Pattern-Oriented

SOFTWARE DIAGNOSTICS
DEBUGGING
MALWARE ANALYSIS
REVERSING

Sample Training Exercises

Facebook LinkedIn Twitter

Software Diagnostics Services

Training Courses

- Accelerated Windows Memory Dump Analysis
- Accelerated .NET Memory Dump Analysis
- Accelerated Mac OS X Core Dump Analysis
- Advanced Windows Memory Dump Analysis with Data Structures
- Advanced Windows RT Memory Dump Analysis, ARM edition
- Accelerated Windows Debugging[3]
- Accelerated Windows Malware Analysis with Memory Dumps
- Accelerated Disassembly, Reconstruction and Reversing

Training Packs

- Platform-Independent Crash Dump Analysis Training Pack
- Trace and Log Analysis Training Pack
- Windows Crash Dump Analysis Training Pack
- Windows Debugging Training Pack
- Windows Complete Memory Dump Analysis Training Pack
- Windows Memory Forensics Training Pack
- Enterprise Windows Software Diagnostics and Debugging Pack

Training Roadmap

Training Upgrades

Enterprise Windows Software Diagnostics and Debugging Pack

Windows Crash Dump Analysis Training Pack

- Accelerated Windows Memory Dump Analysis
- Advanced Windows Memory Dump Analysis

Trace and Log Analysis Training Pack

- Accelerated Windows Software Trace Analysis

Windows Debugging Training Pack

- Accelerated Windows Debugging[3]
- Accelerated Windows Memory Dump Analysis
- Accelerated .NET Memory Dump Analysis

Windows Memory Forensics Training Pack

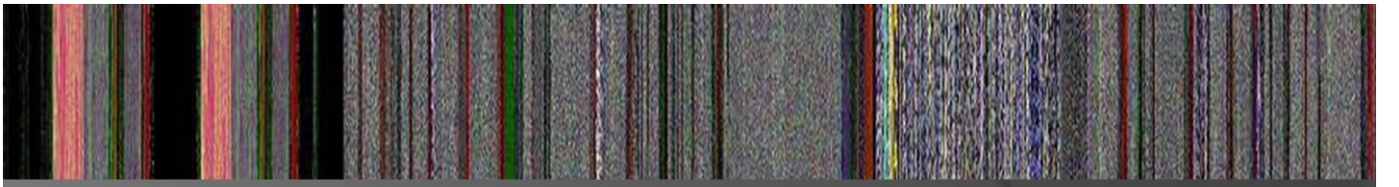
- Advanced Windows Memory Dump Analysis
- Advanced Windows RT Memory Dump Analysis
- Accelerated Windows Malware Analysis
- Accelerated Disassembly, Reconstruction and Reversing

Windows Complete Memory Dump Analysis Training Pack

Windows Crash Dump Analysis Training Pack

- Accelerated Windows Memory Dump Analysis
- Advanced Windows Memory Dump Analysis

- Accelerated .NET Memory Dump Analysis



Windows Memory Dump Analysis Accelerated

Version 3.0

Dmitry Vostokov
Software Diagnostics Services

Published by OpenTask, Republic of Ireland

OpenTask books and magazines are available through booksellers and distributors worldwide. For further information or comments send requests to press@opentask.com.

A CIP catalogue record for this book is available from the British Library.

ISBN-l3: 978-09558328-2-6 (Paperback)

Version 3, 2014

Contents

Exercise P1: Analysis of a normal application process dump (32-bit notepad)

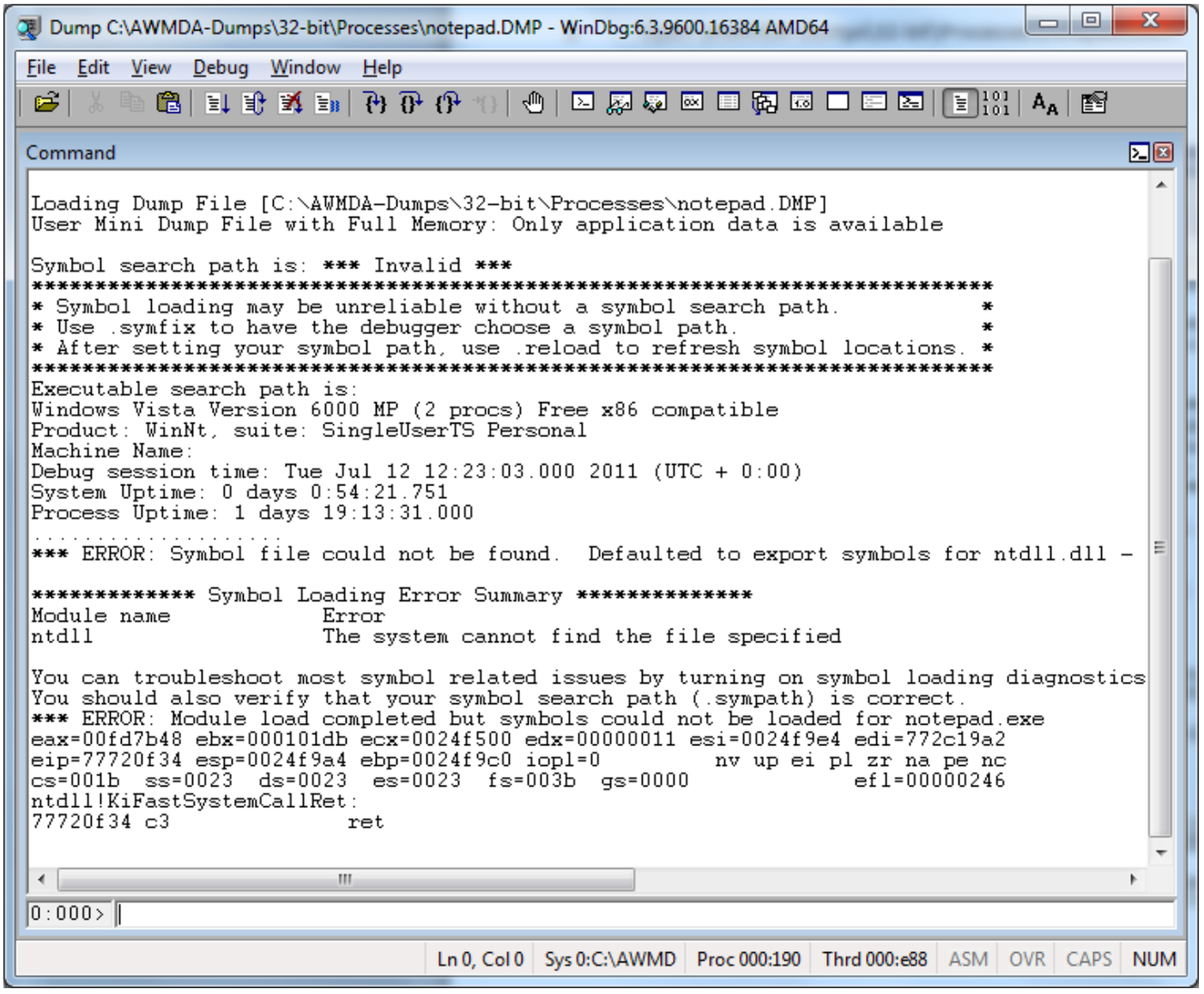
Goal: Learn how to see dump file type and version, get a stack trace, check its correctness, perform default analysis, list modules, check their version information, check process environment.

Patterns: Manual Dump; Stack Trace; Not My Version; Environment Hint

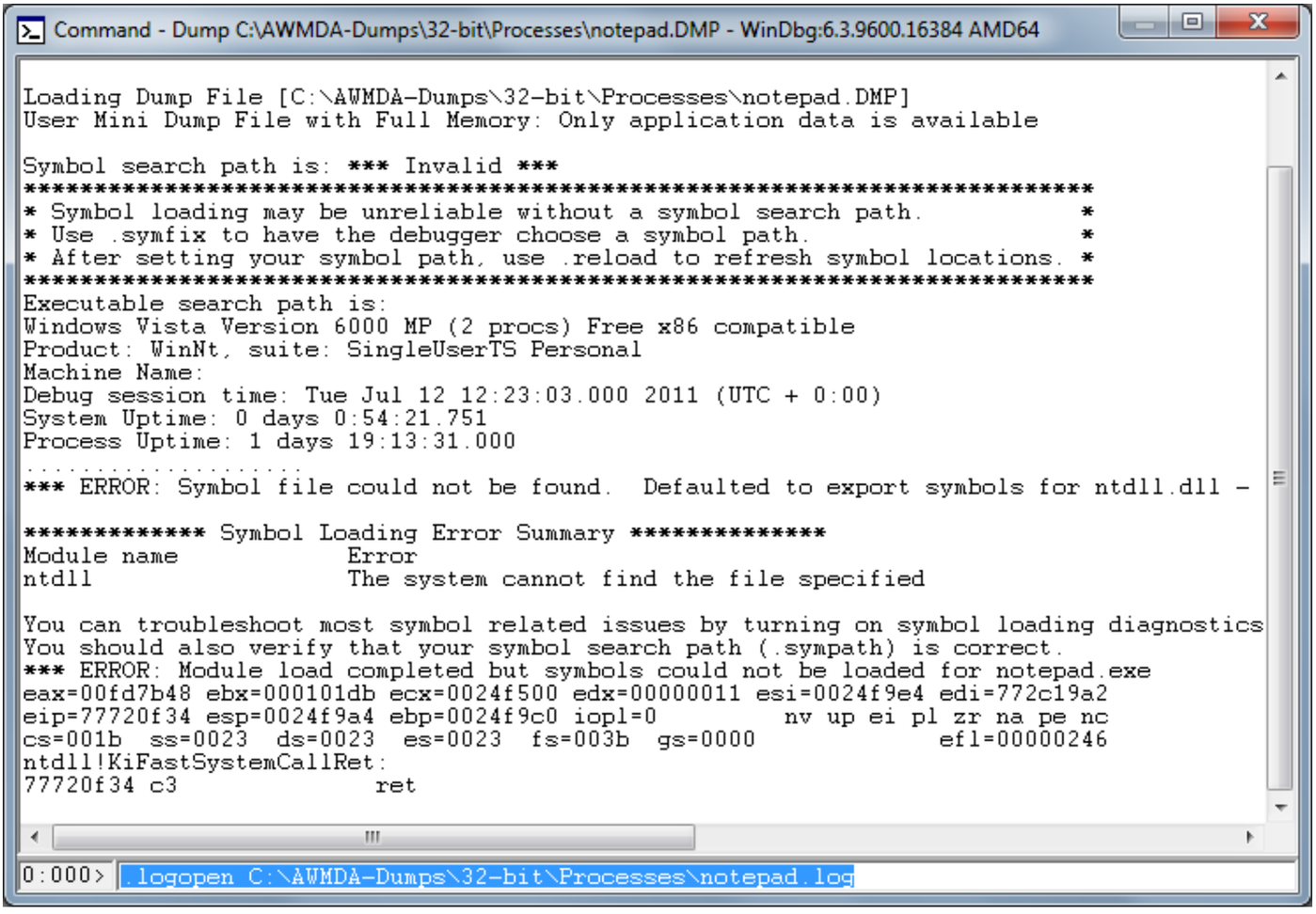
1. Launch WinDbg from Windows Kits \ Debugging Tools for Windows (X64).

