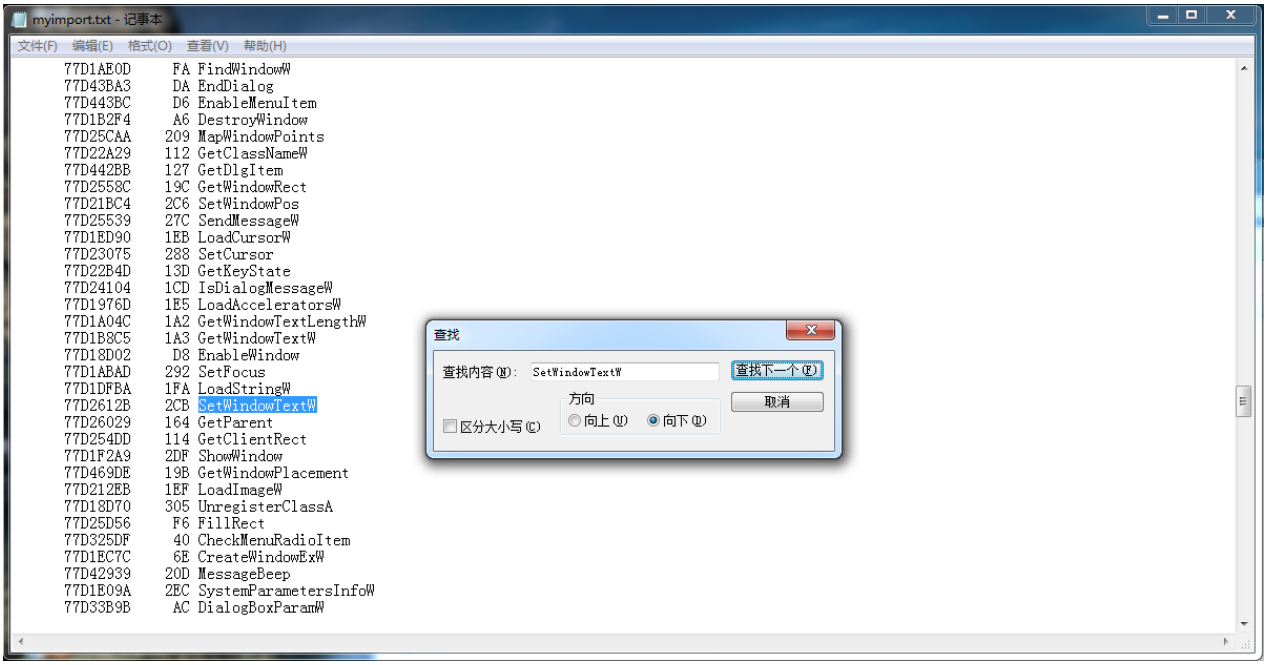
软件与系统安全：任务四

本文所有操作在windows7环境中运行，不同的操作系统环境会影响操作实验的成功

1.首先我们利用VS的开发者工具命令行对C:\Windows\SysWOW64文件夹中32位的计算器calc.exe进行dumpbin查看可执行文件的导入表

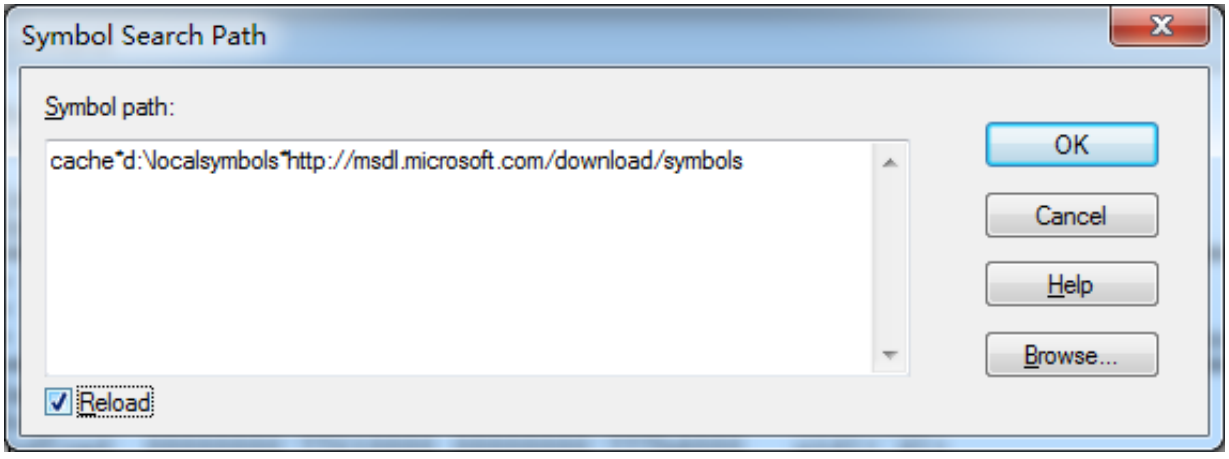
```
c:\Windows\SysWOW64>dumpbin /imports calc.exe>D:\oooaaa\myimport.txt
```

