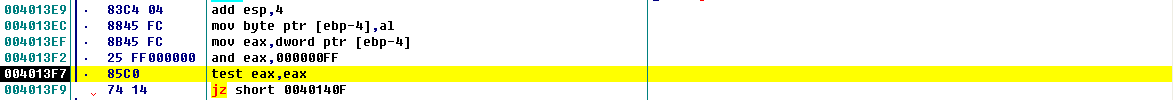
0040136B cmp ebp,esp

0040136D call \_chkesp ；检查esp

00401372 mov esp,ebp ；esp回到之前main函数的esp状态，栈帧 恢复

00401374 pop ebp ；ebp弹出

00401375 retn ；返回到main函数中



检查eax是否是1（test eax, eax ；将eax自身做与操作，并给ZF赋值）

与的结果是0则跳转到0040140F，是1则顺序执行

顺序执行：

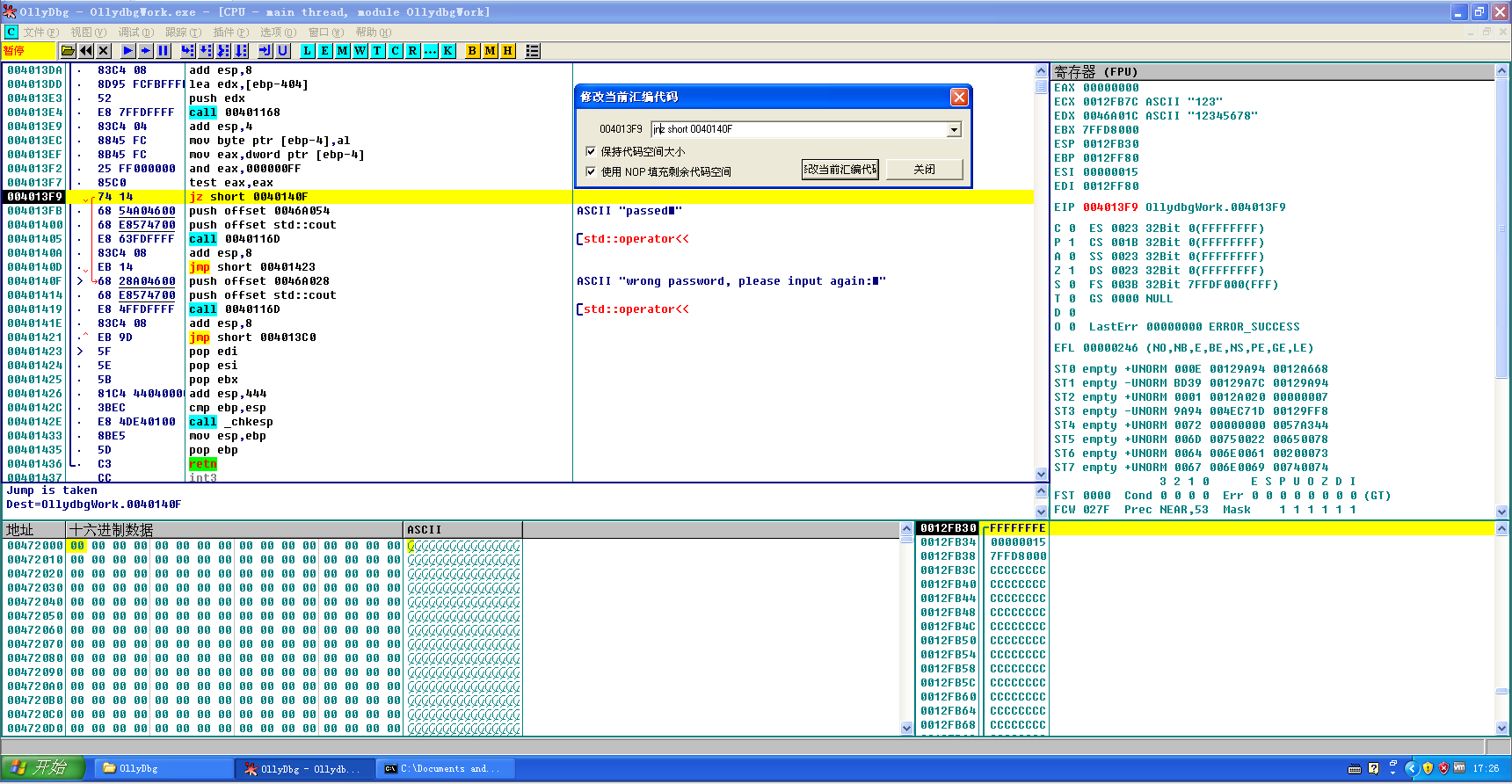


跳转执行（eax是0时）；



1. **实现软件破解**

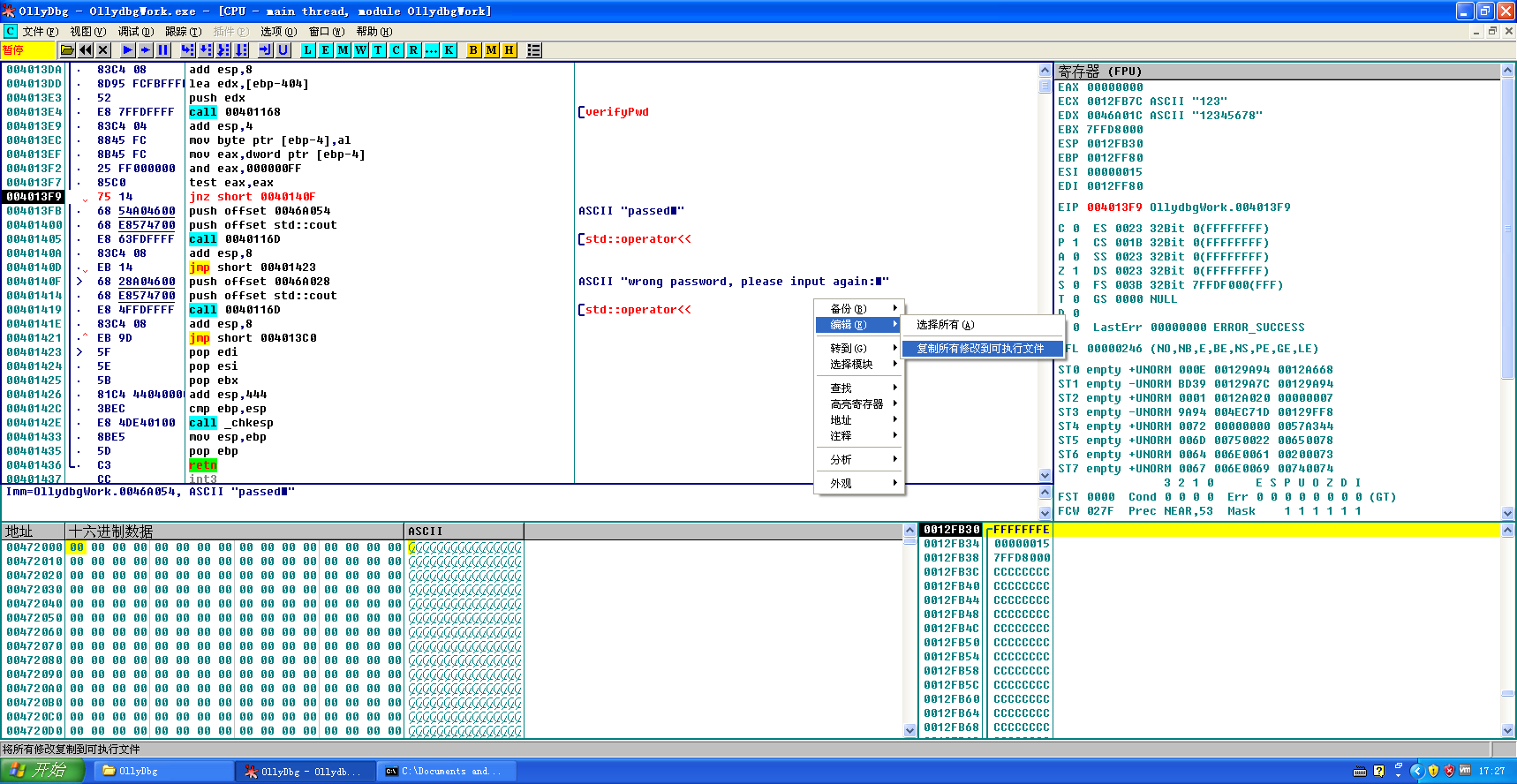
修改方式一：直接修改跳转逻辑，实现当密码错误时通过，密码正确时不通过：将jz逻辑取反，换成jnz。