2. Open \AWMDA-Dumps\32-bit\Processes\notepad.DMP.

3. We get the dump file loaded:

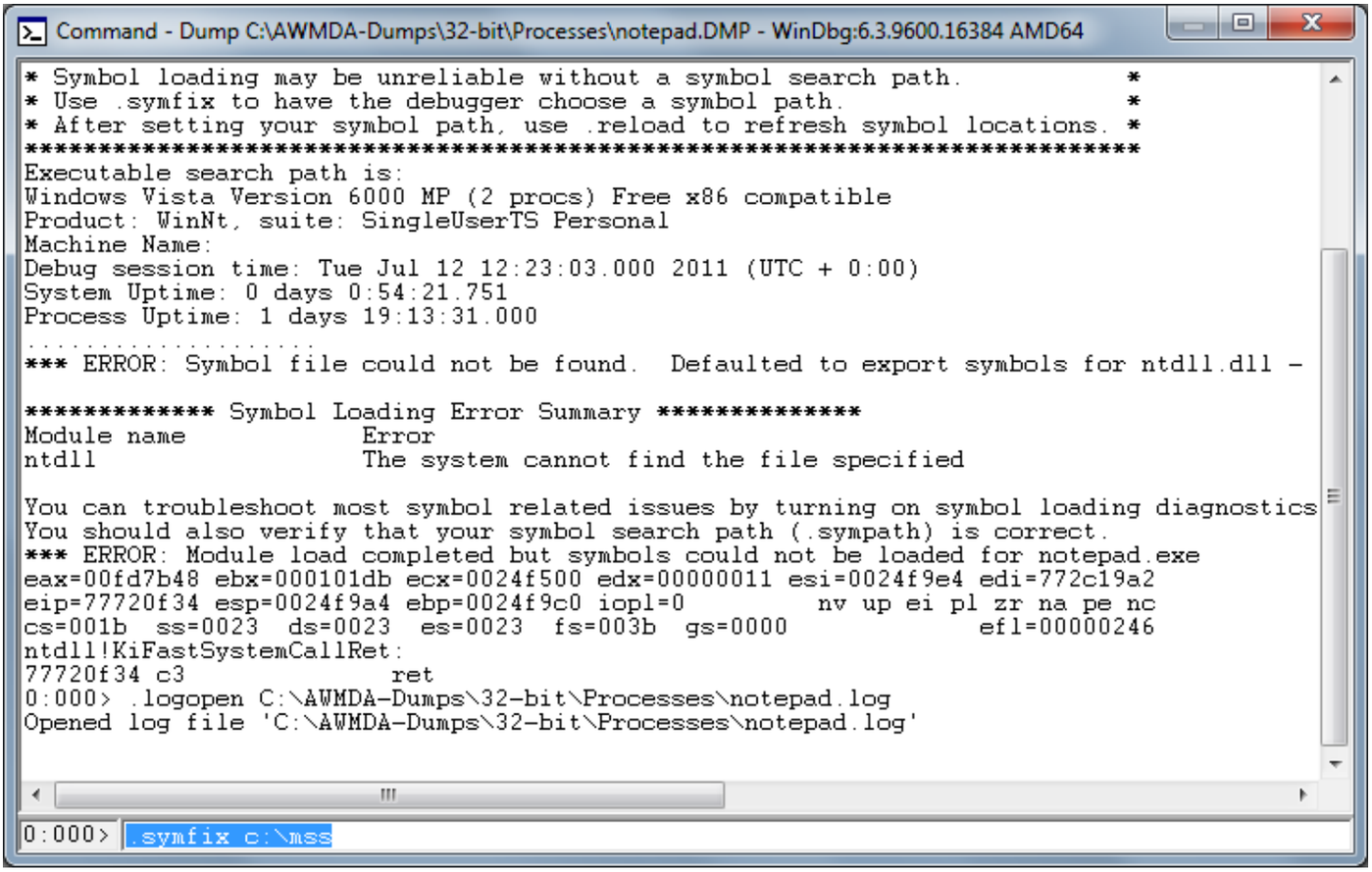


4. Open a log file to save all future output using **.logopen** command:

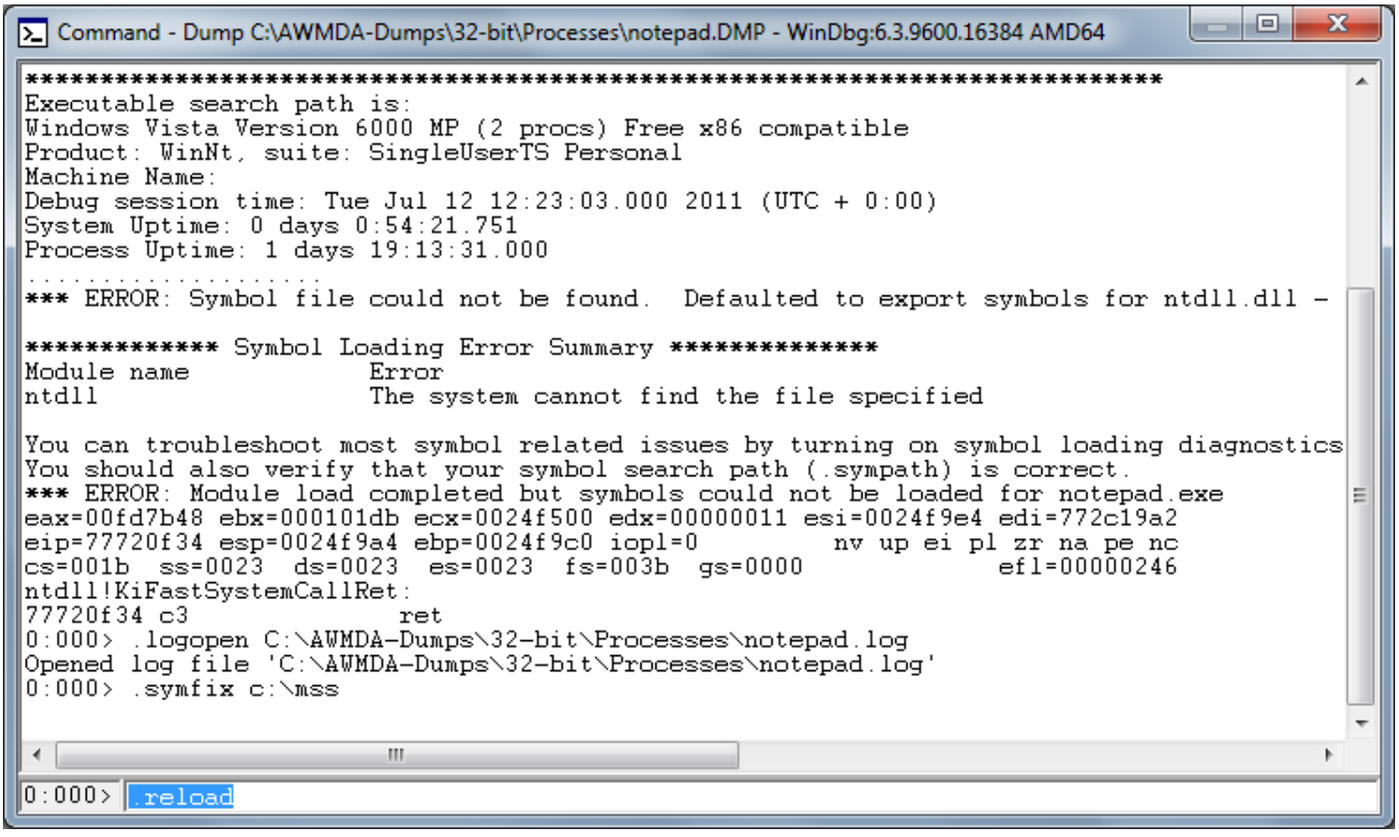


Note: You can type any comment by using ***** command.

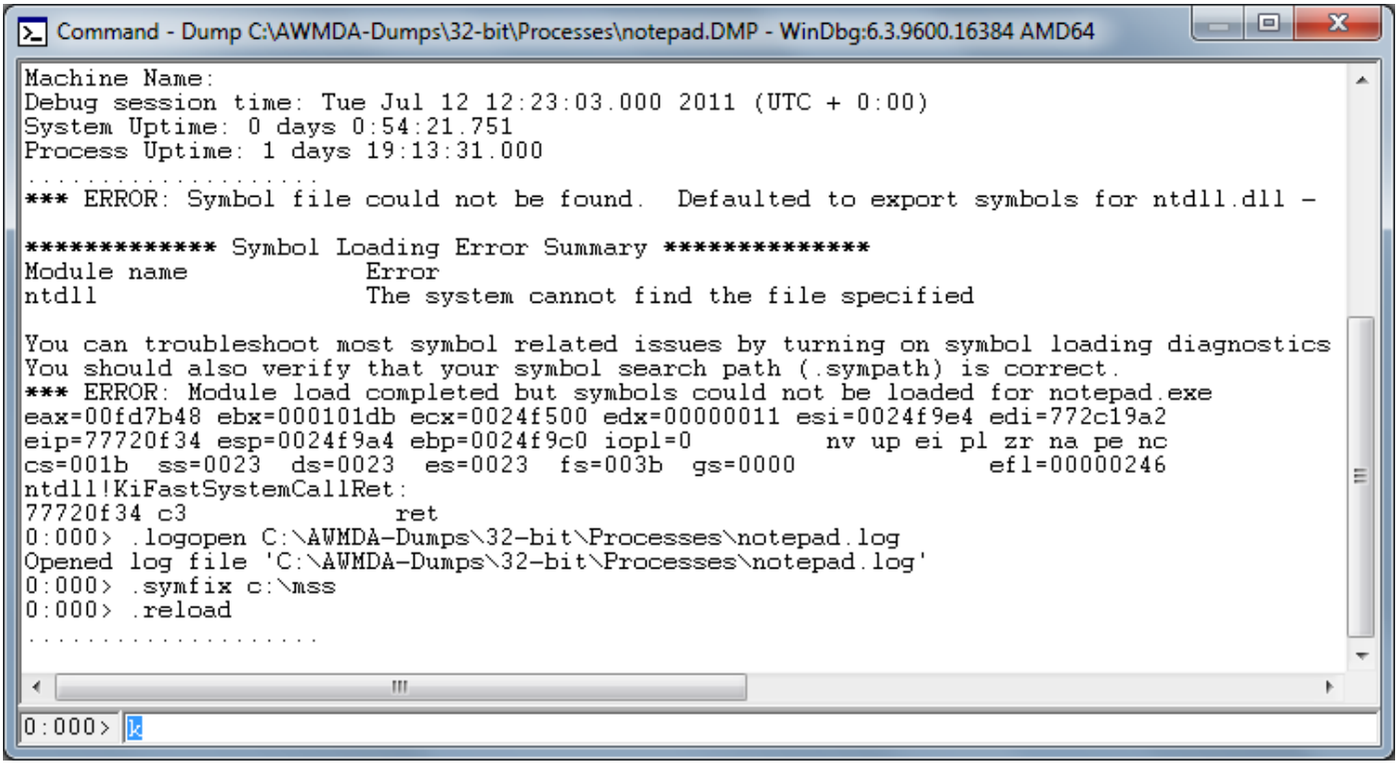
5. Type the command **.symfix c:\mss** to set a path to download symbol files from Microsoft symbol file server:

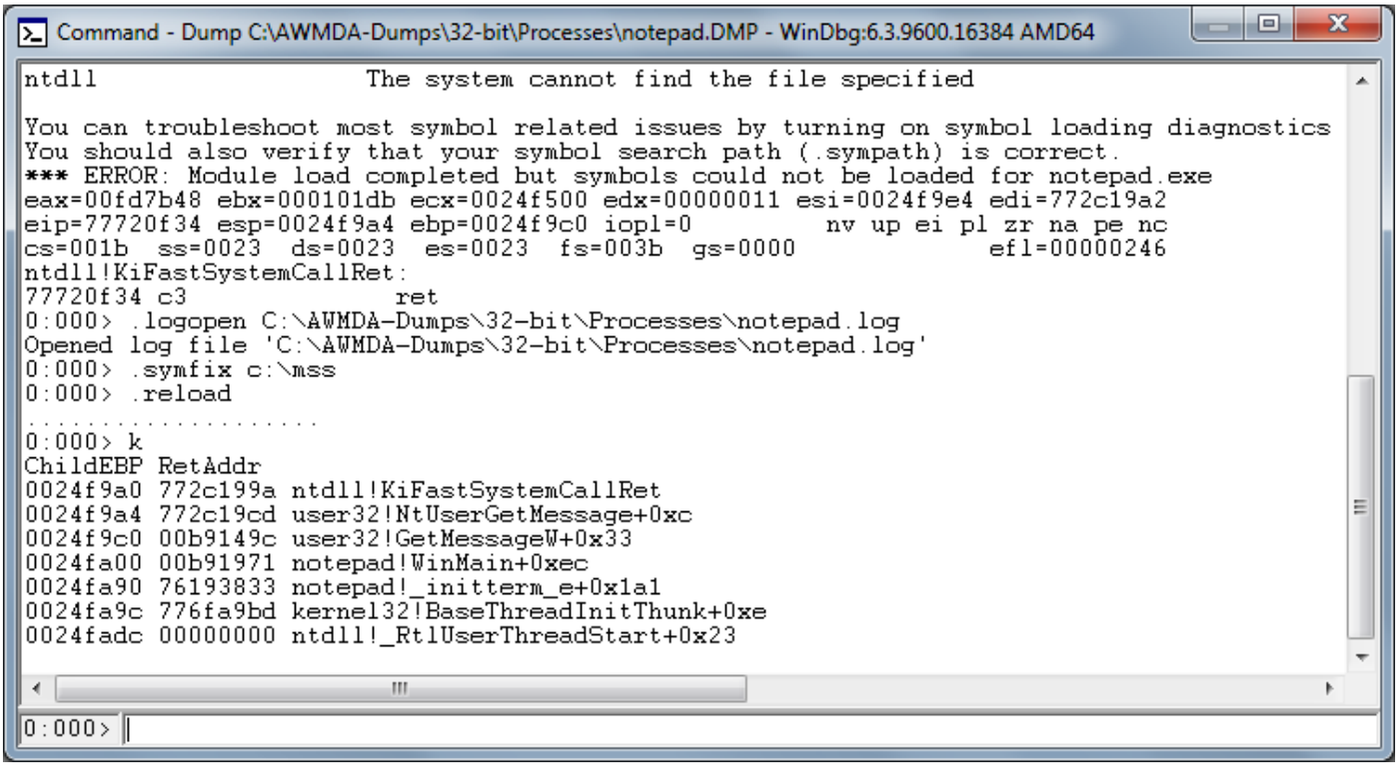


6. Type **.reload** command to download symbols if necessary:

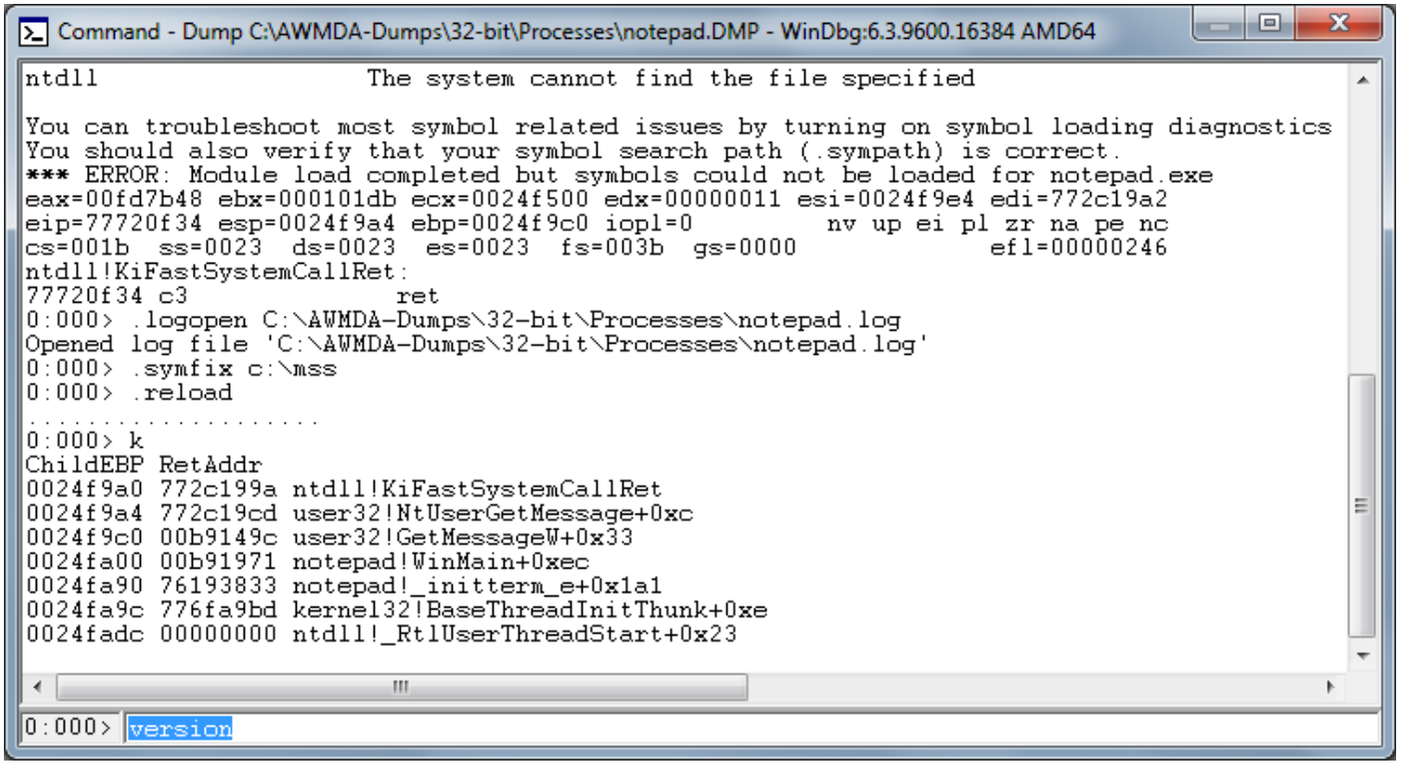


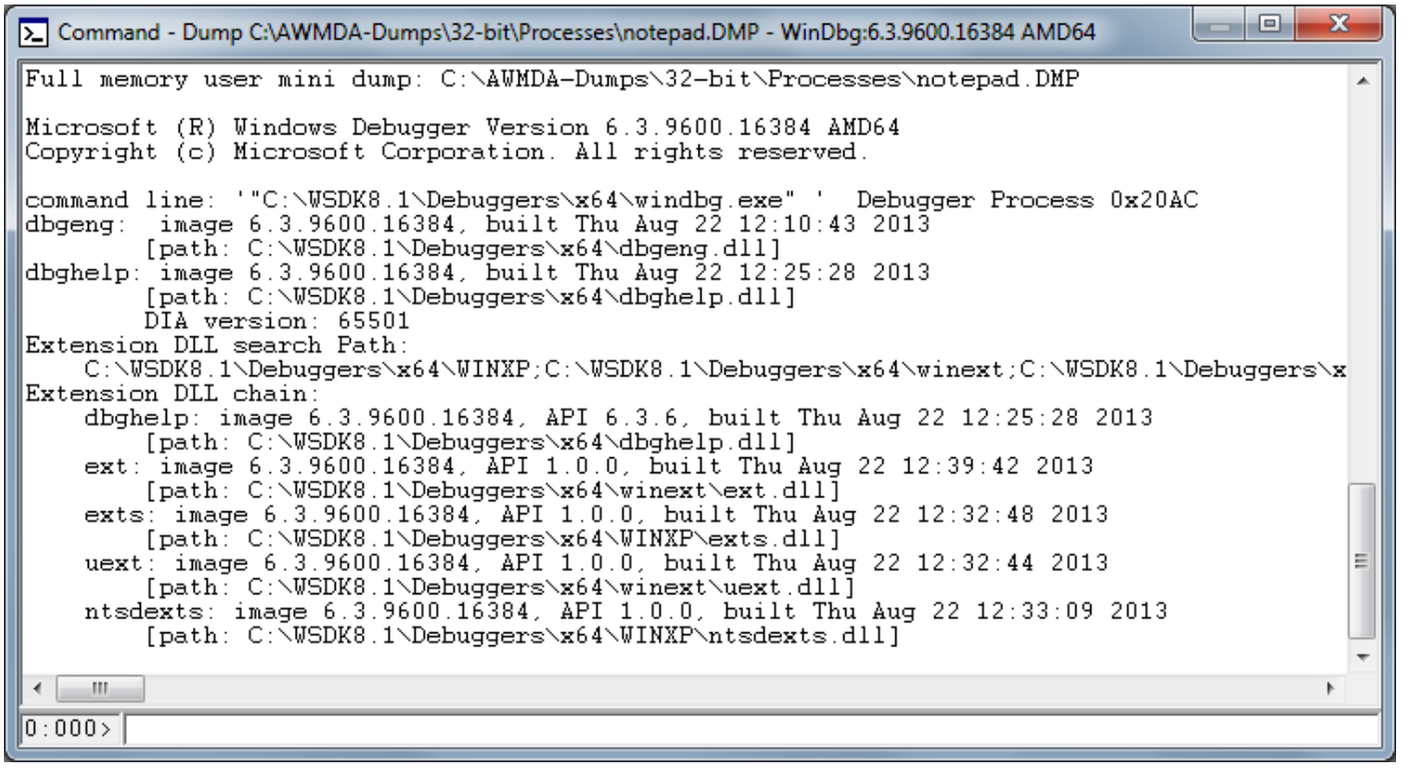
7. Type **k** command to verify the correctness of the stack trace:





8. Type **version** command to get OS version, system and process uptimes, the dump file timestamp and its type:



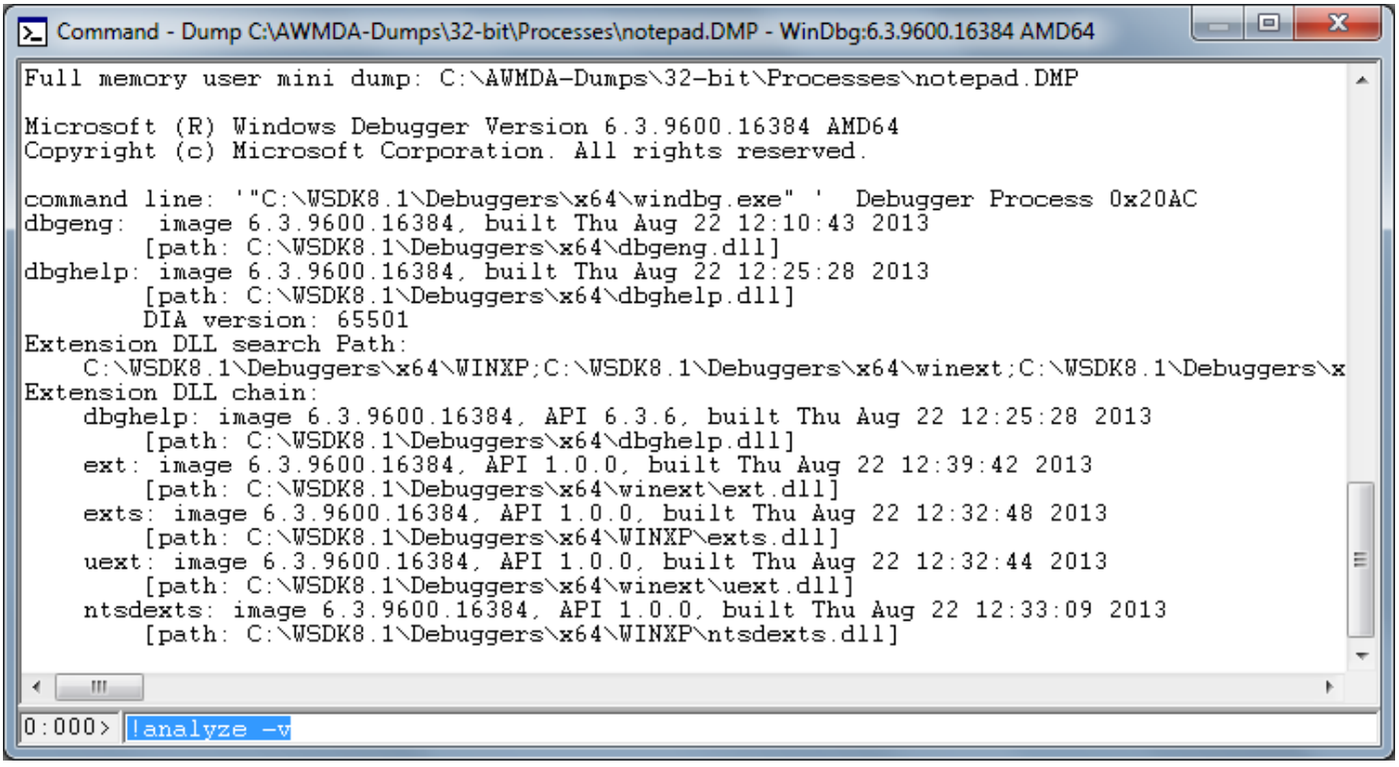


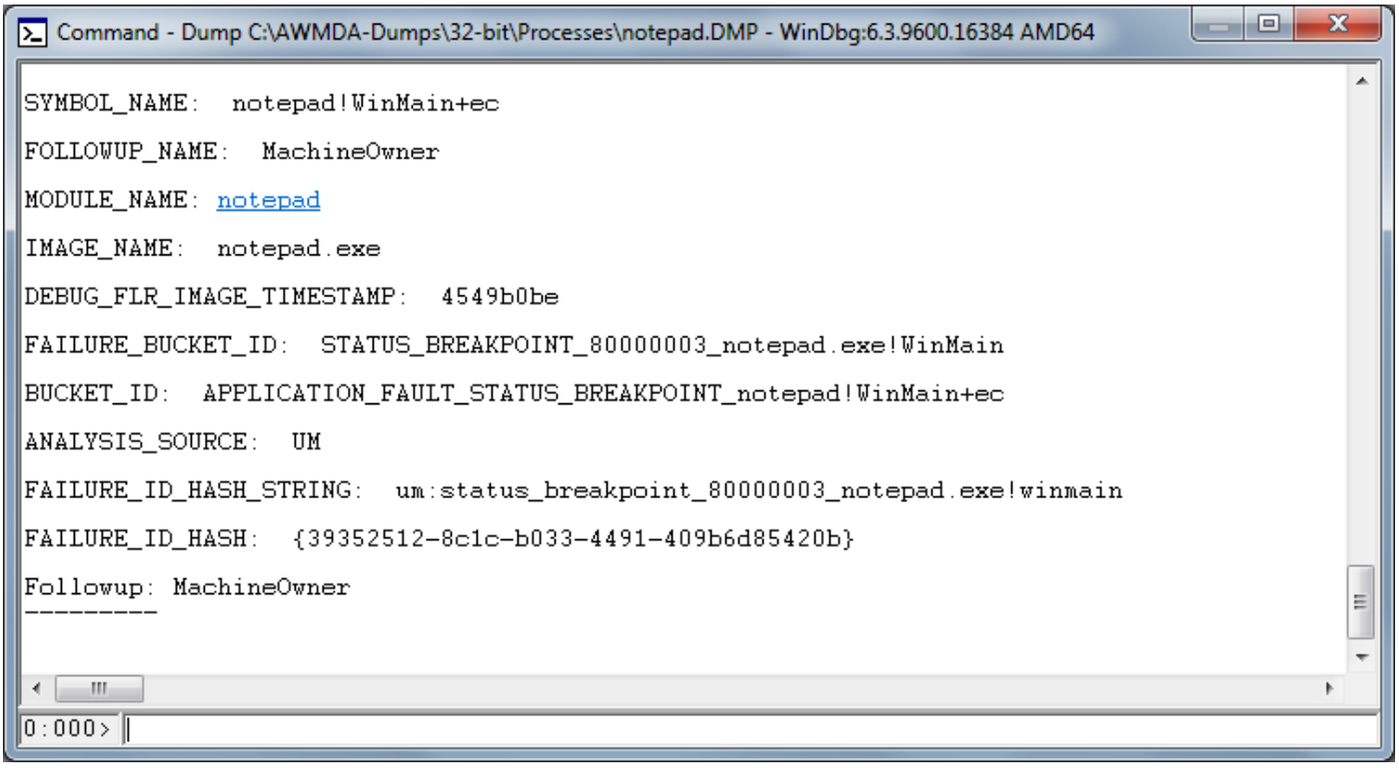
Note: This is the full output:

```
0:000> version
Windows Vista Version 6000 MP (2 procs) Free x86 compatible
Product: WinNt, suite: SingleUserTS Personal
kernel32.dll version: 6.0.6000.16386 (vista_rtm.061101-2205)
Machine Name:
Debug session time: Tue Jul 12 12:23:03.000 2011 (UTC + 0:00)
System Uptime: 0 days 0:54:21.751
Process Uptime: 1 days 19:13:31.000
  Kernel time: 0 days 0:00:00.000
  User time: 0 days 0:00:00.000
Full memory user mini dump: C:\AWMDA-Dumps\32-bit\Processes\notepad.DMP

Microsoft (R) Windows Debugger Version 6.3.9600.16384 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.

command line: '"C:\WSDK8.1\Debuggers\x64\windbg.exe" '  Debugger Process 0x23D8
dbgeng:  image 6.3.9600.16384, built Thu Aug 22 12:10:43 2013
        [path: C:\WSDK8.1\Debuggers\x64\dbgeng.dll]
dbghelp: image 6.3.9600.16384, built Thu Aug 22 12:25:28 2013
        [path: C:\WSDK8.1\Debuggers\x64\dbghelp.dll]
        DIA version: 65501
Extension DLL search Path:

C:\WSDK8.1\Debuggers\x64\WINXP;C:\WSDK8.1\Debuggers\x64\winext;C:\WSDK8.1\Debuggers\x64\winext\
arcade;C:\WSDK8.1\Debuggers\x64\pri;C:\WSDK8.1\Debuggers\x64;C:\WSDK8.1\Debuggers\x64\winext\ar
cade;C:\Program Files\Common Files\Microsoft Shared\Windows Live;C:\Program Files (x86)\Common
Files\Microsoft Shared\Windows Live;C:\Program Files (x86)\Intel\iCLS Client\;C:\Program
Files\Intel\iCLS
Client\;C:\windows\system32;C:\windows;C:\windows\System32\Wbem;C:\windows\System32\WindowsPowe
rShell\v1.0\;C:\Program Files\Intel\Intel(R) Management Engine Components\DAL;C:\Program
Files\Intel\Intel(R) Management Engine Components\IPT;C:\Program Files (x86)\Intel\Intel(R)
Management Engine Components\DAL;C:\Program Files (x86)\Intel\Intel(R) Management Engine
Components\IPT;C:\Program Files (x86)\Intel\OpenCL SDK\2.0\bin\x86;C:\Program Files
(x86)\Intel\OpenCL SDK\2.0\bin\x64;C:\Program Files\Intel\WiFi\bin\;C:\Program Files\Common
Files\Intel\WirelessCommon\;C:\Program Files (x86)\Windows Live\Shared;C:\Program
Files\Microsoft\Web Platform Installer\;C:\Program Files (x86)\Microsoft ASP.NET\ASP.NET Web
Pages\v1.0\;C:\Program Files\Microsoft SQL Server\110\Tools\Binn\;C:\WSDK8.1\Windows
Performance Toolkit\
Extension DLL chain:
    dbghelp: image 6.3.9600.16384, API 6.3.6, built Thu Aug 22 12:25:28 2013
        [path: C:\WSDK8.1\Debuggers\x64\dbghelp.dll]
    ext: image 6.3.9600.16384, API 1.0.0, built Thu Aug 22 12:39:42 2013
        [path: C:\WSDK8.1\Debuggers\x64\winext\ext.dll]
    exts: image 6.3.9600.16384, API 1.0.0, built Thu Aug 22 12:32:48 2013
        [path: C:\WSDK8.1\Debuggers\x64\WINXP\exts.dll]
    uext: image 6.3.9600.16384, API 1.0.0, built Thu Aug 22 12:32:44 2013
        [path: C:\WSDK8.1\Debuggers\x64\winext\uext.dll]
    ntsdexts: image 6.3.9600.16384, API 1.0.0, built Thu Aug 22 12:33:09 2013
        [path: C:\WSDK8.1\Debuggers\x64\WINXP\ntsdexts.dll]
```

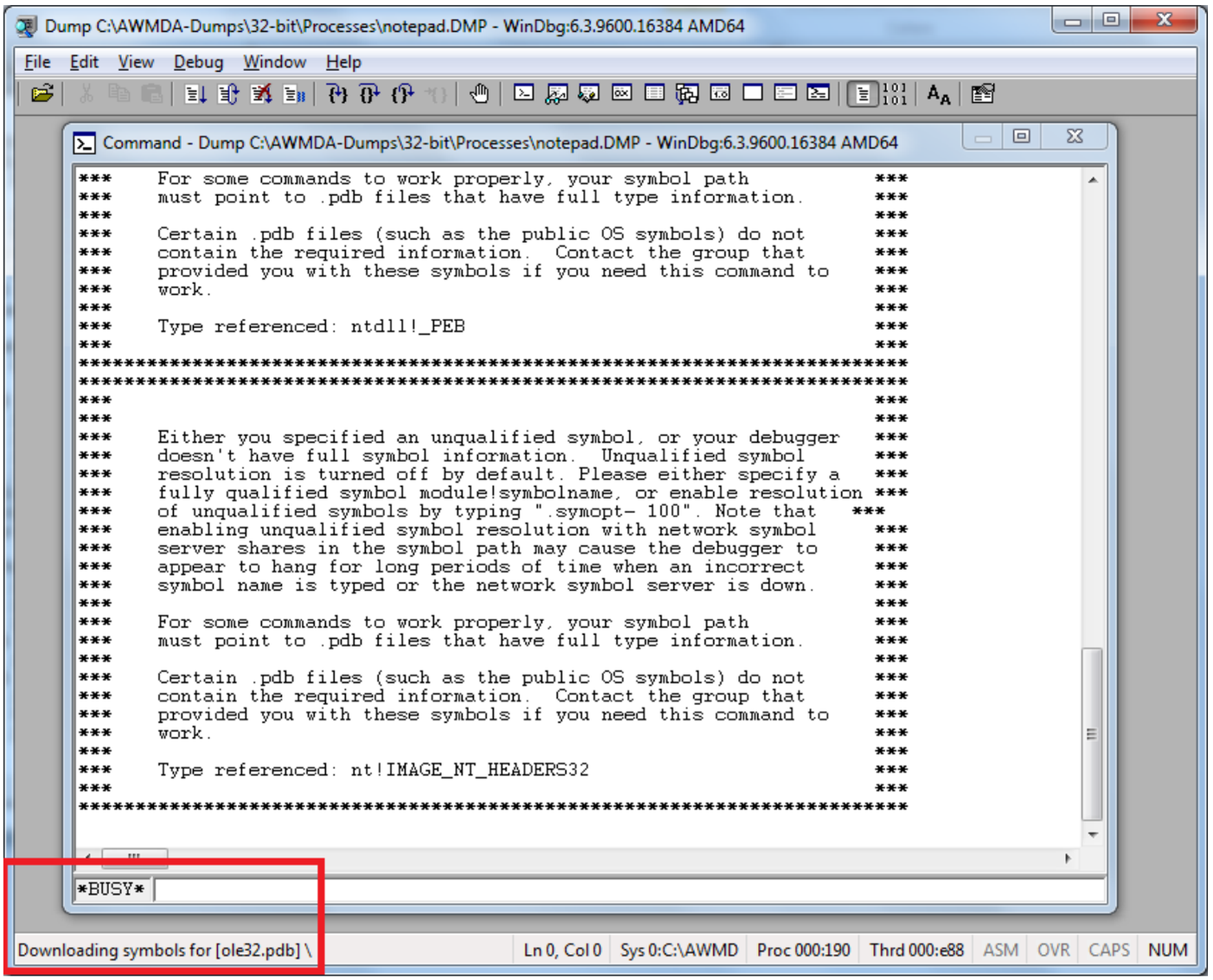
Note: Debug session time is when the dump was generated. Although the dump is called mini dump it is a full memory user dump with all process memory included.

