/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

To be the apostrophe which changed "Impossible" into "I'm possible"!

POC code of chapter 3.7 in book "Vulnerability Exploit and Analysis Technique"

file name : crack\_me.c

author : failwest

date : 2006.9.20

description : used as a simple demo to show how to crack a PE file

Noticed : should be complied with VC6.0 and build into release version

version : 1.0

E-mail : failwest@gmail.com

Only for educational purposes enjoy the fun from exploiting :)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "stdio.h"

#define PASSWORD "1234567"

int verify\_password (char \*password)

{

int authenticated;

authenticated=strcmp(password,PASSWORD);

return authenticated;

}

main()

{

int valid\_flag=0;

char password[1024];

while(1)

{

printf("please input password: ");

scanf("%s",password);

valid\_flag = verify\_password(password);

if(valid\_flag)

{

printf("incorrect password!\n\n");

}

else

{

printf("Congratulation! You have passed the verification!\n");

break;

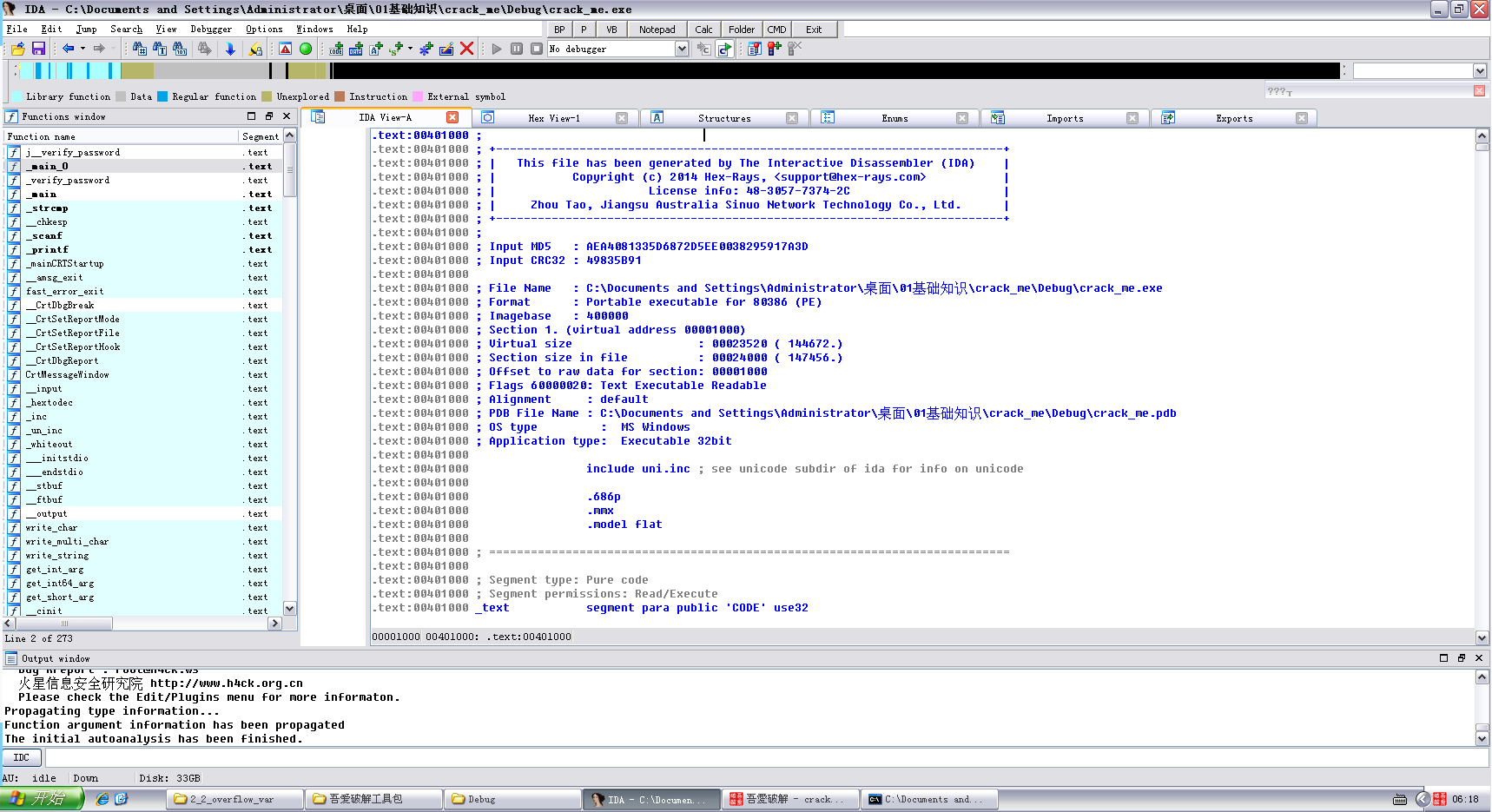
}

}

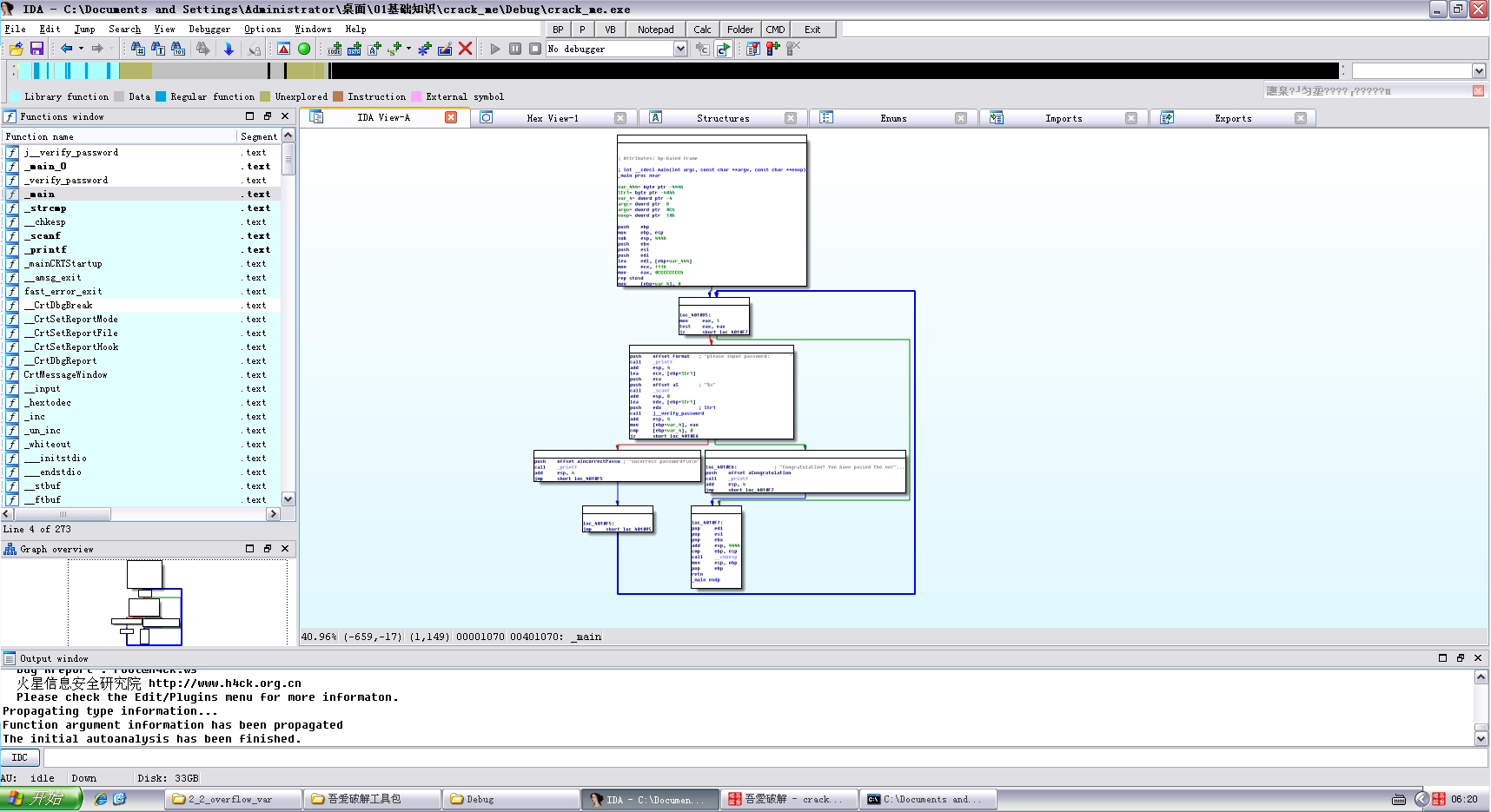
system("pause");