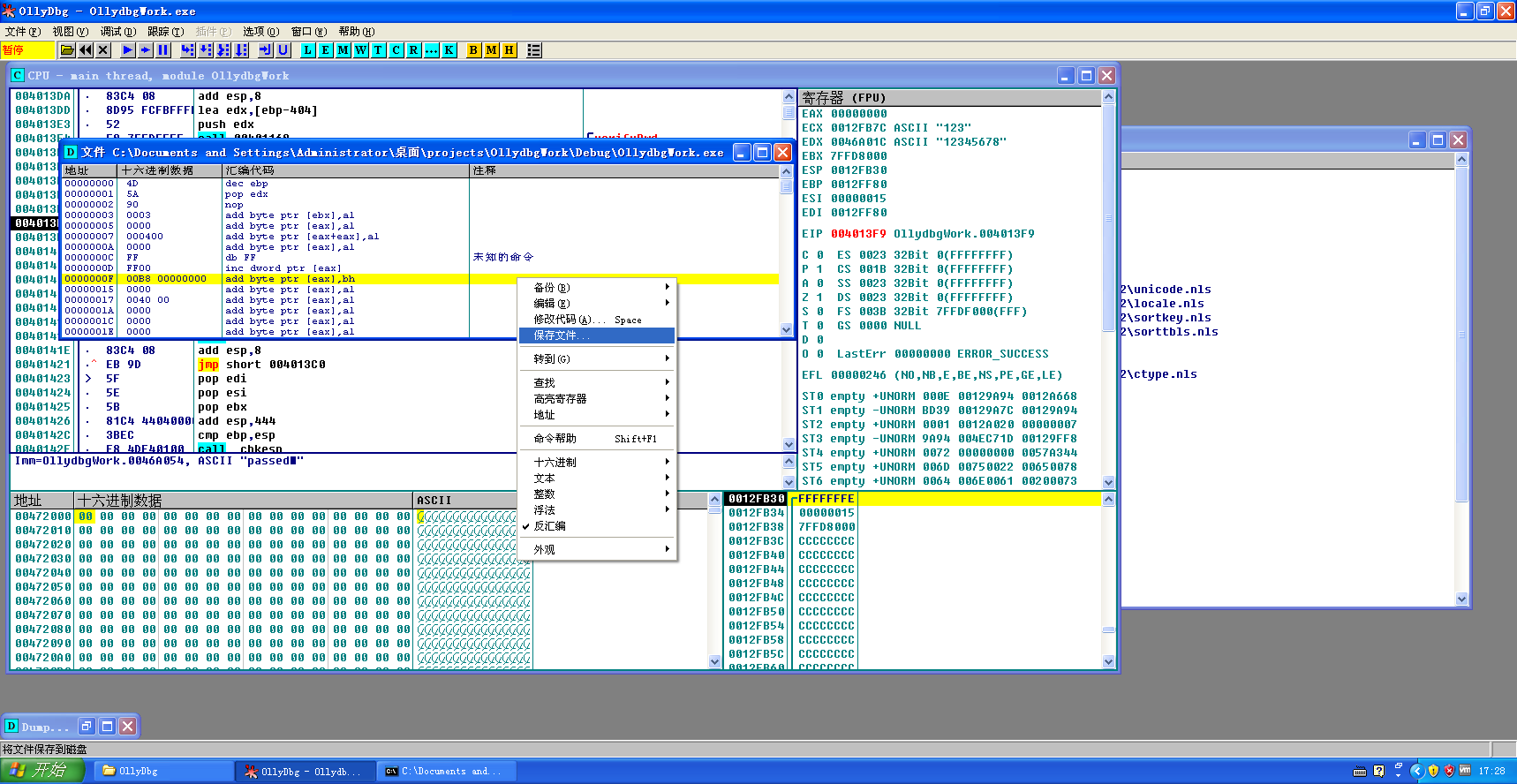
修改之后：



将修改添加到可执行文件当中：



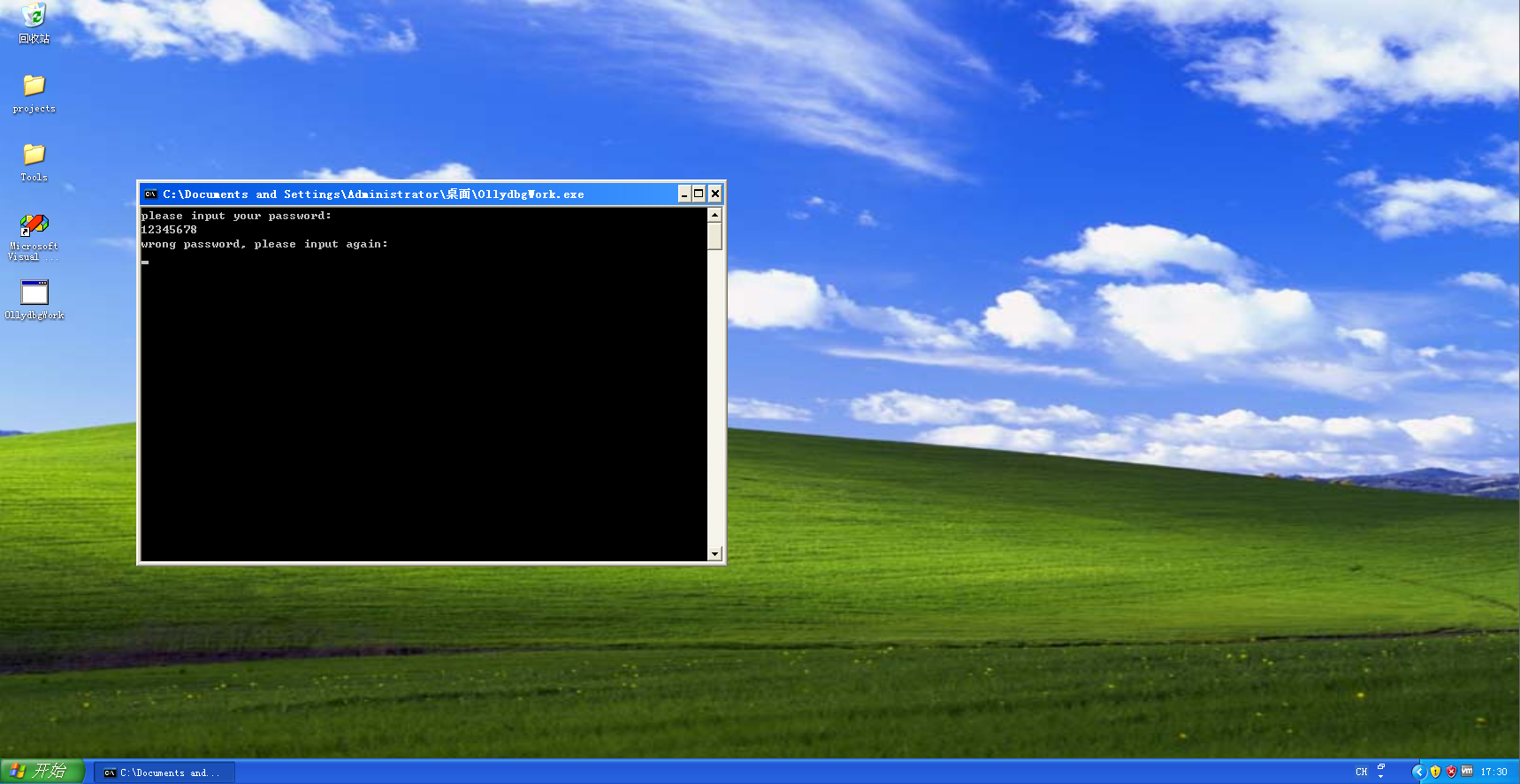
保存可执行文件：



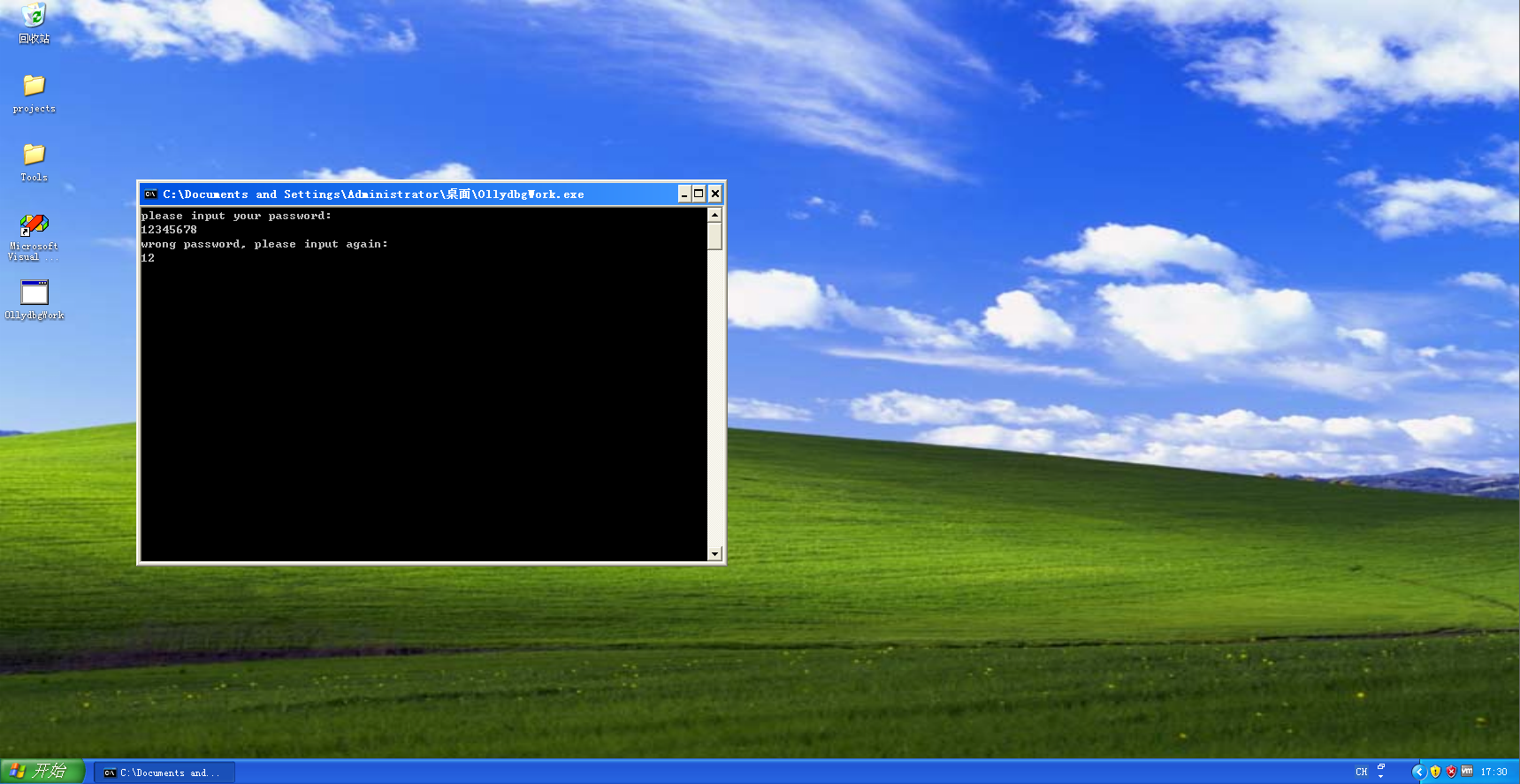