9. Type the default analysis command **!analyze -v**:





Note: This (or**.reload** command) may take some time initially as symbols are downloaded from the symbol server:



10. Let's now look at the output in more detail:

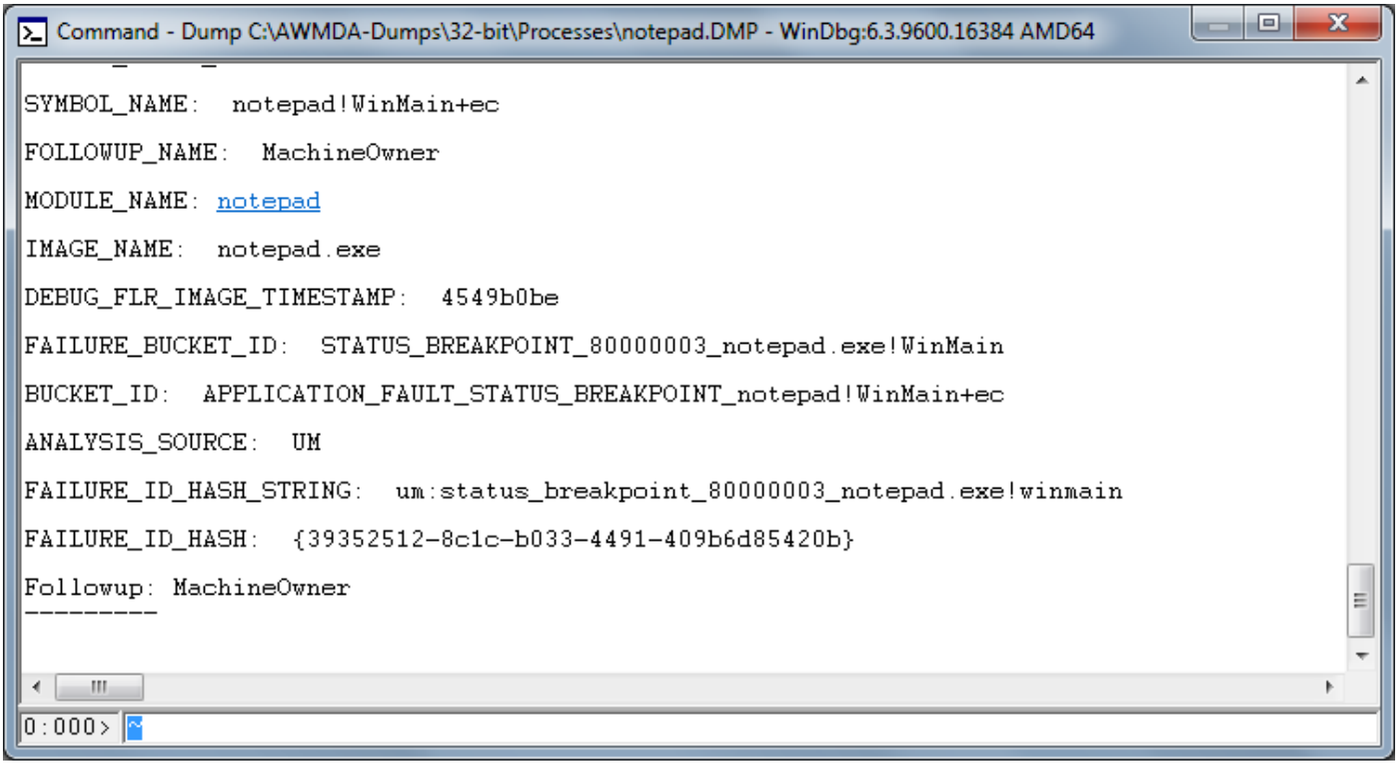
```
                                /* start of WinDbg output */

0:000> !analyze -v
*******************************************************************************
*                                                                             *
*                        Exception Analysis                                   *
*                                                                             *
*******************************************************************************

FAULTING_IP:
+44dd220
00000000 ??              ???

EXCEPTION_RECORD:  ffffffff -- (.exr 0xffffffffffffffff)
ExceptionAddress: 00000000
   ExceptionCode: 80000003 (Break instruction exception)
  ExceptionFlags: 00000000
NumberParameters: 0

CONTEXT:  00000000 -- (.cxr 0x0;r)
eax=00fd7b48 ebx=000101db ecx=0024f500 edx=00000011 esi=0024f9e4 edi=772c19a2
eip=77720f34 esp=0024f9a4 ebp=0024f9c0 iopl=0         nv up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000246
ntdll!KiFastSystemCallRet:
77720f34 c3              ret

FAULTING_THREAD:  00000e88

DEFAULT_BUCKET_ID:  STATUS_BREAKPOINT

PROCESS_NAME:  notepad.exe

ERROR_CODE: (NTSTATUS) 0x80000003 - {EXCEPTION}  Breakpoint  A breakpoint has been reached.

EXCEPTION_CODE: (HRESULT) 0x80000003 (2147483651) - One or more arguments are invalid

NTGLOBALFLAG:  0

APPLICATION_VERIFIER_FLAGS:  0

APP:  notepad.exe

ANALYSIS_VERSION: 6.3.9600.16384 (debuggers(dbg).130821-1623) amd64fre

PRIMARY_PROBLEM_CLASS:  STATUS_BREAKPOINT

BUGCHECK_STR:  APPLICATION_FAULT_STATUS_BREAKPOINT

LAST_CONTROL_TRANSFER:  from 772c199a to 77720f34

STACK_TEXT:
0024f9a0 772c199a 772c19cd 0024f9e4 00000000 ntdll!KiFastSystemCallRet
0024f9a4 772c19cd 0024f9e4 00000000 00000000 user32!NtUserGetMessage+0xc
0024f9c0 00b9149c 0024f9e4 00000000 00000000 user32!GetMessageW+0x33
0024fa00 00b91971 00b90000 00000000 00401aaa notepad!WinMain+0xec
0024fa90 76193833 7ffd9000 0024fadc 776fa9bd notepad!_initterm_e+0x1a1
0024fa9c 776fa9bd 7ffd9000 002468e0 00000000 kernel32!BaseThreadInitThunk+0xe
0024fadc 00000000 00b931f8 7ffd9000 00000000 ntdll!_RtlUserThreadStart+0x23
```

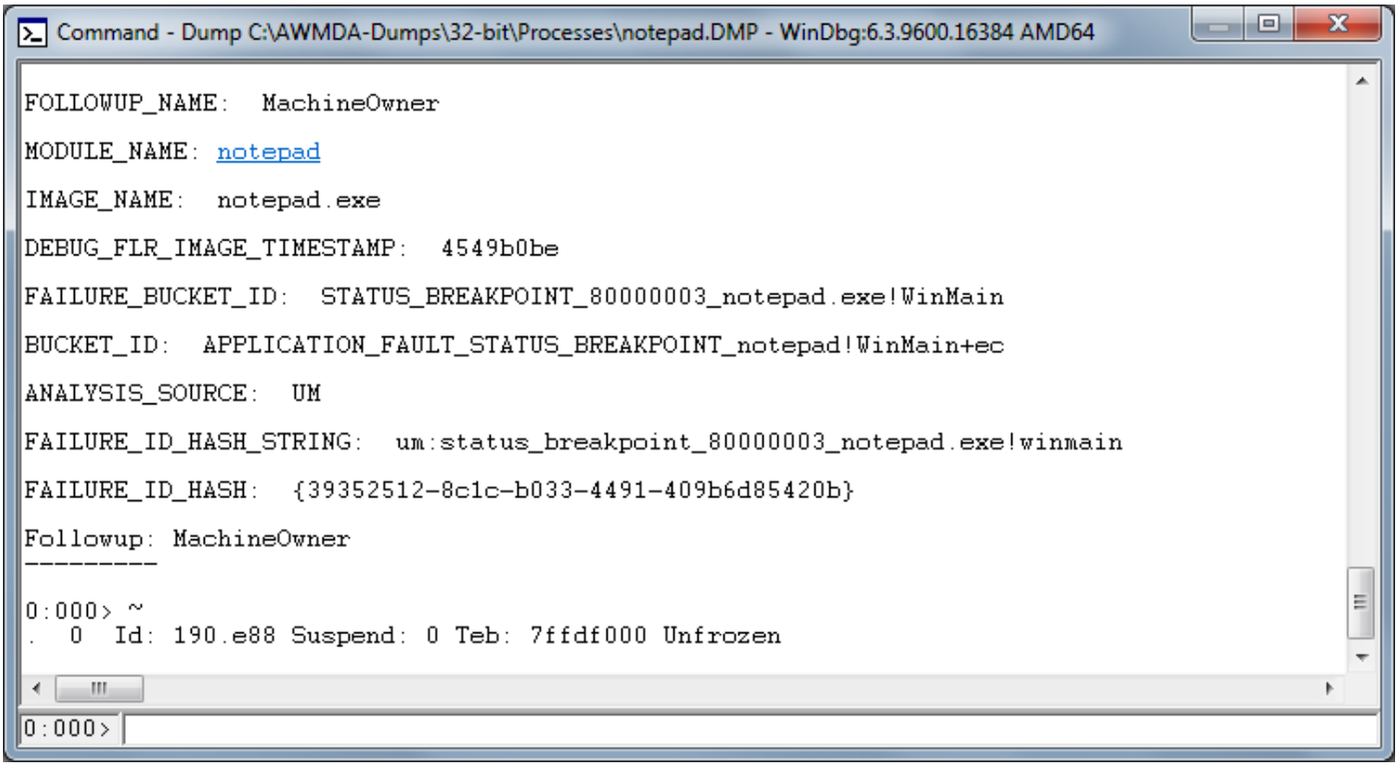
```
STACK_COMMAND:  ~0s; .ecxr ; kb

FOLLOWUP_IP:
notepad!WinMain+ec
00b9149c 85c0            test    eax,eax

SYMBOL_STACK_INDEX:  3

SYMBOL_NAME:  notepad!WinMain+ec

FOLLOWUP_NAME:  MachineOwner

MODULE_NAME: notepad

IMAGE_NAME:  notepad.exe

DEBUG_FLR_IMAGE_TIMESTAMP:  4549b0be

FAILURE_BUCKET_ID:  STATUS_BREAKPOINT_80000003_notepad.exe!WinMain

BUCKET_ID:  APPLICATION_FAULT_STATUS_BREAKPOINT_notepad!WinMain+ec

ANALYSIS_SOURCE:  UM

FAILURE_ID_HASH_STRING:  um:status_breakpoint_80000003_notepad.exe!winmain

FAILURE_ID_HASH:  {39352512-8c1c-b033-4491-409b6d85420b}

Followup: MachineOwner
---------
```

/* end of WinDbg output */

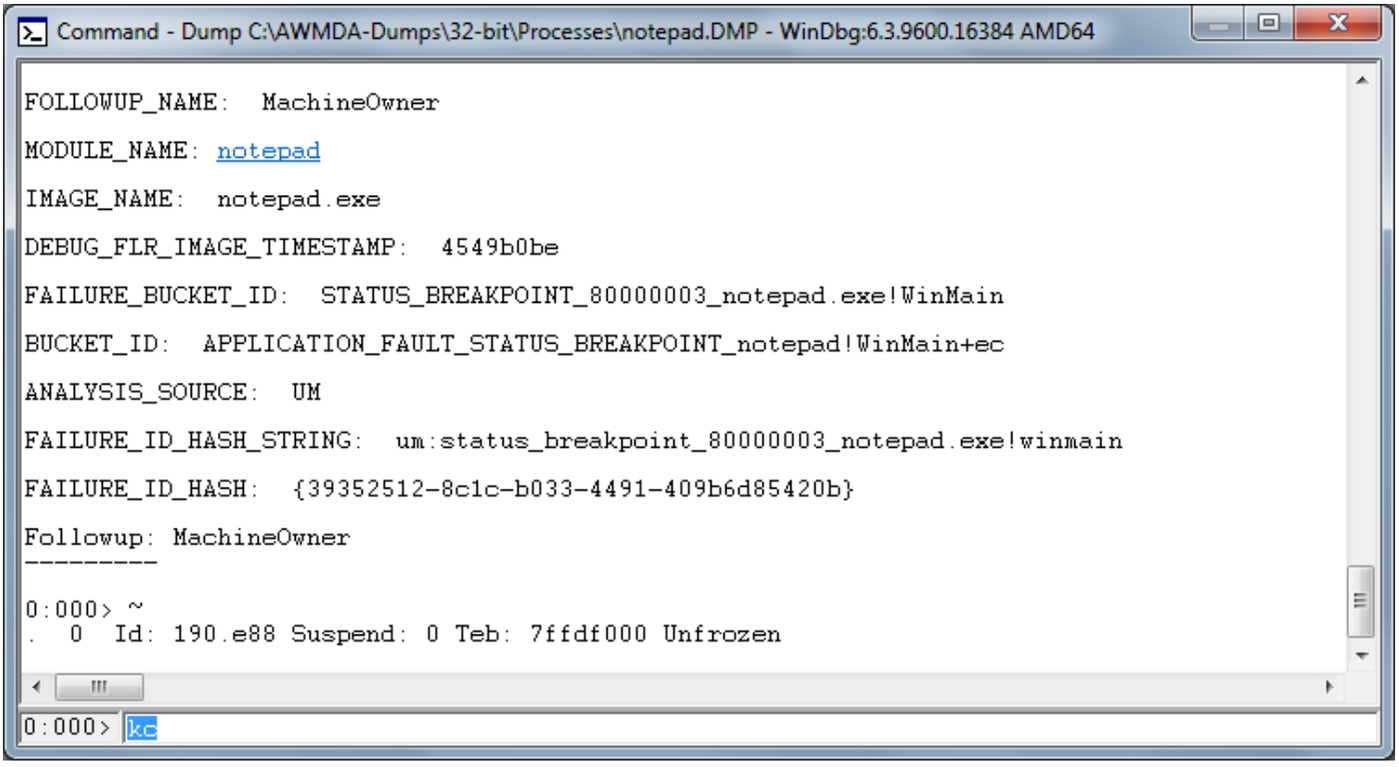
Note: "**Break instruction exception**" can be the sign of **Manual Dump** pattern but often WinDbg is not able to figure out an exception which may be on another thread or hidden. CONTEXT are CPU register values at exception time.