}

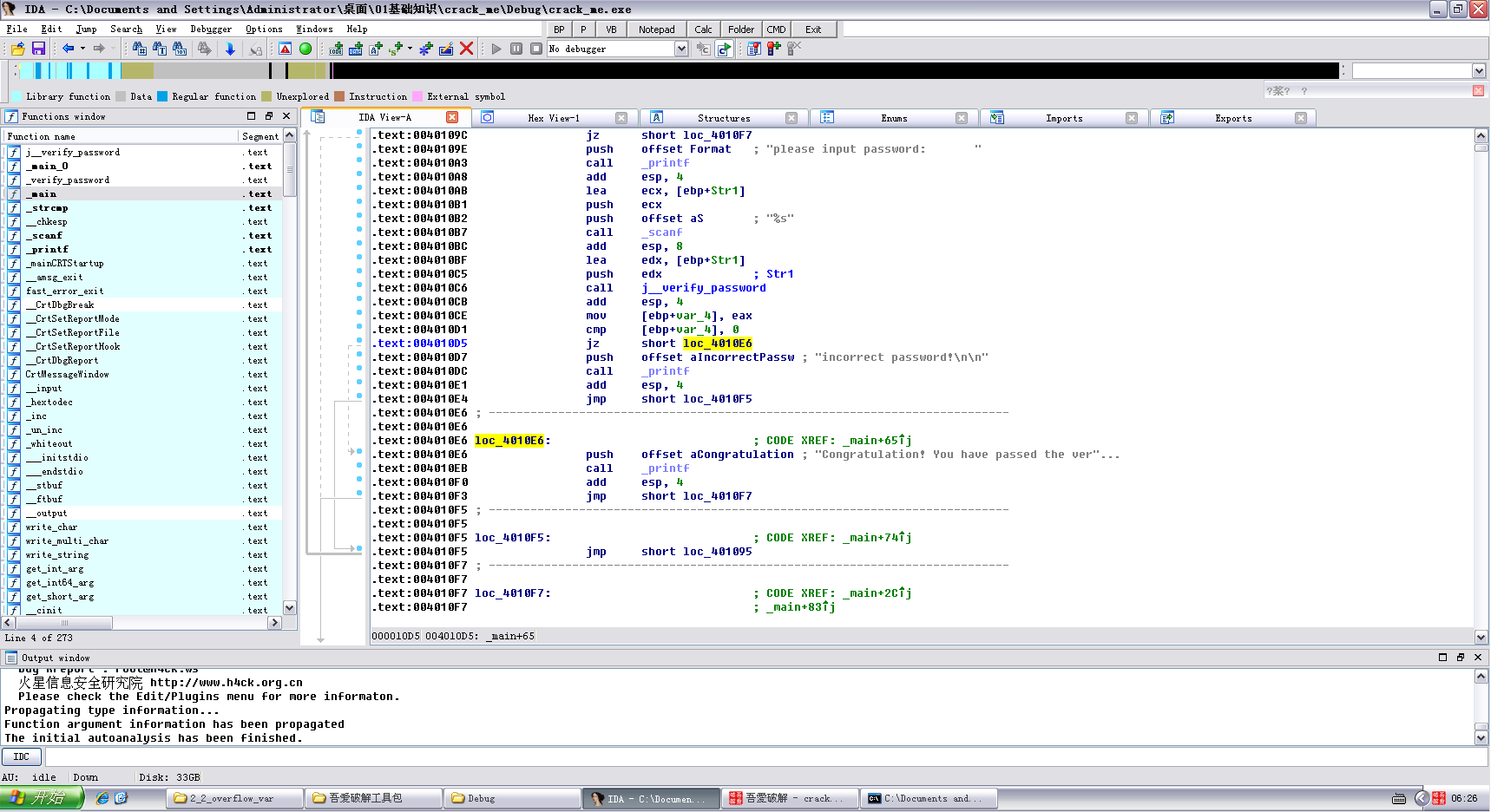
**预备知识：**  
**F2：下断点**  
**F7：跟入单步（比如一个call，会进入这个call的内部）**  
**F8：不跟入单步（比如一个call，不会进入这个call内部，而是直接执行这个call里面所有语句）**  
**F9：运行**  
**这东西拿去在VC++6.0下编译，然后搞出来一个exe**  
**VC++6.0是那么的好用，鄙视过老师用VC的都要拉出去砍了**  
**程序逻辑很是简单呐，就是一个控制台程序，如果输入的是“1234567”就正确，反之错误**  
**我们跟着0day2的排版走，看一看IDA Pro，如果你是第一次用IDA Pro的话呢，抱紧我，一直ok下去，那些可以关掉的提示就永远关了**  
**然后打开后就可以看到最开始的界面啦：）**



