计算机与网络体系结构（2）第三次作业

文庆福
2011013239 thssvince@163.com
清华大学软件学院11班

2014年 3月 13日

1. 写一条逻辑移位指令,将ECX的内容乘以4。

   shl ecx, 2

2. 写指令把BX的最低位移至CX的最高位（要求写出使用和不使用SHRD指令的两个版本）。

   1) shrd cx, bx, 1

   2) ror bx, 1 sar cx, 1

3. 文件的时间戳使用位0~4代表秒数，使用位5~10代表分钟，位11~15代表小时数，写指令小时数、分钟数、秒数并分别将数值复制到变量bHours、bMinutes、bSeconds。不妨设文件的时间戳为t

```
1     mov ax, 11111b
2     mov bx, 111111b
3     mov cx, t
4     shr cx, 11
5     and cx, ax
6     mov bHours, cx
7     mov cx, t
8     shr cx, 5
9     and cx, bx
10    mov bMinutes, cx
11    mov cx, t
12    and cx, ax
13    mov bSeconds, cx
```

4. 以汇编实现下面的C++语句（使用32位有符号整数）：

   var1 = (var2 * -6) / (-var3 % var4)

```
1   mov eax, val3
2   neg eax
3   cdq
4   idiv val4
5   mov ebx, edx
6   mov eax, −6
7   imul val2
8   idiv ebx
9   mov val1, eax
```

5. 分别描述ADC指令和SBB指令的功能。

   ADC指令时将源操作数、目的操作数以及进位标志相加；SBB则是从目的操作数中减去源操作数和进位标志的值。

6. 说明LEA指令与OFFSET指令有何不同？

   LEA是用于返回任意类型的间接操作数的偏移，间接操作数的偏移值只有在运行时才能得知。OFFSET返回编译时的偏移，不需要运行就可以得知。

7. 试声明一个指向双字数组的局部变量dArray。

   LOCAL dArray[10]:DWORD

8. 解释SMALL内存模式和FLAT内存模式。

   SMALL内存模式：包括一个代码段（64KB）和一个数据段（64KB），默认情况下所有的代码和数据都是近地址的。

   FAT内存模式：保护模式，代码和数据使用32位偏移，所有的代码和数据都在一个32位段中。

9. 声明一个名为MulArray的过程，接收一个指向字节数组的指针和一个指向双字数组的指针，此外还要接收第三个表示数组元素数目的参数。

   MulArray PROC array1:PTR WORD,array2:PTR DOWRD,num:WORD

   MulArray ENDP

10. 找到程序Task_Lock.EXE的注册码（提示：可以采用反汇编的方法，阅读汇编代码，得到答案）。要求

    a 写出得到最终答案的各个步骤，并做一定的描述；

    (a) 先直接启动Task_Lock.EXE，随意输入一个注册码，尝试注册一下，看有什么反应，笔者直接输入了“123456” 点击“OK”之后，返回的结果是“registration number incorrect”

    (b) 启动 OllyDbg，选择菜单 File → Open 载入 Task_Lock.EXE 文件，然后按 F9 调试运行，同样点击 Register 输入 “123456”，点击“OK”，还是会弹出之前同样的错误：

    (c) 然后在反汇编的窗口中，右击鼠标，选择 search for → All referenced strings，就可以看到下面的窗口： 在这个 Text strings 的窗口内按“Ctrl”+”F”键搜索 “registration number incorrect”字符串，发现字符串是在 00402C06 这个地址的命令操作执行的，双击之，反汇编窗口定位到相应地址。

    (d) 在这一行按 F2 添加一个断点，重新调试运行程序，输入“123456” 注册，发现程序在此处暂停。 观察右上角的寄存器的值，特别是 ECX 与 EDX，EDX 为我们输入的注册码，而 ECX 是 8765468413464354654，会不会这就是咱们要的注册码呢？不管这么多，拿来试试先。