得到新的可执行文件



运行查看效果：



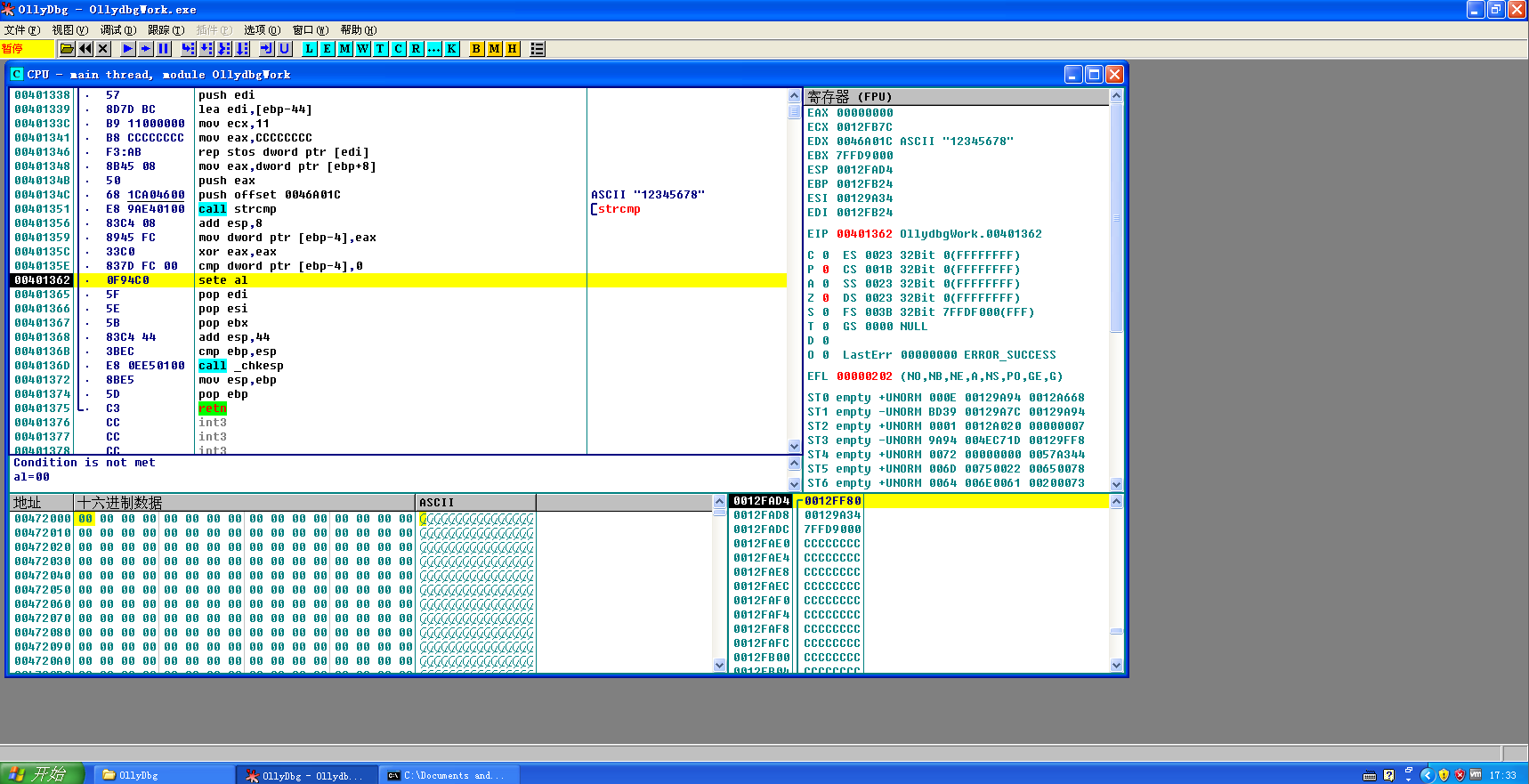
可见当密码正确时，显示为错误



当密码错误时，通过，程序控制台关闭。

破解成功；

* 1. 第二种方式：修改verifyPwd函数的返回值，使得不管输入什么都是正确