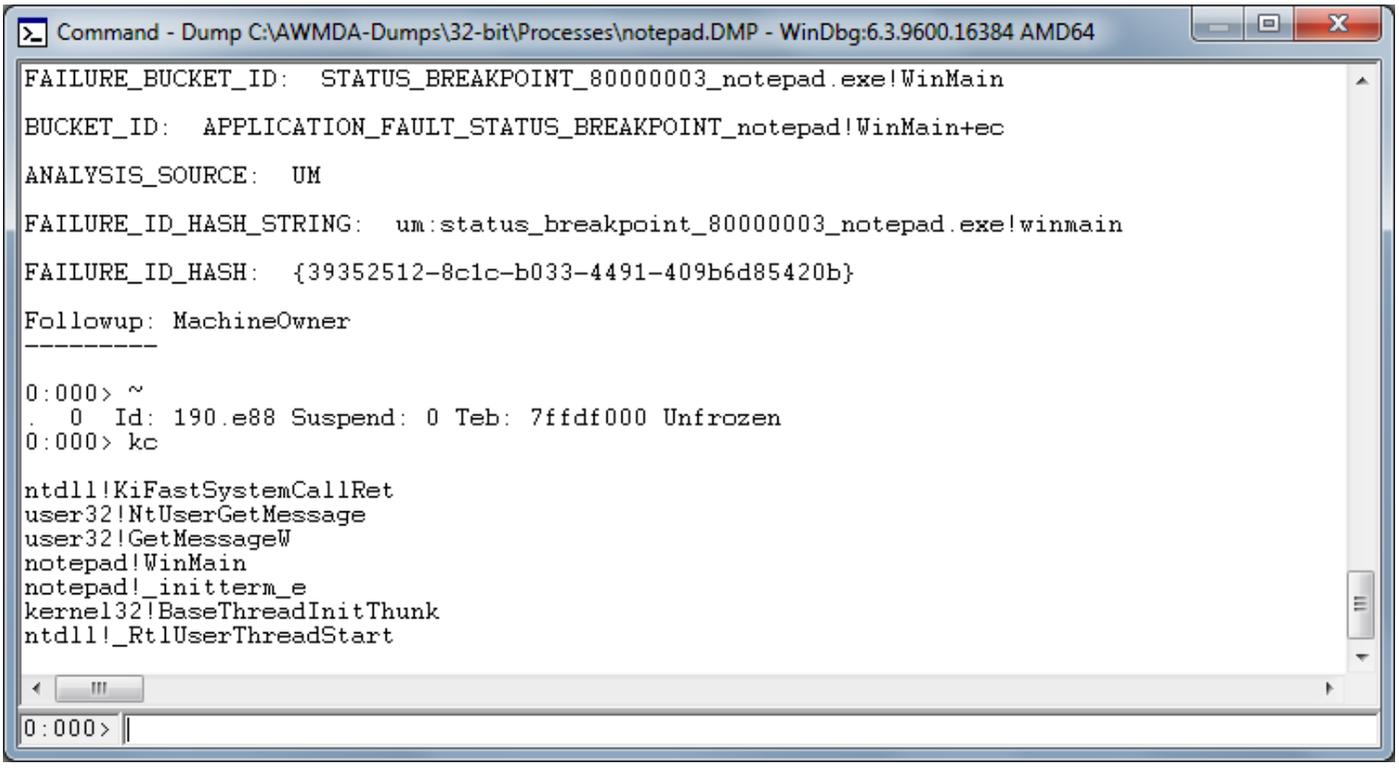
11. Now we check how many threads by using ~ command:



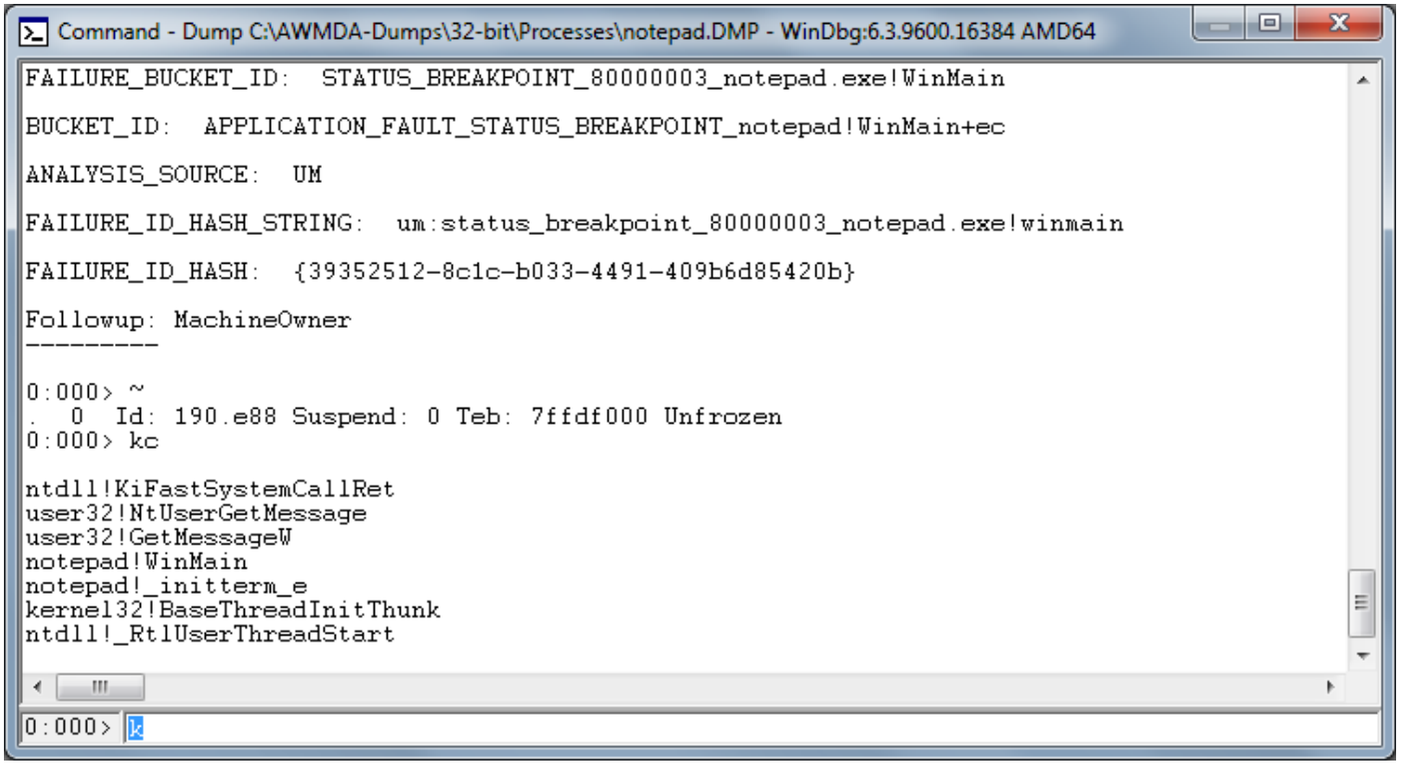


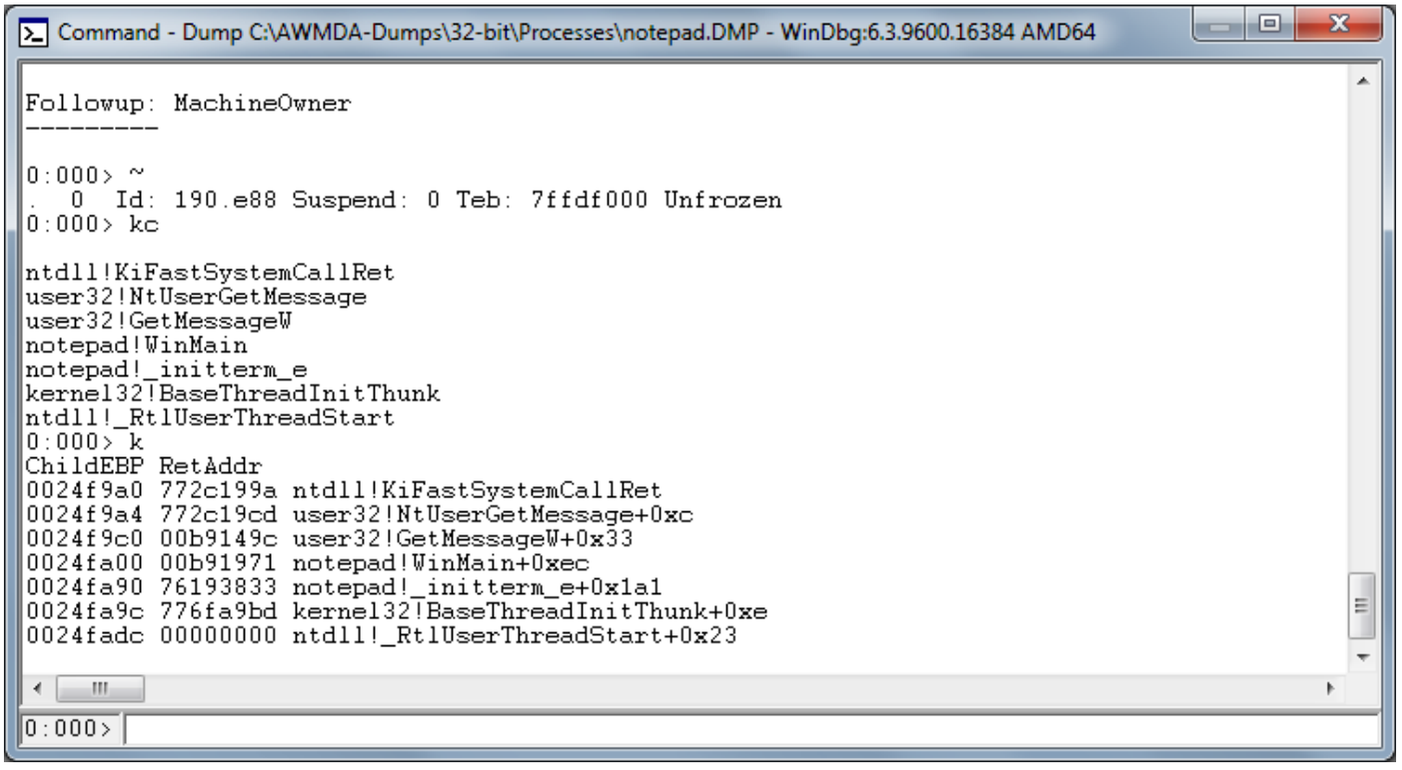
12. Now we dump a stack trace using **kc** command (only modules and symbols):





13. Now we dump the stack trace using **k** command (with symbols, return addresses and function offsets):





```
0:000> k
ChildEBP RetAddr
0024f9a0 772c199a ntdll!KiFastSystemCallRet
0024f9a4 772c19cd user32!NtUserGetMessage+0xc
0024f9c0 00b9149c user32!GetMessageW+0x33
0024fa00 00b91971 notepad!WinMain+0xec
0024fa90 76193833 notepad!_initterm_e+0x1a1
0024fa9c 776fa9bd kernel32!BaseThreadInitThunk+0xe
0024fadc 00000000 ntdll!_RtlUserThreadStart+0x23
```

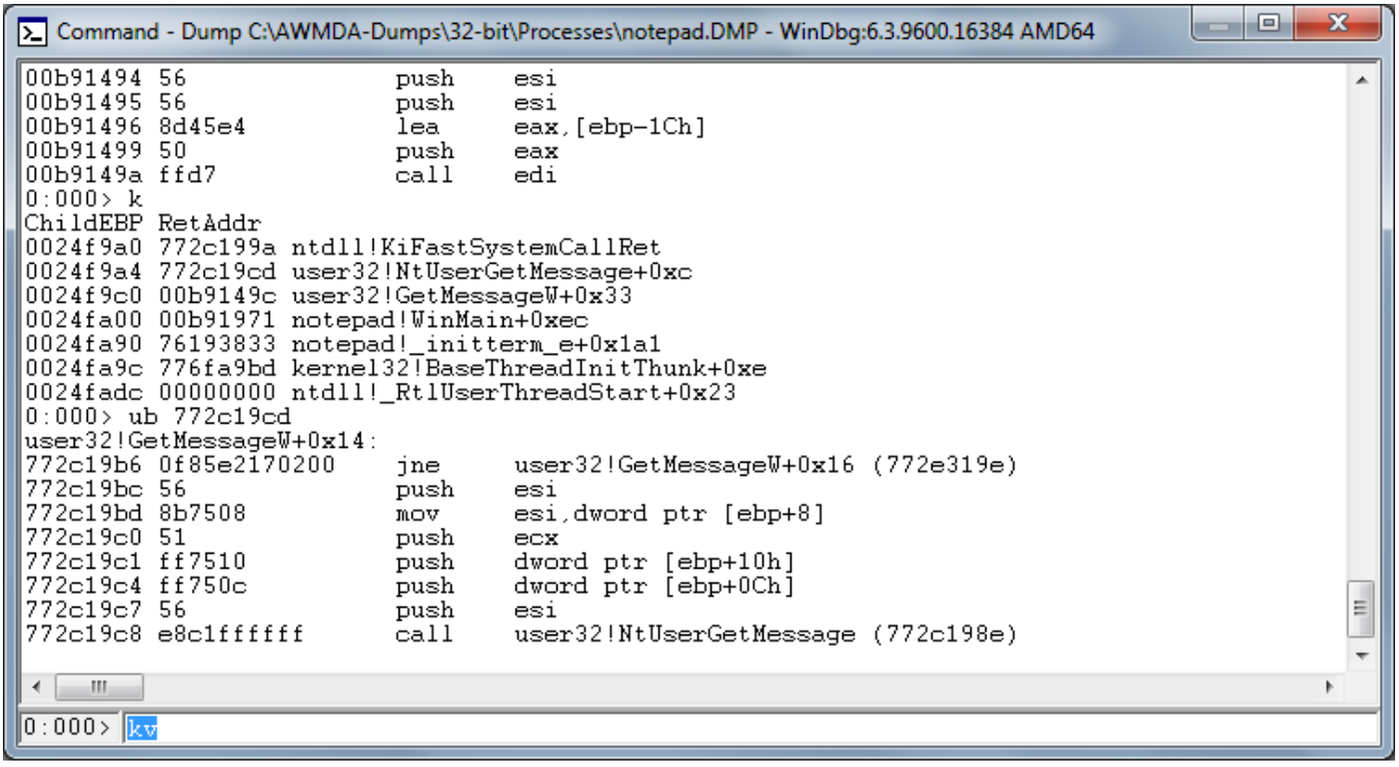
Hint: How to check that the stack trace is correct. Use **ub** command (**u**nassemble **b**ackwards) to check if there is a *call* instruction. We check that *GetMessageW* function was called from *WinMain* function:

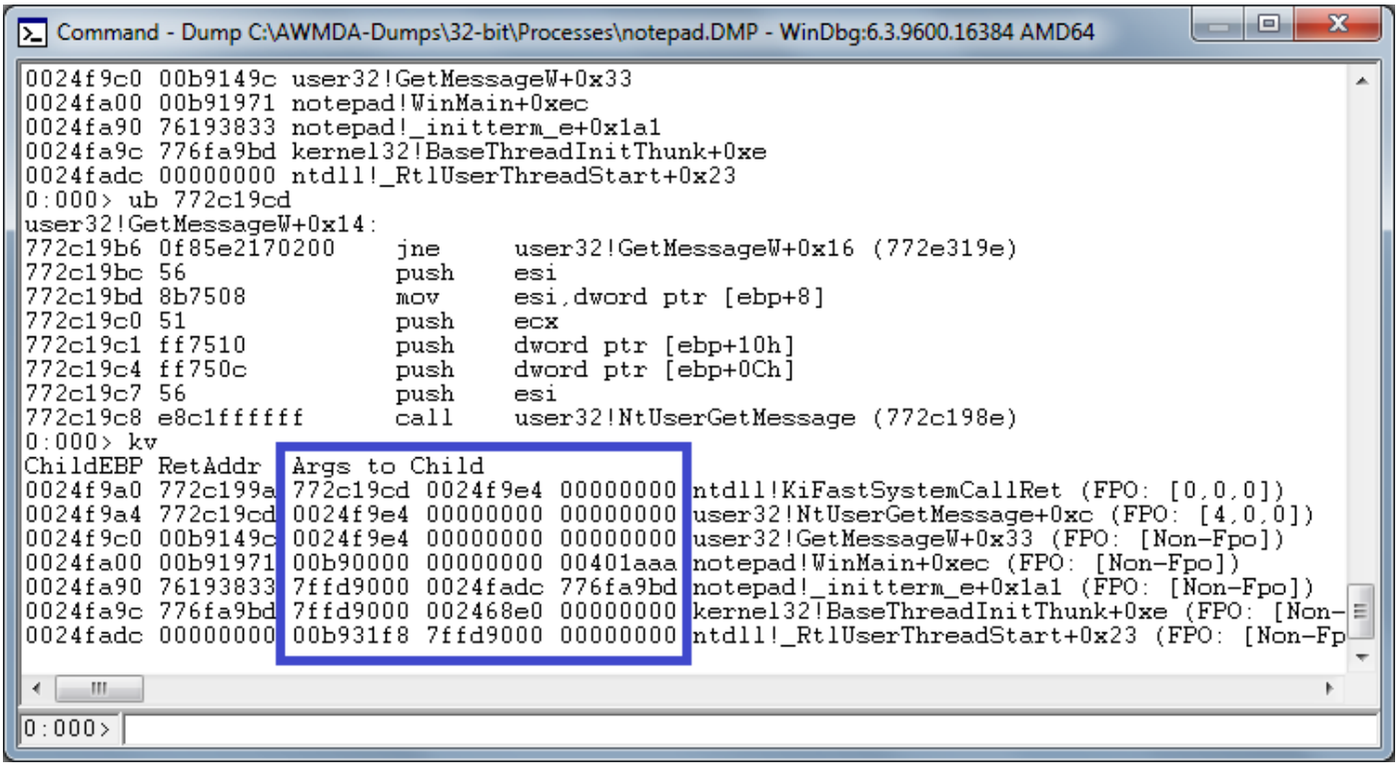
```
0:000> k
ChildEBP RetAddr
0024f9a0 772c199a ntdll!KiFastSystemCallRet
0024f9a4 772c19cd user32!NtUserGetMessage+0xc
0024f9c0 00b9149c user32!GetMessageW+0x33
0024fa00 00b91971 notepad!WinMain+0xec
0024fa90 76193833 notepad!_initterm_e+0x1a1
0024fa9c 776fa9bd kernel32!BaseThreadInitThunk+0xe
0024fadc 00000000 ntdll!_RtlUserThreadStart+0x23

0:000> ub 00b9149c
notepad!WinMain+0x7a:
00b9148b 8b3d2812b900    mov     edi,dword ptr [notepad!_imp__GetMessageW (00b91228)]
00b91491 8bd8            mov     ebx,eax
00b91493 56              push    esi
00b91494 56              push    esi
00b91495 56              push    esi
00b91496 8d45e4          lea     eax,[ebp-1Ch]
00b91499 50              push    eax
00b9149a ffd7            call    edi
```