    (e) 不试不知道，一试吓一跳，真是咱们要找的注册码。梦里寻它千百度,蓦然回首，原来它就藏在 ECX 处。

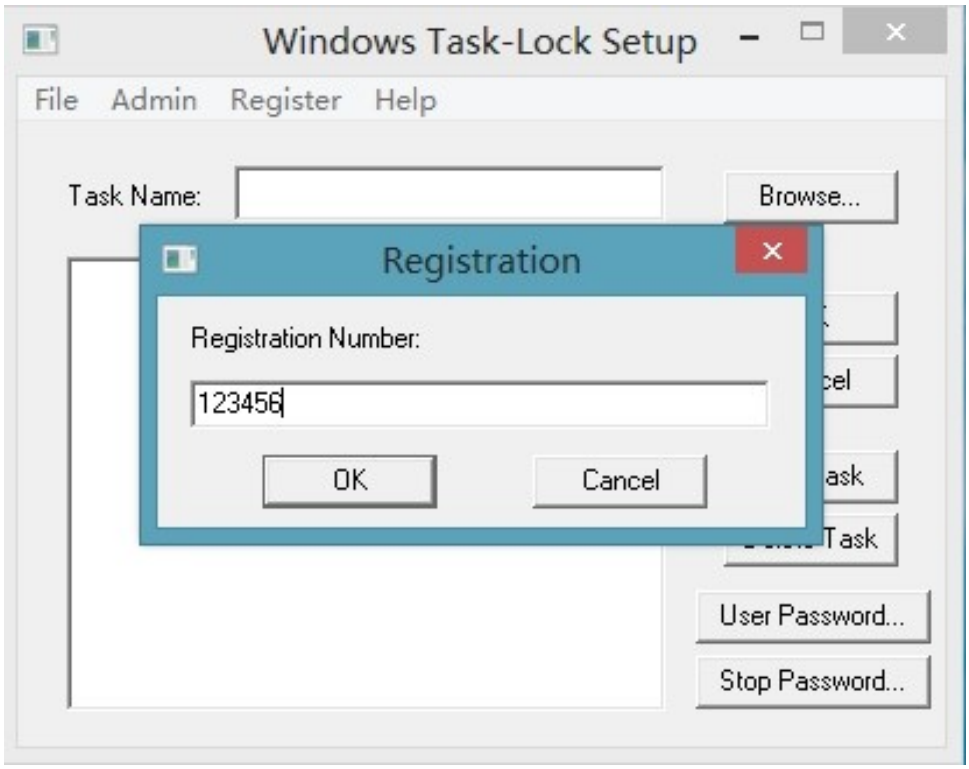
Windows Task-Lock Setup
File Admin Register Help
Task Name:
Browse...
Registration
Registration Number:
123456
OK
Cancel
User Password...
Stop Password...

图 1:


Windows Task-Lock Setup
File Admin Register Help
Windows Task-Lock
Registration number incorrect
确定
User Password...
Stop Password...

图 2:

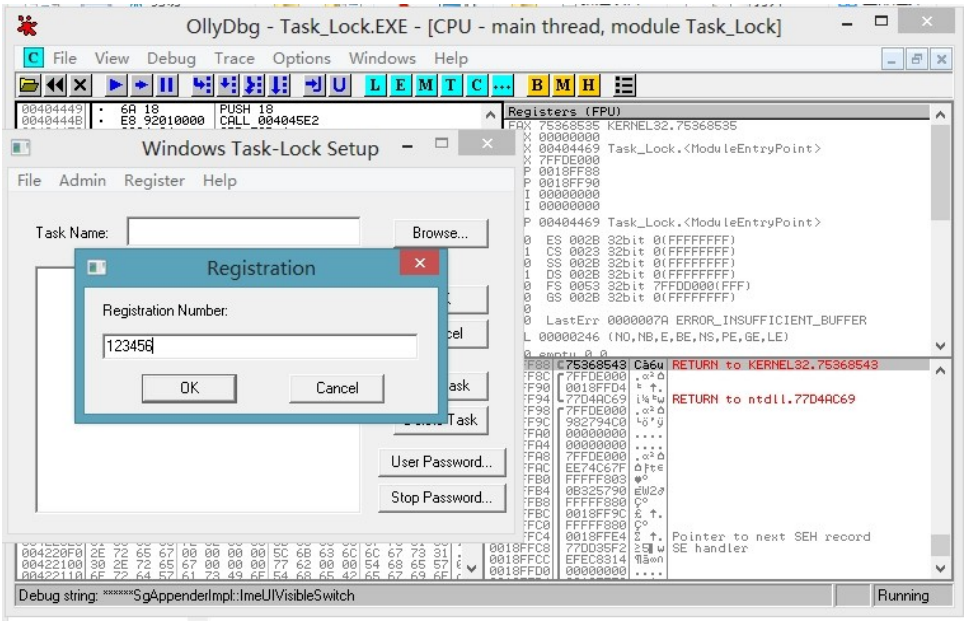

图 3:

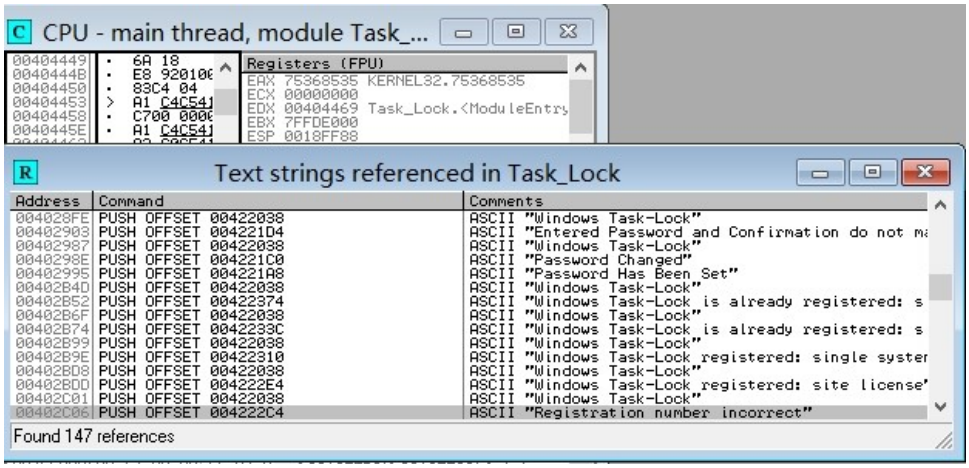

图 4:

b 写出程序对注册码正确性的检验方法；

既然还要写出验证方法，那么我们则需要对整个验证流程要再走一遍，于是我先尝试去找验证注册码的函数入口，向上不断寻找发现了这样一行（图8）：

这不是 Win32 的系统调用获取对话框中文本框的值么，那么验证注册码应该在获取了值之后，所以在这行代码的下一行设置一个断点，一步步 F8 调试运行。（图9）

多次调试运行后发现，原来验证码不止一个，有两个。分别存在 local.39 和 local.26，其中 local.39 处存放的是 6543545735465657712，而 local.26 处存放的是 8765468413464354654。还有一个位置 local.13 存放的是

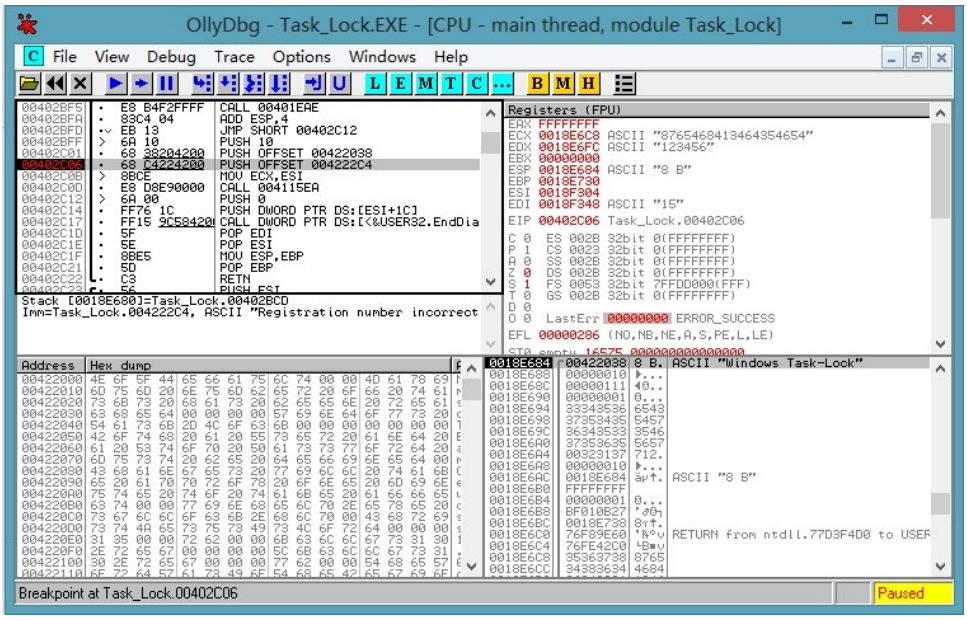

图 5:

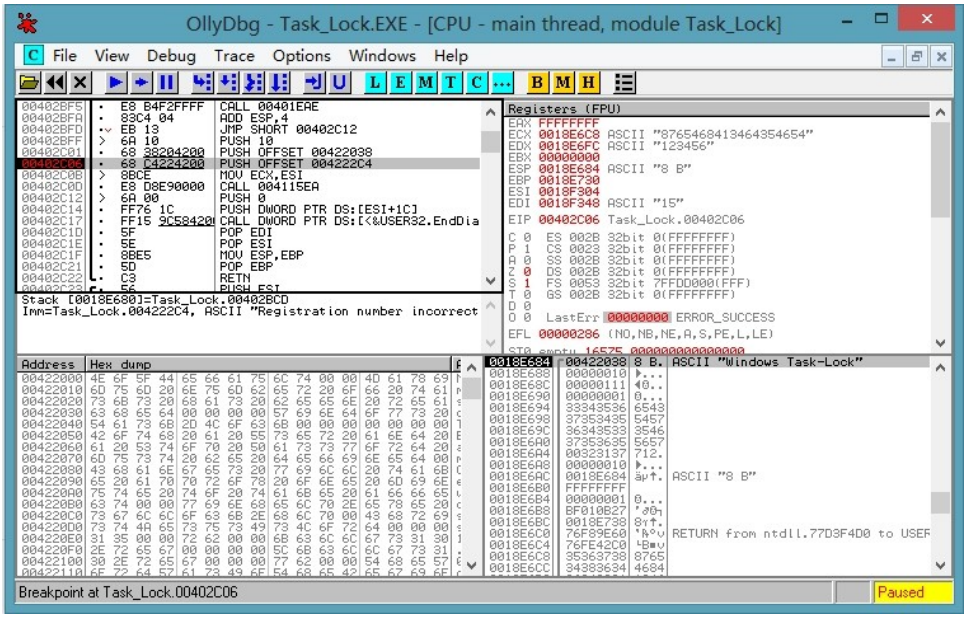

图 6:

我们输入的注册码，先与 6543545735465657712 比较，如果不匹配，则与 8765468413464354654 比较，根据输出提示可知，这应该是两种不同的注册码。

c 总结本次试验的体会。 刚开始不知道该如何下手，后面请教同学之后，推荐了下面的资料看了一下，很快就找到了注册码，感觉还是挺有意思的，虽然，后面具体注册码判断的细节就尝试了很多次才弄明白，不过整体感觉比较轻松的。

图 7:

图 8:

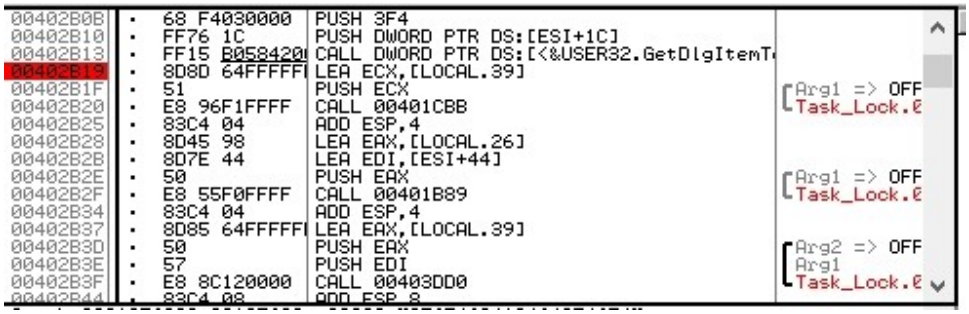

图 9:

d 参考资料：http://www.cnblogs.com/webman/archive/2007/10/08/916795.html