从上一步的xor eax, eax可以看出eax已经被清零；这里将al的值强制设置为1：

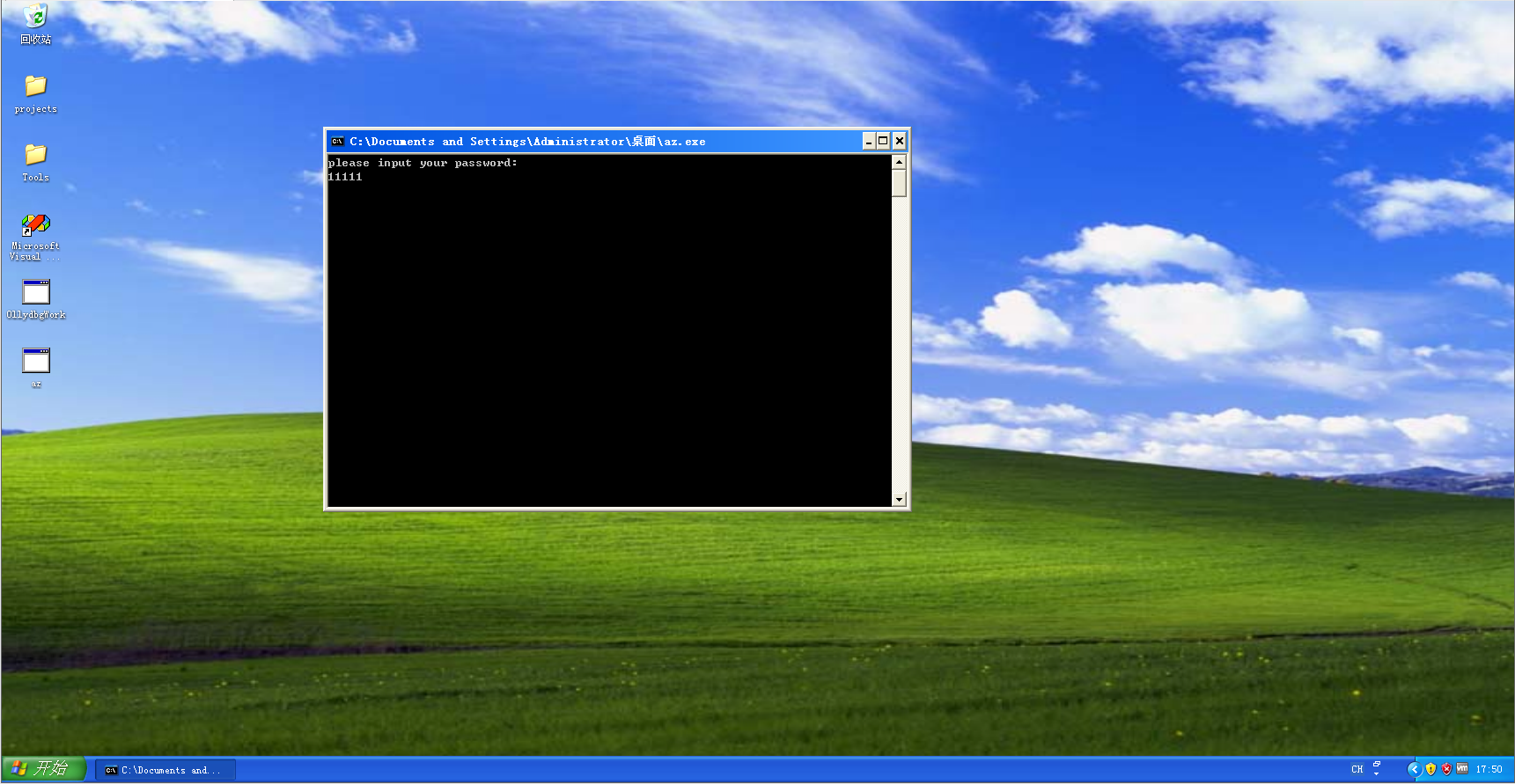


同样将修改后的代码保存到可执行文件当中，得到程序az



运行效果如下：

输入错误密码11111



通过，程序控制台关闭

破解成功

**心得体会：**

通过本次实验，我学会了OllyDBG的基本操作，对反汇编指令更加熟悉，学会了利用OllyDBG进行源码的修改、破解。期间遇到了一点问题：如为什么要在给eax赋返回值后还需要把eax给到其他的地方，然后还要再清零，清零之后再进行一次比较、赋值；如在将栈帧调整过后，esp上移开辟空间后，rep stos的循环操作目的是什么。同时也使我对这门课程更加感兴趣。我还有一点想法：

第一种破解方式中，可以将jz改为jmp执行无条件跳转到通过的指令地址，同样可以达到无论输入什么都是正确的目的。

第二种修改方式中除了使用mov，应该也可以使用add等指令，只要能让AL为1即可。