Then we check that *NtUserGetMessage* function was called from *GetMessageW* function:

```
0:000> k
ChildEBP RetAddr
0024f9a0 772c199a ntdll!KiFastSystemCallRet
0024f9a4 772c19cd user32!NtUserGetMessage+0xc
0024f9c0 00b9149c user32!GetMessageW+0x33
0024fa00 00b91971 notepad!WinMain+0xec
0024fa90 76193833 notepad!_initterm_e+0x1a1
0024fa9c 776fa9bd kernel32!BaseThreadInitThunk+0xe
0024fadc 00000000 ntdll!_RtlUserThreadStart+0x23

0:000> ub 772c19cd
user32!GetMessageW+0x14:
772c19b6 0f85e2170200    jne     user32!GetMessageW+0x16 (772e319e)
772c19bc 56              push    esi
772c19bd 8b7508          mov     esi,dword ptr [ebp+8]
772c19c0 51              push    ecx
772c19c1 ff7510          push    dword ptr [ebp+10h]
772c19c4 ff750c          push    dword ptr [ebp+0Ch]
772c19c7 56              push    esi
772c19c8 e8c1ffffff      call    user32!NtUserGetMessage (772c198e)
```

14. Now we dump the stack trace using verbose **kv** command (includes the first *possible* function parameters):

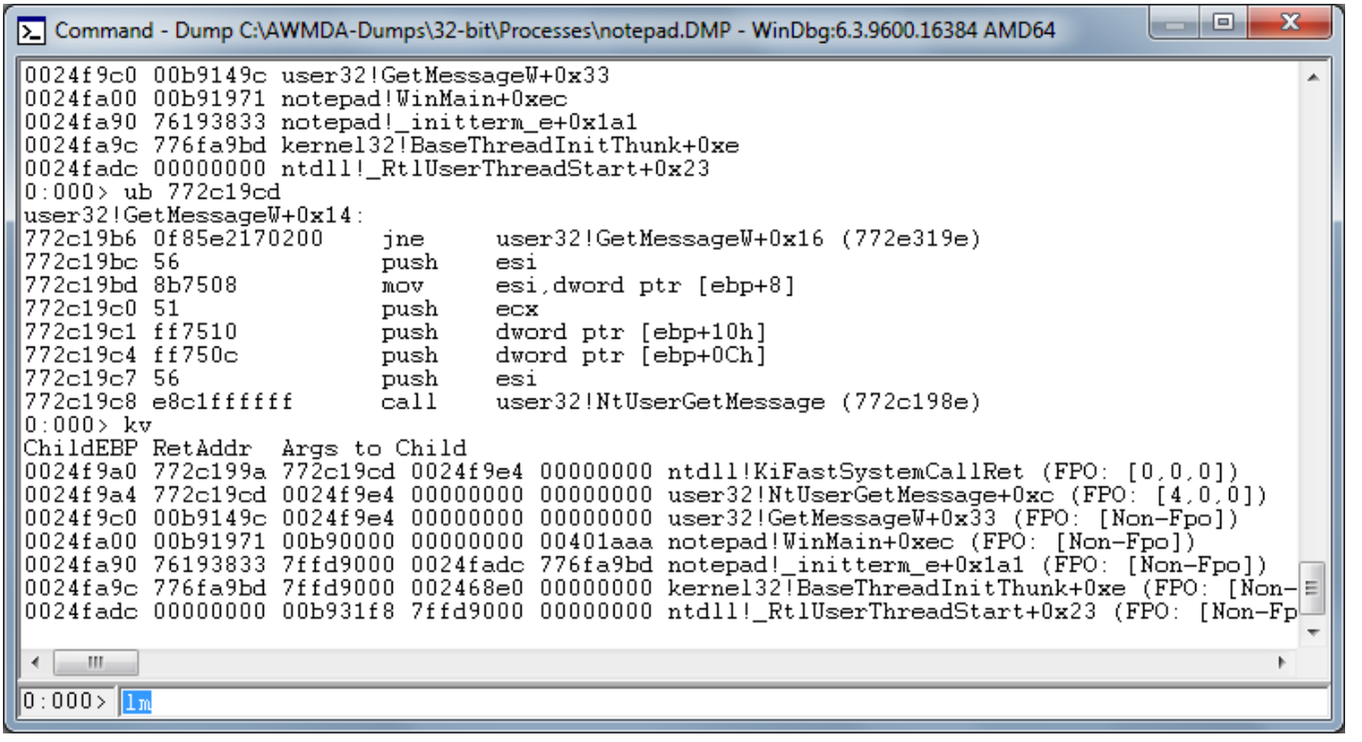


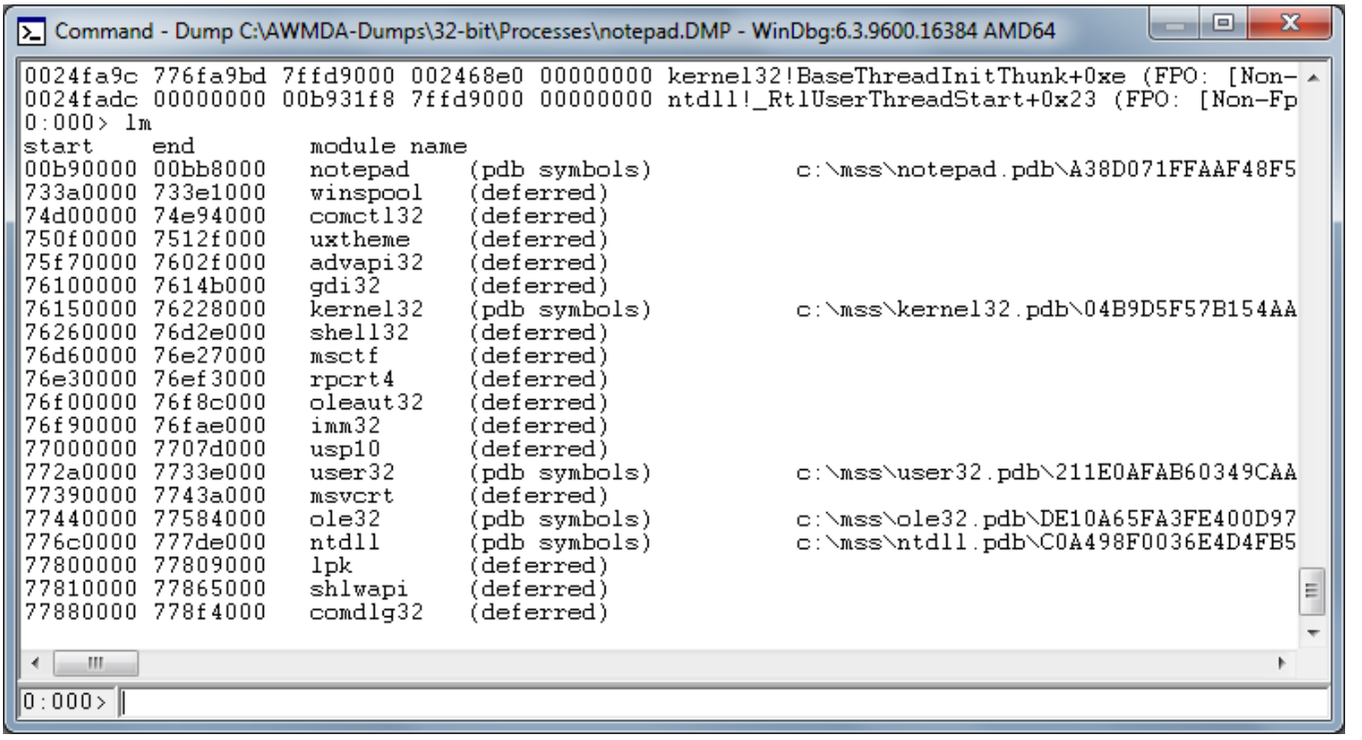


Note: Remember the functions call each other from bottom to top. The topmost function is the last one that was called. **ExceptionAddress** or **FAULTING_IP** may point to the last one. We would come to this in the real exception process dumps later. Here in another example below I would like to point out that the top function call **func1** has a return address already (to *func2*) and the function was being executed somewhere in its code at *0x20* offset:

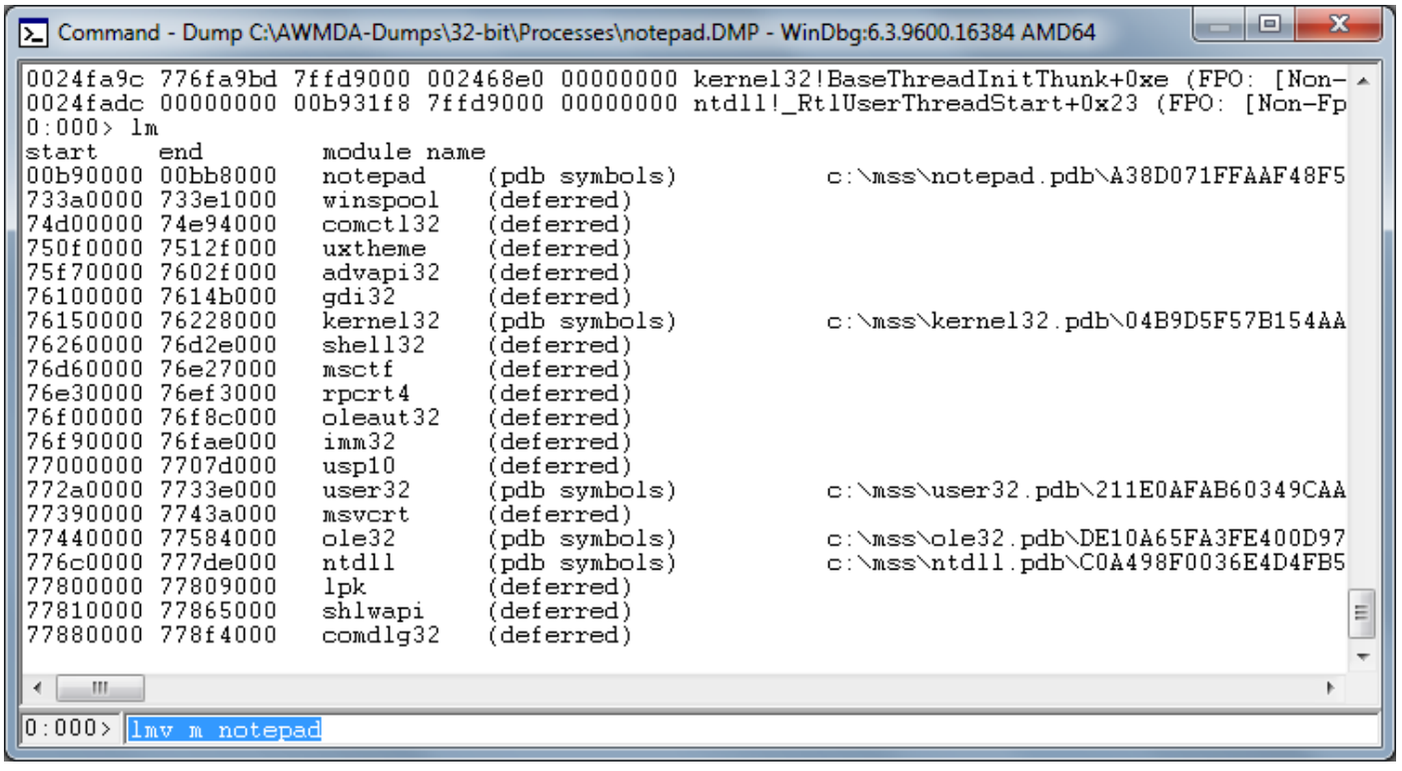
```
0:000> k
ChildEBP RetAddr
0024f9a0 772c199a ModuleA!func1+0x20
0024f9a4 772c19cd ModuleA!func2+0x16
[...]
0024fa9c 776fa9bd kernel32!BaseThreadInitThunk+0xe
0024fadc 00000000 ntdll!_RtlUserThreadStart+0x23
```

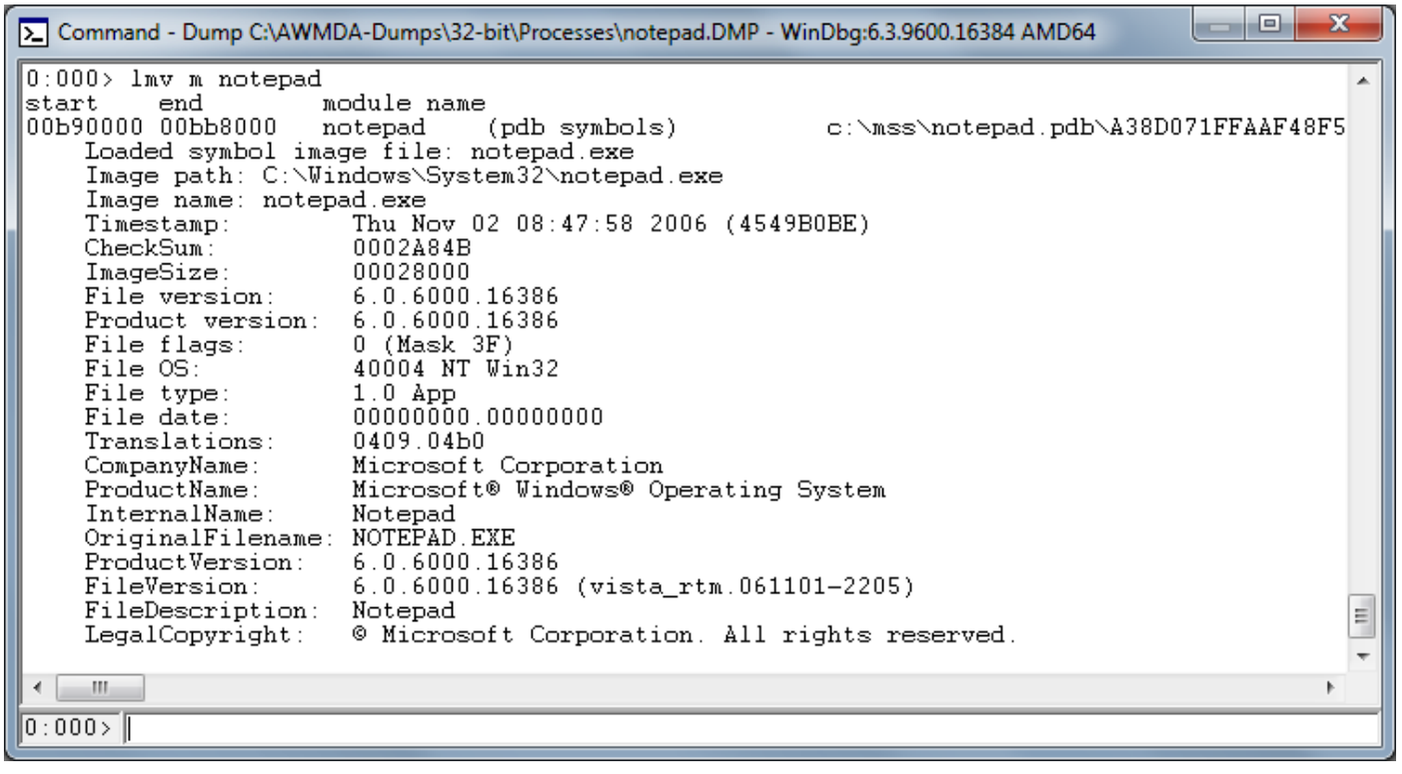
15. Now we check the list of loaded modules using **lm** command:



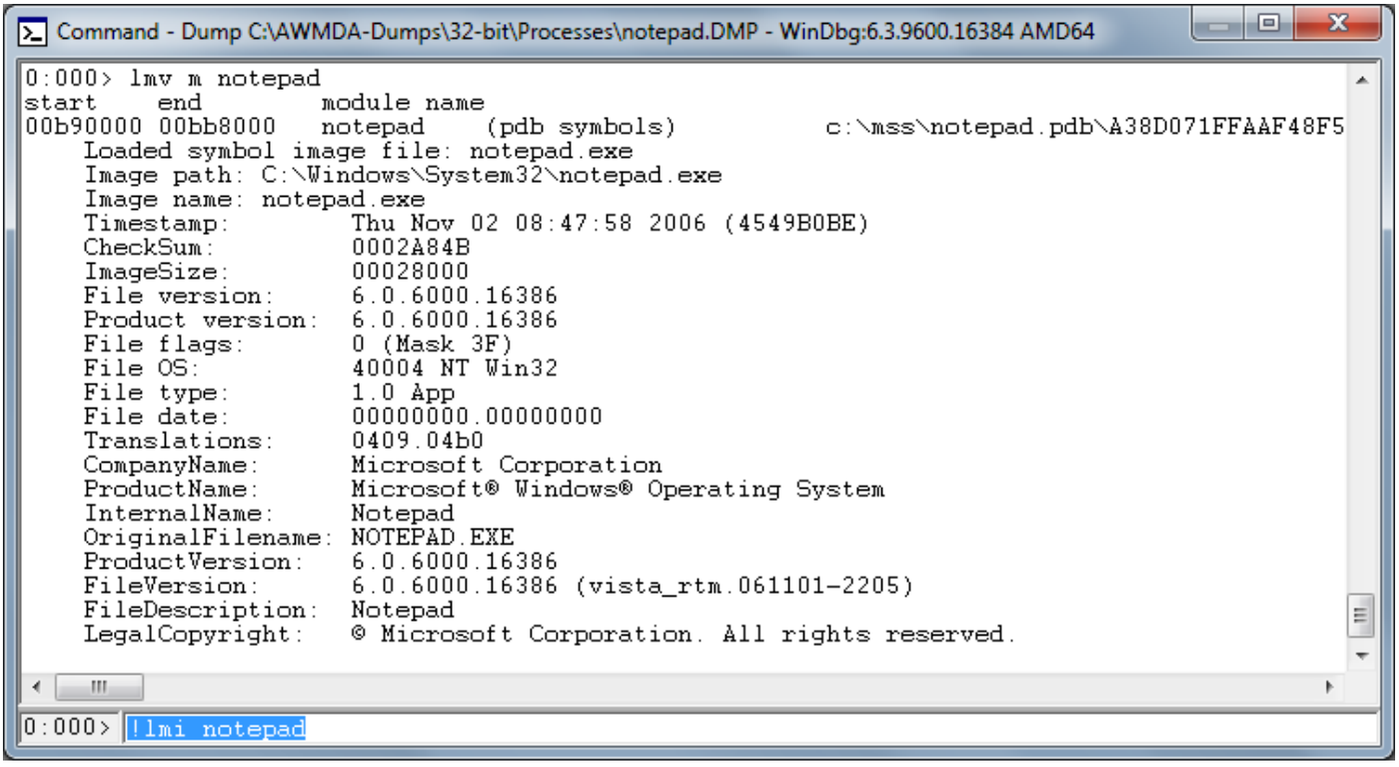


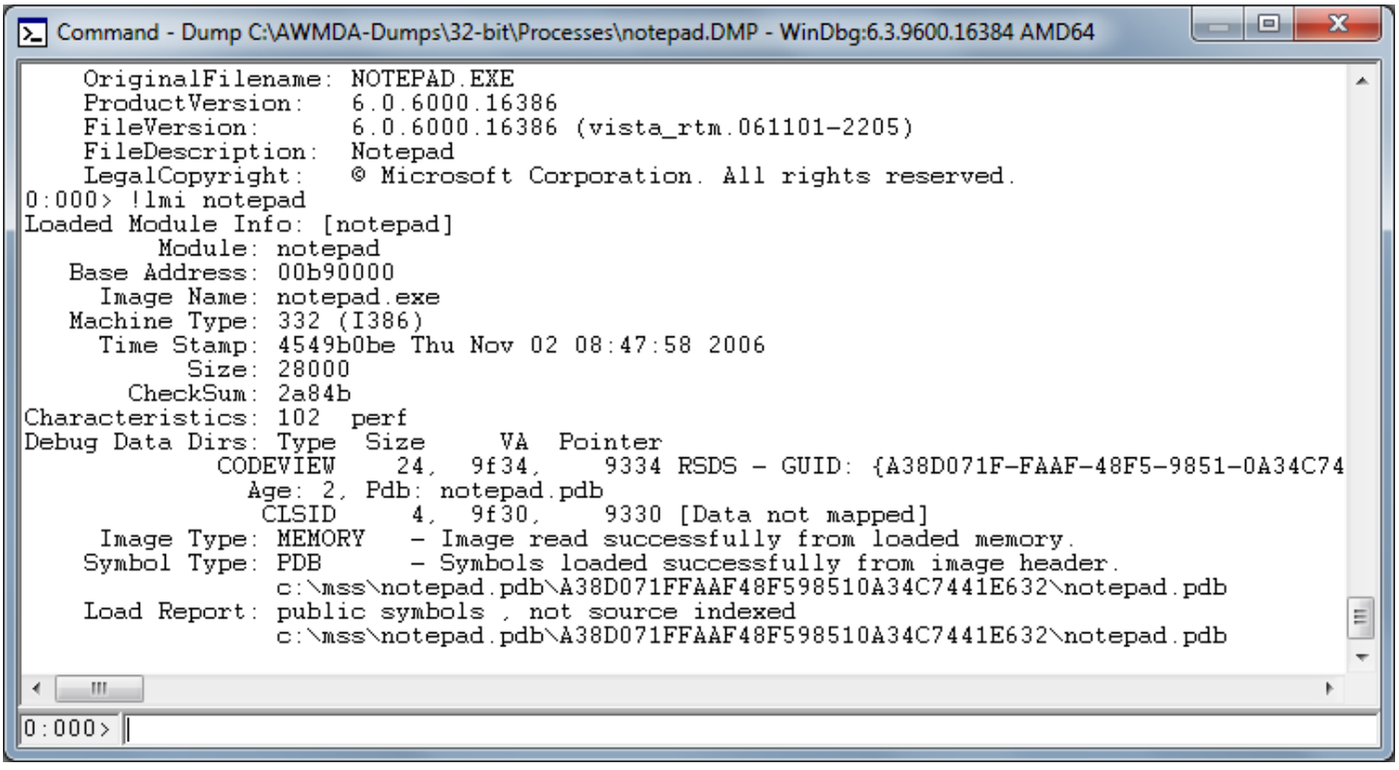
16. We can check verbose module information using **lmv** command or use **lmv m** *<module name>* to check an individual module (**Not My Version** pattern):





17. Sometimes **lmv** command doesn't show much and **!lmi** command might give extra information:





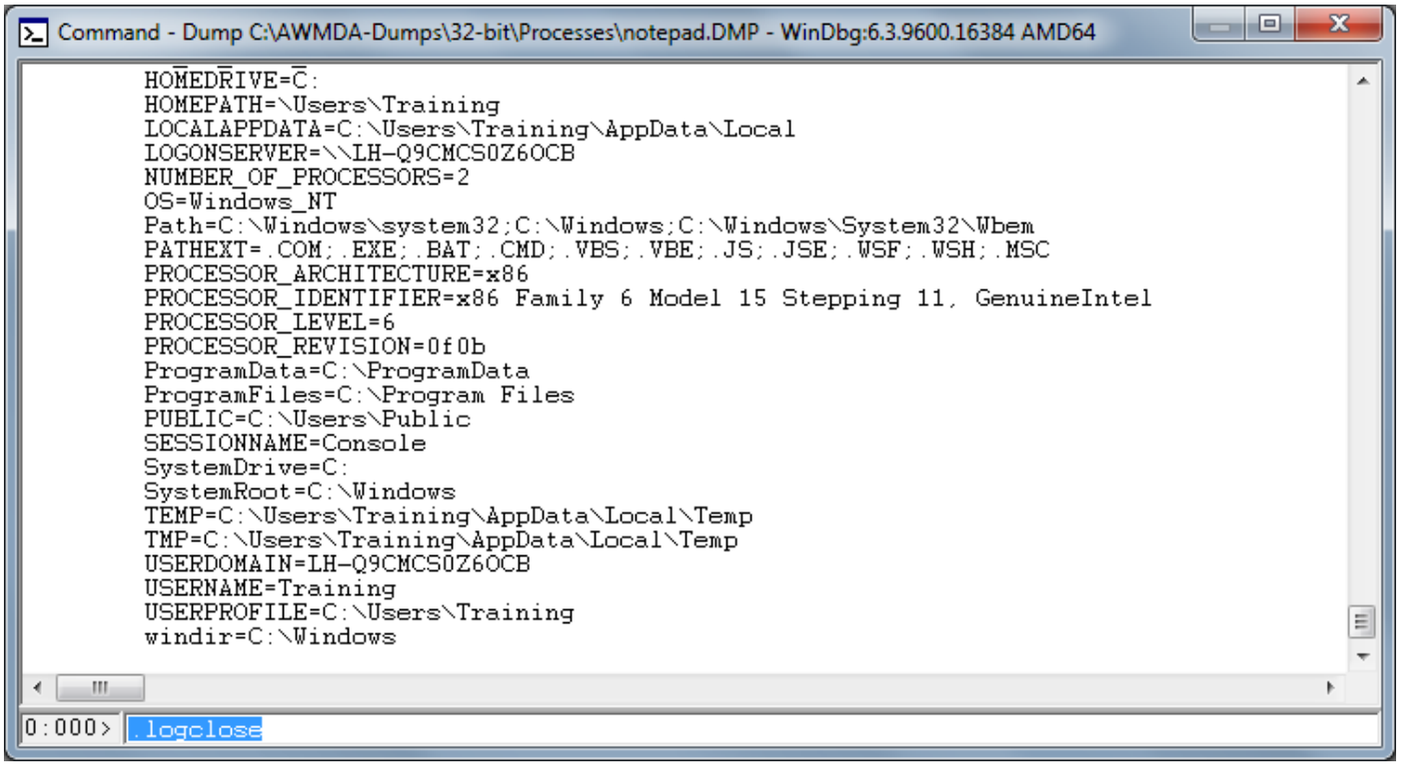
Note: We can also use **lmt** command variant if we are interested in timestamps only.

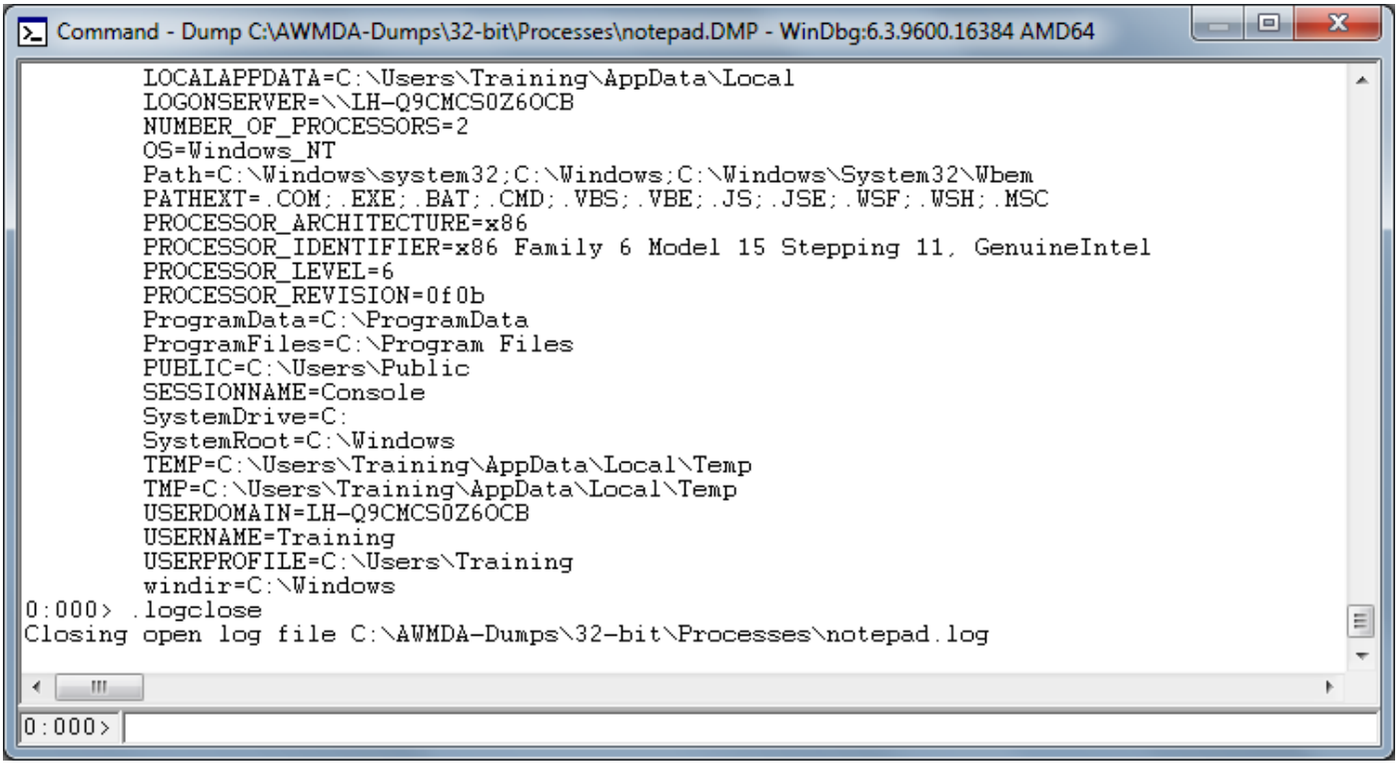
18. Sometimes **Environment Hint** pattern can give troubleshooting suggestions related to environment variables and DLL paths. **!peb** command (**P**rocess **E**nvironment **B**lock):

```
0:000> !peb
PEB at 7ffd9000
    InheritedAddressSpace:    No
    ReadImageFileExecOptions: No
    BeingDebugged:            No
    ImageBaseAddress:         00b90000
    Ldr                       77785d00
    Ldr.Initialized:          Yes
    Ldr.InInitializationOrderModuleList: 004016d8 . 00418b38
    Ldr.InLoadOrderModuleList:           00401658 . 00418b28
    Ldr.InMemoryOrderModuleList:         00401660 . 00418b30
            Base TimeStamp                     Module
          b90000 4549b0be Nov 02 08:47:58 2006 C:\Windows\System32\notepad.exe
         776c0000 4549bdc9 Nov 02 09:43:37 2006 C:\Windows\system32\ntdll.dll
         76150000 4549bd80 Nov 02 09:42:24 2006 C:\Windows\system32\kernel32.dll
         75f70000 4549bcd2 Nov 02 09:39:30 2006 C:\Windows\system32\ADVAPI32.dll
         76e30000 4549bdab Nov 02 09:43:07 2006 C:\Windows\system32\RPCRT4.dll
         76100000 4549bcd3 Nov 02 09:39:31 2006 C:\Windows\system32\GDI32.dll
         772a0000 4549bde0 Nov 02 09:44:00 2006 C:\Windows\system32\USER32.dll
         77390000 4549bd61 Nov 02 09:41:53 2006 C:\Windows\system32\msvcrt.dll
         77880000 4549bd09 Nov 02 09:40:25 2006 C:\Windows\system32\COMDLG32.dll
         77810000 4549bdb9 Nov 02 09:43:21 2006 C:\Windows\system32\SHLWAPI.dll
         74d00000 4549bd09 Nov 02 09:40:25 2006 C:\Windows\WinSxS\x86_microsoft.windows.common-
controls_6595b64144ccf1df_6.0.6000.16386_none_5d07289e07e1d100\COMCTL32.dll
         76260000 4549bdb4 Nov 02 09:43:16 2006 C:\Windows\system32\SHELL32.dll
         733a0000 4549be2a Nov 02 09:45:14 2006 C:\Windows\System32\WINSPOOL.DRV
         77440000 4549bd92 Nov 02 09:42:42 2006 C:\Windows\system32\ole32.dll
         76f00000 4549bd95 Nov 02 09:42:45 2006 C:\Windows\system32\OLEAUT32.dll
         76f90000 4549bd29 Nov 02 09:40:57 2006 C:\Windows\system32\IMM32.DLL
         76d60000 4549bd4a Nov 02 09:41:30 2006 C:\Windows\system32\MSCTF.dll
         77800000 4549bcff Nov 02 09:40:15 2006 C:\Windows\system32\LPK.DLL
         77000000 4549bde3 Nov 02 09:44:03 2006 C:\Windows\system32\USP10.dll
         750f0000 4549bde7 Nov 02 09:44:07 2006 C:\Windows\System32\uxtheme.dll
    SubSystemData:     00000000
    ProcessHeap:       00400000
    ProcessParameters: 00400f48
    CurrentDirectory:  'C:\Users\Training\'
    WindowTitle:  'C:\Users\Training\AppData\Roaming\Microsoft\Windows\Start
Menu\Programs\Accessories\Notepad.lnk'
    ImageFile:    'C:\Windows\System32\notepad.exe'
    CommandLine:  '"C:\Windows\System32\notepad.exe" '
    DllPath:
'C:\Windows\System32;C:\Windows\system32;C:\Windows\system;C:\Windows;.;C:\Windows\system32;C:\
Windows;C:\Windows\System32\Wbem'
    Environment:  004007e8
        =::=::\
        ALLUSERSPROFILE=C:\ProgramData
        APPDATA=C:\Users\Training\AppData\Roaming
        CommonProgramFiles=C:\Program Files\Common Files
        COMPUTERNAME=LH-Q9CMCS0Z6OCB
        ComSpec=C:\Windows\system32\cmd.exe
        FP_NO_HOST_CHECK=NO
        HOMEDRIVE=C:
        HOMEPATH=\Users\Training
        LOCALAPPDATA=C:\Users\Training\AppData\Local
        LOGONSERVER=\\LH-Q9CMCS0Z6OCB
        NUMBER_OF_PROCESSORS=2
```