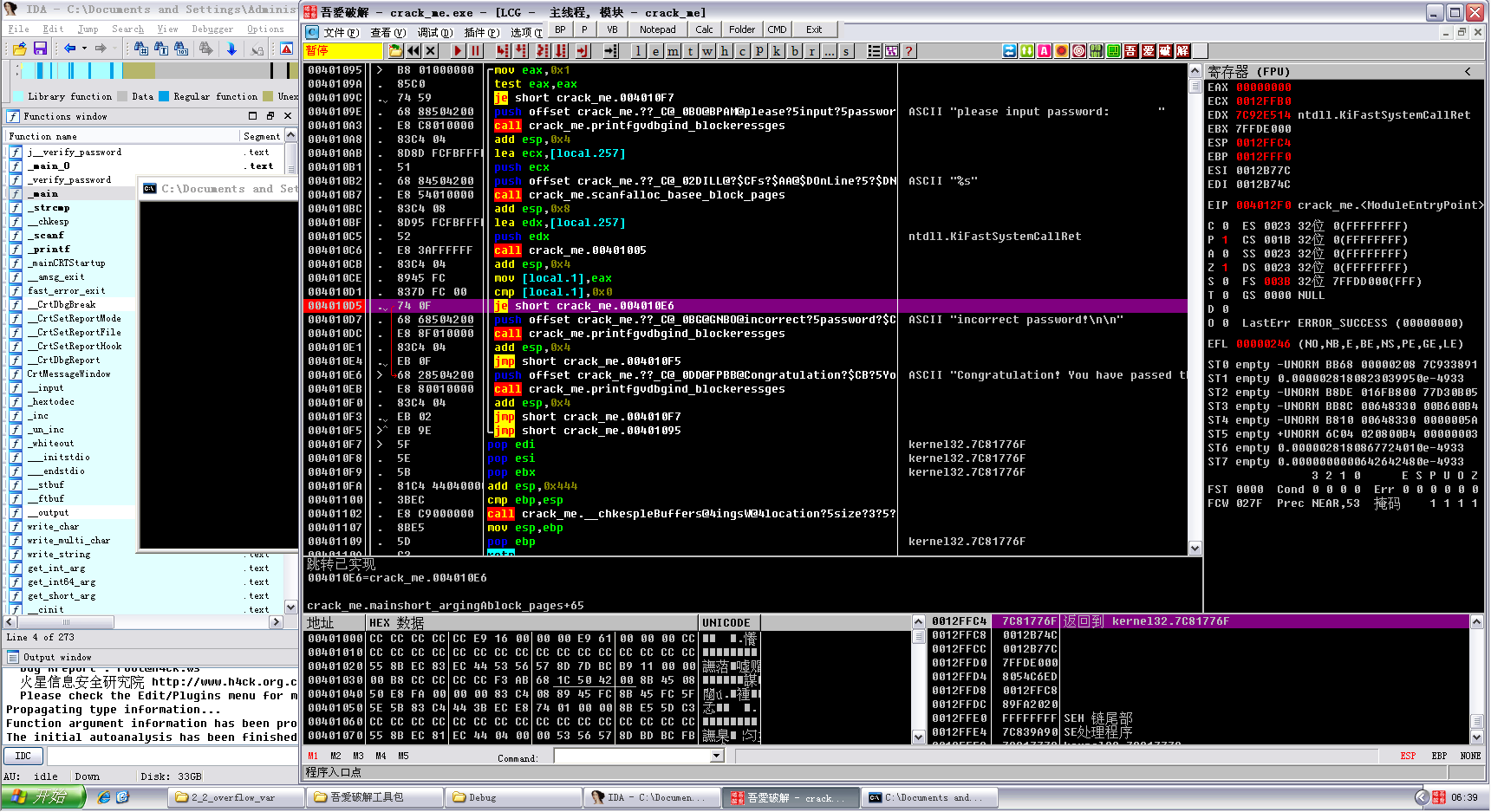
**左边都是函数之类的模块，都已经标记好了，比如上面的main就是main函数，点进去看看，然后按空格就可以看到流程图了，很明显看出来if所在**