2.在win7环境下我们可以在文件中通过搜索发现，系统通过调用setWindowTextW系统API来在屏幕上显示计算器中的所有按键以及屏幕上显示的数字



win10环境下在该导入表中未搜索到该API函数，使用搜索引擎未找到合适的解释，推测是win10将该函数封装在某一动态数据库中了

3.利用windbg32位打开前文中32位的计算器可执行程序exe，试图在setWindowTextW处下断点`bp setWindowTextW`，但发现返回ERROR：无法解析的外部符号；发现是符号表缺失，只能通过外部手动导入的方式解决。



通过windbg symbol path关键字搜索，在文档中找到对应的源路径，然后在windbg中更改符号表路径

For example, the following command tells the debugger to use a symbol server to get symbols from the store at https://msdl.microsoft.com/download/symbols and cache the symbols in the `c:\MySymbols` directory.

dbgcmd Copy

```
.sympath cache*c:\MySymbols;srv*https://msdl.microsoft.com/download/symbo
```

不能使用64位的windbg进行实验，因为会一直无法读入符号表，至今原因不明

4.更改符号表源路径之后通过.reload /f /i 载入更新所有需要的pcb，之后可以正常对API函数下断点，对TextOutW和TextOutA也要同时下断点

```
0:000> .reload /f /i
```

```
0:000> .reload
Reloading current modules
.....................
0:000> bp SetWindowTextW
breakpoint 0 redefined
breakpoint 1 redefined
0:000> bl
 0 e 75611b4c     0001 (0001)  0:**** USER32!SetWindowTextW
0:000> bp TextOutW
0:000> bp TextOutA
0:000> cl
*** ERROR: Symbol file could not be found.  Defaulted to expc
Couldn't resolve error at 'l'
0:000> bl
 0 e 75611b4c     0001 (0001)  0:**** USER32!SetWindowTextW
 1 e 7548d41c     0001 (0001)  0:**** GDI32!TextOutW
 2 e 7548eda3     0001 (0001)  0:**** GDI32!TextOutA
```