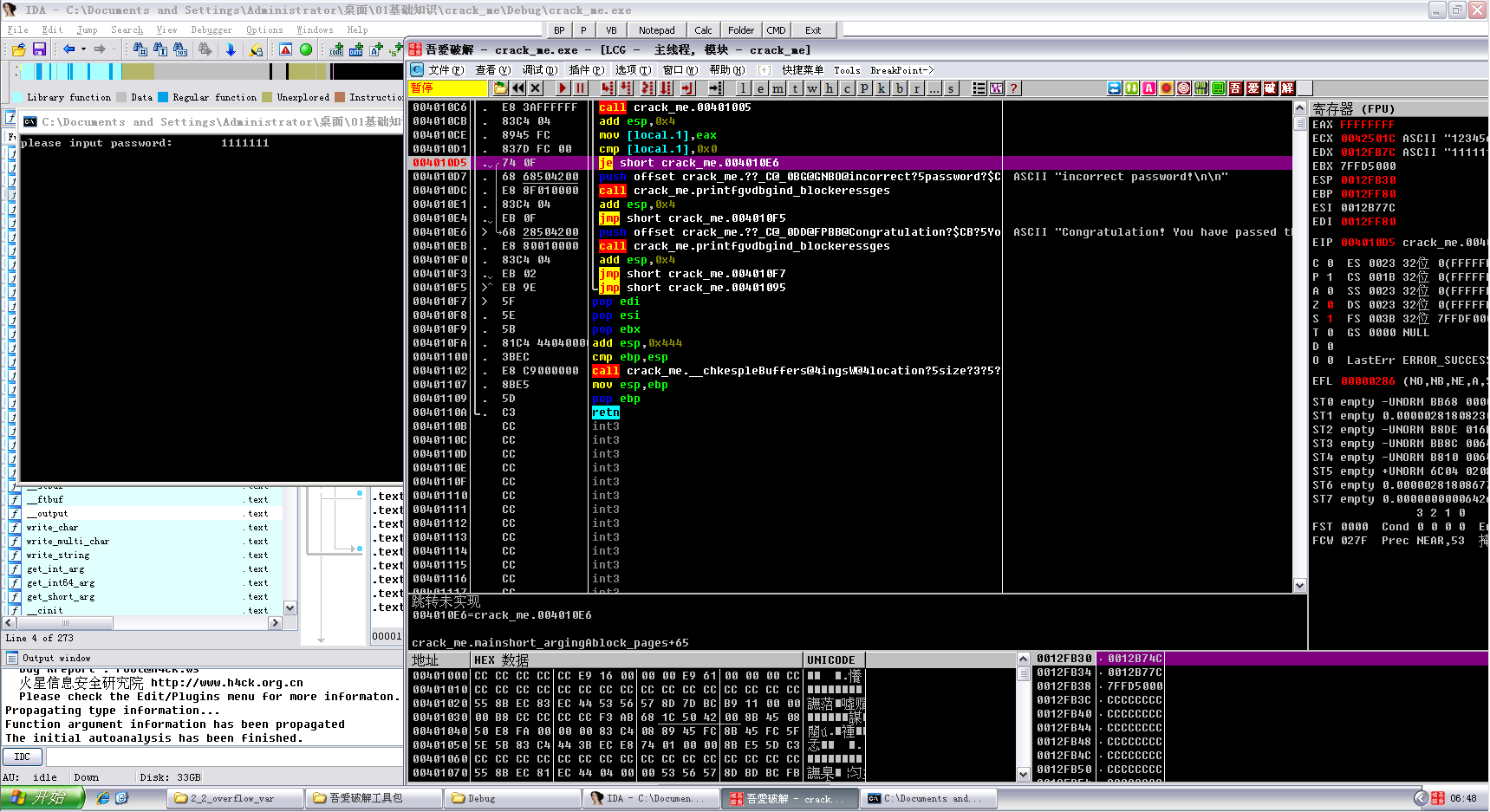
**如图亮点，第一个黄色正是if判断的跳转语句，记下VA，VA忘了是什么的抱紧我，VA全名virtual address，是虚拟内存地址，还记不记得RVA，忘了的抱紧我，RVA是relative virtual address，是相对虚拟地址，再回顾个Image Base，装载基址，PE被载入内存的时候，Image Base是最开始的地址，然后加上RVA就是VA，公式是VA=Image Base+RVA**