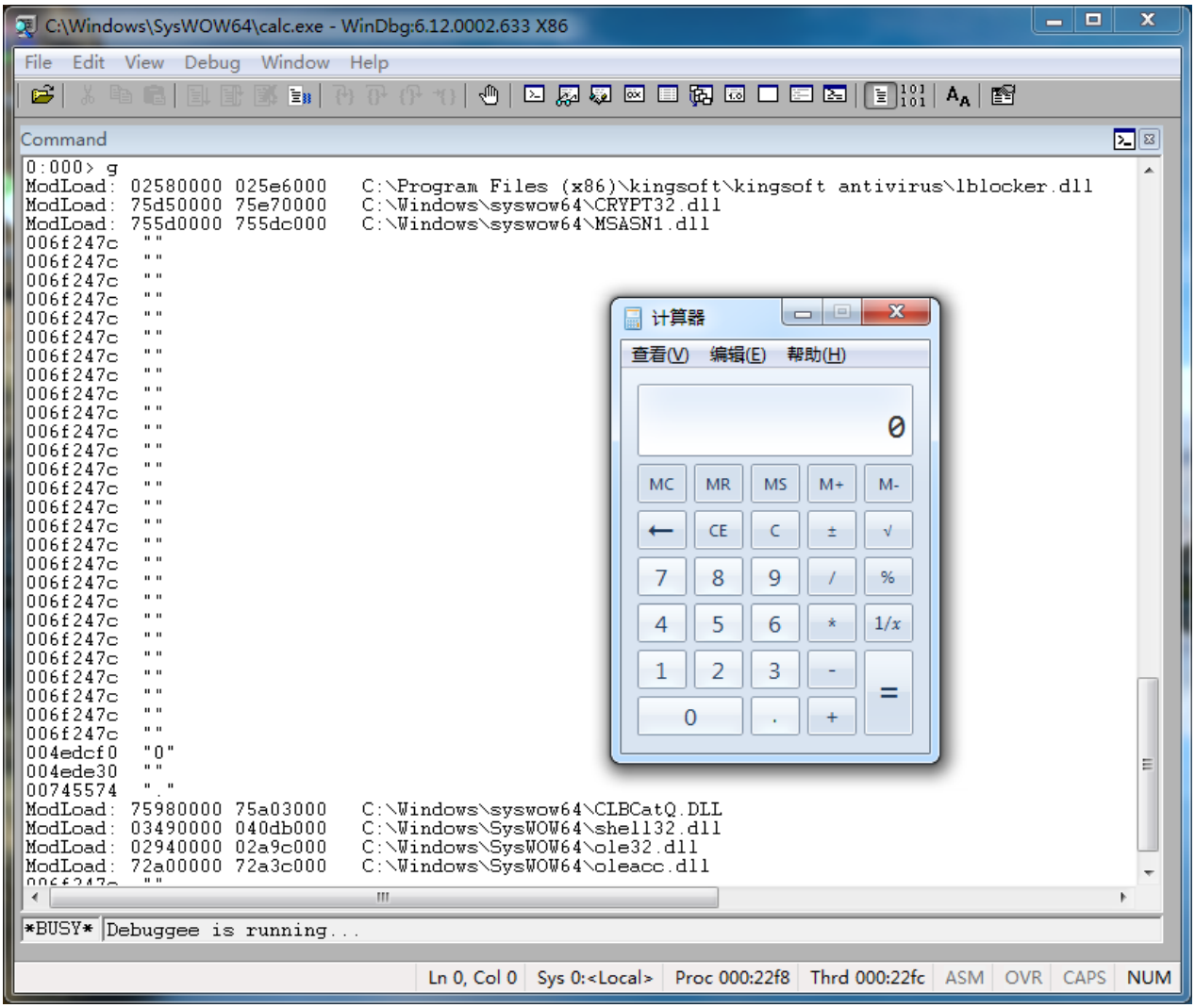
5.g指令运行之后发现击中了一个setWindowTextW，通过dd查看当前地址情况，由于我们根据原函数的参数表有两个参数可推出，esp+8处的地址的指针指向的应该是将要显示在屏幕上的内容，为了验证该想法，我们先bc取消原来设在setWindowTextW处的断点后重新设断点，指令为`bp setWindowTextW"du poi(esp+0x8);g"` poi为取地址操作

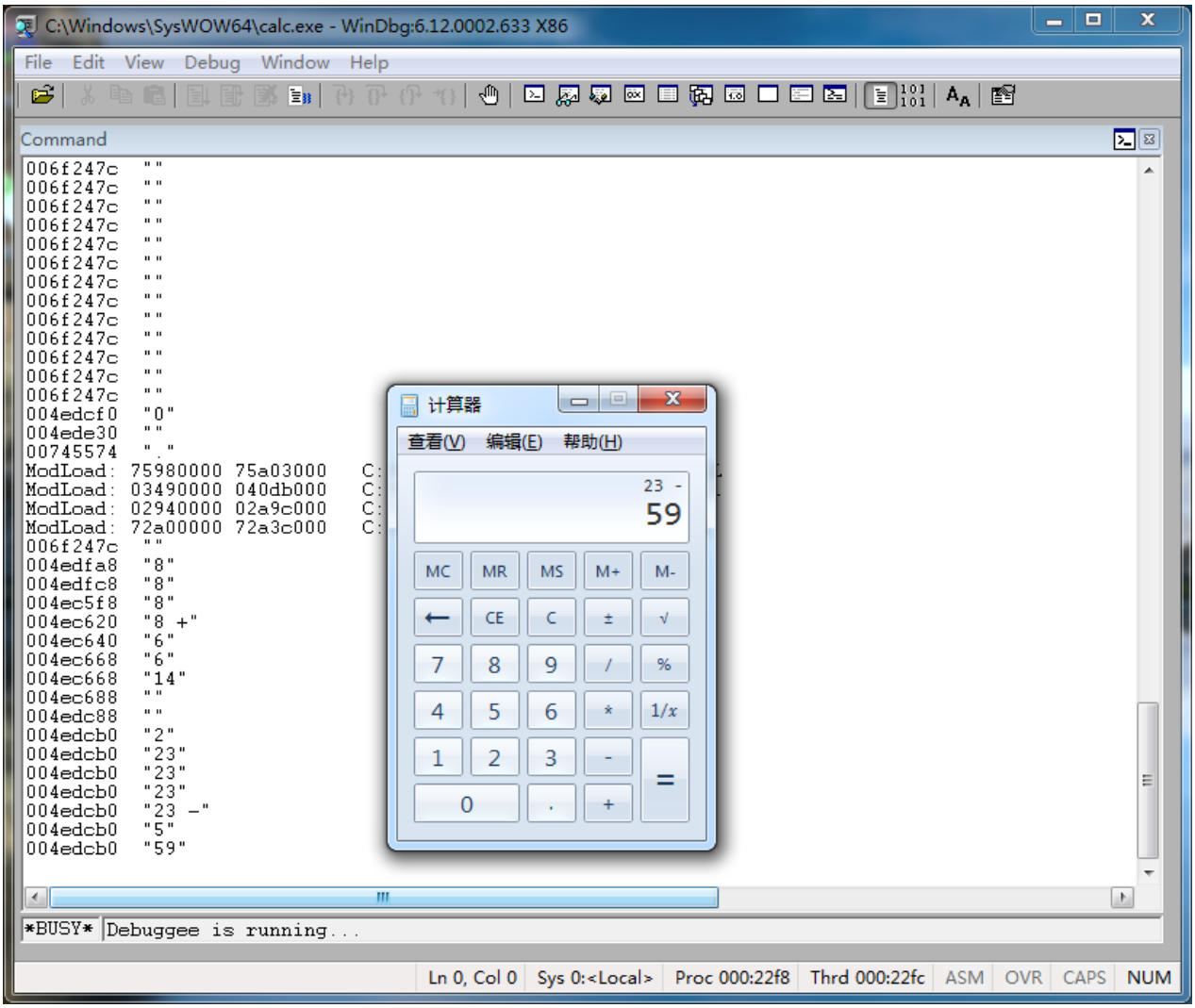
```
0:000> dd esp
000cf8b0  77950203 00000000 00000000 7efde000
000cf8c0  00000000 000cf8b0 77861985 000cfa98
000cf8d0  77861985 00184717 00000000 000cfa58
000cf8e0  77871617 7efdd000 7efde000 778f206c
000cf8f0  00000000 00000000 00000000 00000000
000cf900  00000000 00000000 00000000 00000000
000cf910  00000000 00000000 00000000 00000030
000cf920  00000000 00000000 00000000 00000000
```

```
0:000> g
ModLoad: 75a80000 75ae0000   C:\Windows\SysWOW64\IMM32.DLL
ModLoad: 75800000 758cc000   C:\Windows\syswow64\MSCTF.dll
ModLoad: 730d0000 731cb000   C:\Windows\SysWOW64\WindowsCodecs.dll
ModLoad: 01dc0000 01e40000   C:\Windows\SysWOW64\uxtheme.dll
ModLoad: 01dc0000 01e40000   C:\Windows\SysWOW64\uxtheme.dll
ModLoad: 01f40000 01fc0000   C:\Windows\SysWOW64\uxtheme.dll
ModLoad: 01f40000 01fc0000   C:\Windows\SysWOW64\uxtheme.dll
ModLoad: 10000000 10153000   c:\program files (x86)\ksafe\ksfmon.dll
ModLoad: 75a10000 75a45000   C:\Windows\syswow64\WS2_32.dll
ModLoad: 755e0000 755e6000   C:\Windows\syswow64\NSI.dll
ModLoad: 02070000 020e8000   C:\Program Files (x86)\kingsoft\kingsoft antivirus\kwsui.dll
ModLoad: 75460000 75465000   C:\Windows\syswow64\PSAPI.DLL
ModLoad: 75100000 7510d000   C:\Windows\SysWOW64\WTSAPI32.dll
ModLoad: 74130000 74143000   C:\Windows\SysWOW64\dwmapi.dll
Breakpoint 0 hit
eax=220a1dcf ebx=001205d8 ecx=00670a70 edx=00000030 esi=004e4f40 edi=00000000
eip=75611b4c esp=000ce4d4 ebp=000ce514 iopl=0         nv up ei pl zr na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
USER32!SetWindowTextW:
75611b4c 8bff            mov     edi,edi
```