****

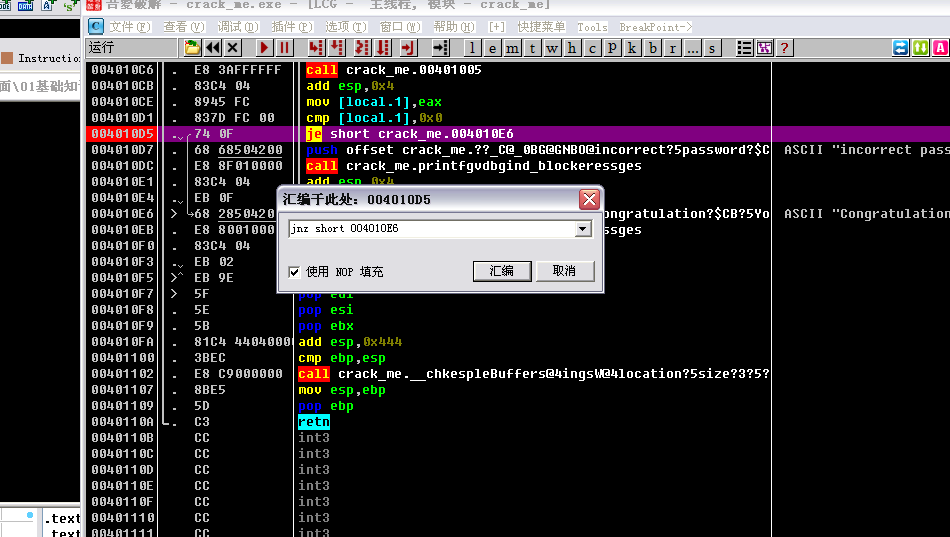
**上面搞清楚了之后讲一下节偏移：**  
**之所以有节偏移的存在是因为文件数据的存放单位和内存数据的存放单位不一样，PE文件中的数据按照磁盘数据存放，以0x200字节为基本单位，当一个数据 节不足0x200字节时，多余的填充0x00，超过0x200字节的使用下一0x200字节的空间，所以PE数据节的大小为0x200的整数倍**  
**但是在内存里单位就不是0x200字节了，而是0x1000，所以就出现了节偏移**  
**既然出现了节偏移，那么文件偏移地址的计算公式就是：文件偏移地址=VA-Image Base-节偏移=RVA-节偏移**  
**打开OD，载入控制台程序，刚刚也找到了跳转语句，所以Ctrl+G，直接输入VA：004010D5**  
**在跳转那里下个断点，不会下断点的抱紧我，F2或者直接双击都能下断点**



**然后运行程序，控制台程序一载入OD就运行起来没有关系，F8单步走下去就好了，或者F9直接运行，这里F9直接运行起来，输入 “1111111”，回车，发现卡住了，下了断点当然卡住了，先观察一下，可以看到je跳转未实现，那么修改的方法就比较多了，可以修改为jnz，或者直 接jmp到下面去，或者直接修改寄存器的值**



**那么就来修改je为jnz吧，这两个跳转指令的条件刚好相反**



**然后复制到可执行文件然后保存所有修改，妥妥的：）**  
**TEST EAX, EAX是什么意思呢？**  
**测试EAX是否为0**  
**test eax,eax**  
**je xxxxxxxx ;eax == 0则跳**  
**test eax,eax**  
**jnz xxxxxxxx ;eax != 0则跳**  
**为什么改为XOR EAX, EAX也可以呢？**  
**因为XOR EAX, EAX是将EAX清0，那么EAX就是0了和TEST判断0的效果是一样的**