6.再输入g指令我们就可以看到弹出的计算器程序了，在计算器上输入数据的时候windbg也会捕获相应的操作详情

```
0:000> bc
0:000> bc SetWindowTextW
         ^ Syntax error in 'bc SetWindowTextW'
0:000> bc 0
0:000> bl
 1 e 7548d41c     0001 (0001)  0:**** GDI32!TextOutW
 2 e 7548eda3     0001 (0001)  0:**** GDI32!TextOutA
0:000> bp SetWindowTextW "du poi(esp+0x8);g"
0:000> bl
 0 e 75611b4c     0001 (0001)  0:**** USER32!SetWindowTextW "du poi(esp+0x8);g"
 1 e 7548d41c     0001 (0001)  0:**** GDI32!TextOutW
 2 e 7548eda3     0001 (0001)  0:**** GDI32!TextOutA
```


C:\Windows\SysWOW64\calc.exe - WinDbg:6.12.0002.633 X86
File Edit View Debug Window Help
Command
0:000> g
ModLoad: 02580000 025e6000 C:\Program Files (x86)\kingsoft\kingsoft antivirus\lblocker.dll
ModLoad: 75d50000 75e70000 C:\Windows\syswow64\CRYPT32.dll
ModLoad: 755d0000 755dc000 C:\Windows\syswow64\MSASN1.dll
006f247c ""
004edcf0 "0"
004ede30 ""
00745574 "."
ModLoad: 75980000 75a03000 C:\Windows\syswow64\CLBCatQ.DLL
ModLoad: 03490000 040db000 C:\Windows\SysWOW64\shell32.dll
ModLoad: 02940000 02a9c000 C:\Windows\SysWOW64\ole32.dll
ModLoad: 72a00000 72a3c000 C:\Windows\SysWOW64\oleacc.dll
计算器
查看(V) 编辑(E) 帮助(H)
0
MC MR MS M+ M-
CE C ± √
7 8 9 / %
4 5 6 * 1/x
1 2 3 -
0 . +
=
BUSY Debuggee is running...
Ln 0, Col 0 Sys 0:<Local> Proc 000:22f8 Thrd 000:22fc ASM OVR CAPS NUM



7.随后我们再次更改设断点的指令为`bp setWindowTextW"du poi(esp+0x8);eu poi(esp+0x8) \"99\";g"`,再次运行g时发现计算器的输入框虽然显示99但是整个程序却卡死崩溃了

```
0:007> bl
 0 e 75611b4c     0001 (0001)  0:**** USER32!SetWindowTextW "du poi(esp+0x8);g"
 1 e 7548d41c     0001 (0001)  0:**** GDI32!TextOutW
 2 e 7548eda3     0001 (0001)  0:**** GDI32!TextOutA
0:007> bc 0
0:007> bu SetWindowTextW "du poi(esp+0x8);eu poi(esp+0x8) \"999\";g"
0:007> bl
 0 e 75611b4c     0001 (0001)  0:**** USER32!SetWindowTextW "du poi(esp+0x8);eu poi(esp+0x8) \"999\";g"
 1 e 7548d41c     0001 (0001)  0:**** GDI32!TextOutW
 2 e 7548eda3     0001 (0001)  0:**** GDI32!TextOutA
```