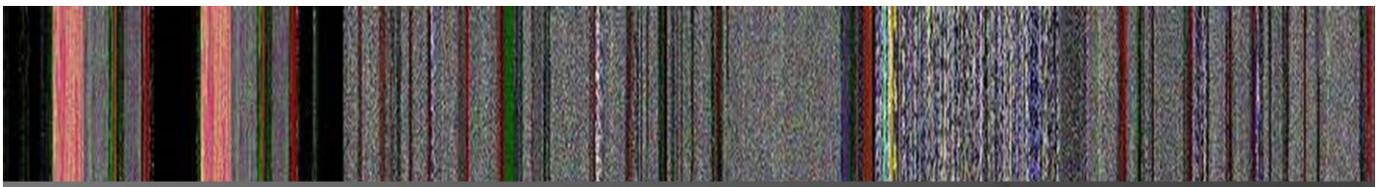
```
OS=Windows_NT
Path=C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 6 Model 15 Stepping 11, GenuineIntel
PROCESSOR_LEVEL=6
PROCESSOR_REVISION=0f0b
ProgramData=C:\ProgramData
ProgramFiles=C:\Program Files
PUBLIC=C:\Users\Public
SESSIONNAME=Console
SystemDrive=C:
SystemRoot=C:\Windows
TEMP=C:\Users\Training\AppData\Local\Temp
TMP=C:\Users\Training\AppData\Local\Temp
USERDOMAIN=LH-Q9CMCS0Z6OCB
USERNAME=Training
USERPROFILE=C:\Users\Training
windir=C:\Windows
```

19. We close logging before exiting WinDbg:





Note: To avoid possible confusion and glitches we recommend to exit WinDbg after each exercise.



.NET Memory Dump Analysis Accelerated

Version 2.0

Dmitry Vostokov
Software Diagnostics Services

Published by OpenTask, Republic of Ireland

OpenTask books and magazines are available through booksellers and distributors worldwide. For further information or comments send requests to press@opentask.com.

A CIP catalogue record for this book is available from the British Library.

ISBN-l3: 978-1-908043-59-7 (Paperback)

Second edition, 1st revision, 2013

Contents

Exercise PN1: Analysis of an application process dump (ApplicationA, 32-bit, CLR 2)

Goal: Learn how to load the correct .NET SOS WinDbg extension and analyze managed space.

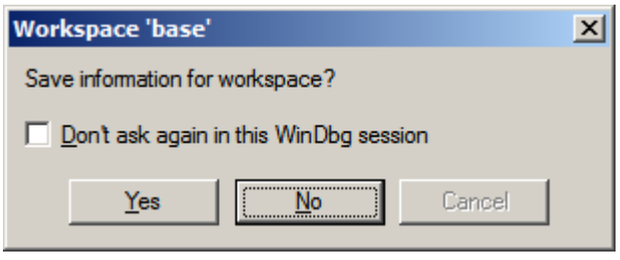
Patterns: Stack Trace Collection; CLR Thread; Version-Specific Extension; Managed Code Exception; Managed Stack Trace

Commands: .logopen, ~*k, !analyze –v, !pe, ~*e, lmv, .chain, .unload, .load, .logclose

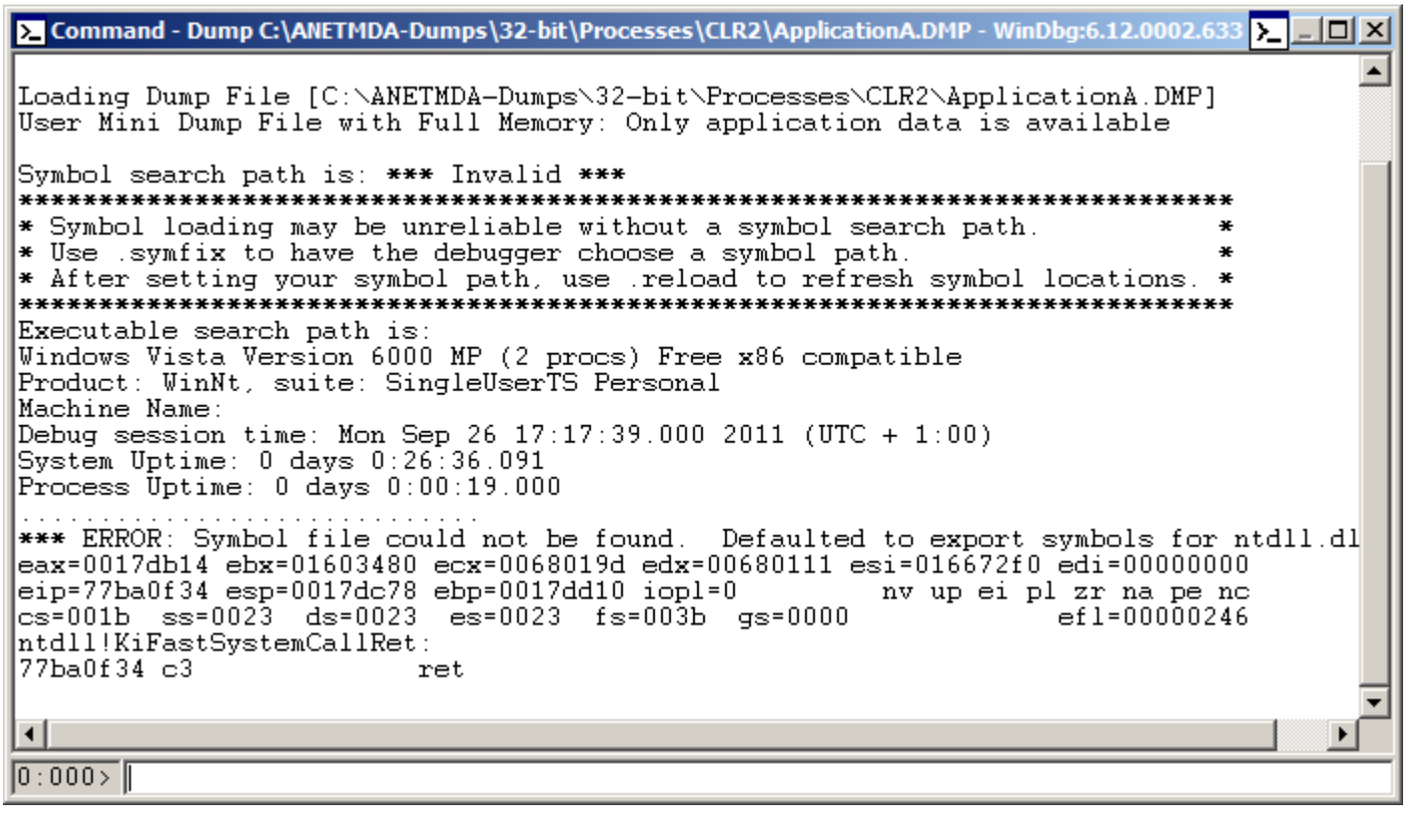
1. Launch WinDbg from Debugging Tools for Windows or Debugging Tools for Windows (x86)

2. Open \ANETMDA-Dumps\32-bit\Processes\CLR2\ApplicationA.DMP

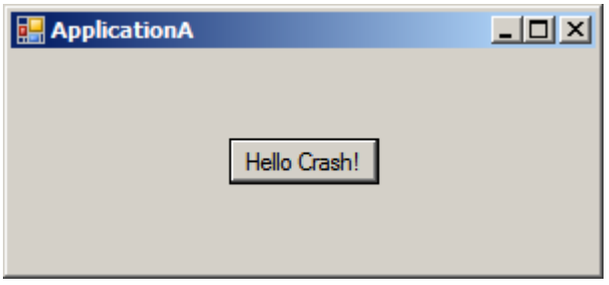
3. If you are presented with this dialog say No:



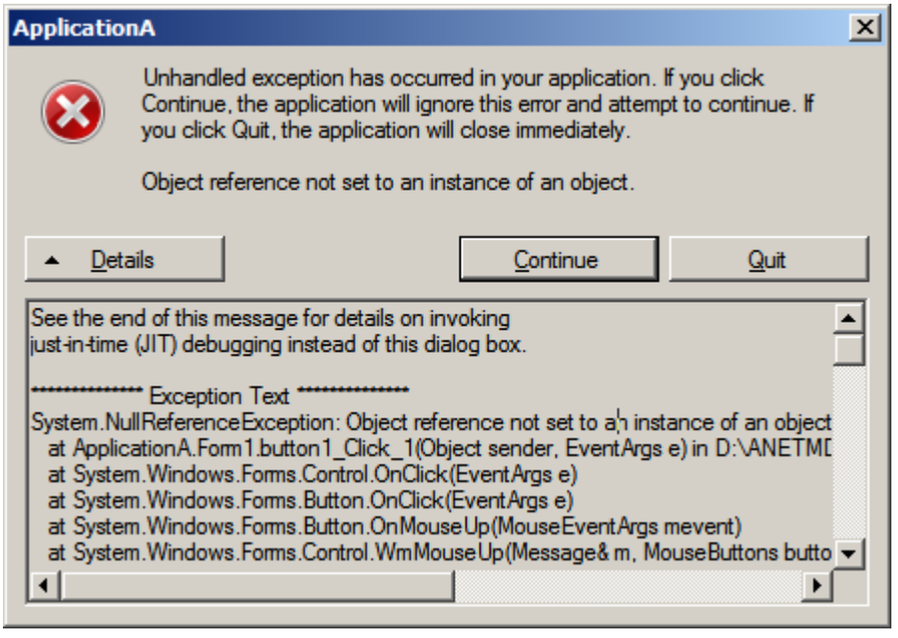
4. We get the dump file loaded:



Note: ApplicationA shows this dialog when launched:



When we click on a button it shows the following exception dialog:



At this point we saved a process memory dump on a Vista system using Task Manager.

5. Open a log file using **.logopen** command and load symbols (**.symfix** and **.reload** commands):

```
0:000> .logopen C:\ANETMDA-Dumps\32-bit\Processes\CLR2\ApplicationA.log
Opened log file 'C:\ANETMDA-Dumps\32-bit\Processes\CLR2\ApplicationA.log'

0:000> .symfix c:\mss

0:000> .reload
.............................
```

Note: The results may be slightly different on your system if you have 32-bit .NET Framework 2.0.50727.**312** already installed or you don't have any .NET Framework 2.0.50727 installed.

6. Type ~*__k__ command to verify the correctness of all stack traces:

```
0:000> ~*k

.  0  Id: dec.dac Suspend: 0 Teb: 7ffdf000 Unfrozen
ChildEBP RetAddr
0017dc74 7720b5b4 ntdll!KiFastSystemCallRet
*** WARNING: Unable to verify checksum for System.Windows.Forms.ni.dll
0017dc78 7b061e04 user32!NtUserWaitMessage+0xc
0017dd10 7b0620f1 System_Windows_Forms_ni+0x91e04
0017dd80 7b062293 System_Windows_Forms_ni+0x920f1
0017ddb0 7b2256ae System_Windows_Forms_ni+0x92293
0017de94 7b2251af System_Windows_Forms_ni+0x2556ae
0017ded4 7b0d1b87 System_Windows_Forms_ni+0x2551af
0017ed38 77211a10 System_Windows_Forms_ni+0x101b87
0017ed64 77211ae8 user32!InternalCallWinProc+0x23
0017eddc 77212a47 user32!UserCallWinProcCheckWow+0x14b
0017ee40 77212a98 user32!DispatchMessageWorker+0x322
0017ee50 00911046 user32!DispatchMessageW+0xf
WARNING: Frame IP not in any known module. Following frames may be wrong.
0017ee6c 7b061c9a 0x911046
0017ef24 7b0620f1 System_Windows_Forms_ni+0x91c9a
0017ef94 7b062293 System_Windows_Forms_ni+0x920f1
0017efc4 7b0c7a3a System_Windows_Forms_ni+0x92293
0017eff0 79e826bd System_Windows_Forms_ni+0xf7a3a
0017f070 79e8451b mscorwks!CallDescrWorkerWithHandler+0xa3
0017f1b4 79e84403 mscorwks!MethodDesc::CallDescr+0x19c
0017f1cc 79e843e0 mscorwks!MethodDesc::CallTargetWorker+0x20
0017f1e0 79ef3e20 mscorwks!MethodDescCallSite::CallWithValueTypes+0x18
0017f344 79ef3c19 mscorwks!ClassLoader::RunMain+0x220
0017f5ac 79ef3aee mscorwks!Assembly::ExecuteMainMethod+0xa6
0017fa7c 79ef3737 mscorwks!SystemDomain::ExecuteMainMethod+0x398
0017facc 79ef18d1 mscorwks!ExecuteEXE+0x59
0017fb14 6e1e55ab mscorwks!_CorExeMain+0x11b
0017fb20 6e187f16 mscoreei!_CorExeMain+0x38
0017fb30 6e184de3 mscoree!ShellShim__CorExeMain+0x99
0017fb38 77563833 mscoree!_CorExeMain_Exported+0x8
0017fb44 77b7a9bd kernel32!BaseThreadInitThunk+0xe
0017fb84 00000000 ntdll!_RtlUserThreadStart+0x23

   1  Id: dec.d14 Suspend: 0 Teb: 7ffde000 Unfrozen
ChildEBP RetAddr
015ff9d0 77ba0690 ntdll!KiFastSystemCallRet
015ff9d4 77567e09 ntdll!ZwWaitForMultipleObjects+0xc
015ffa70 77568150 kernel32!WaitForMultipleObjectsEx+0x11d
015ffa8c 79eec20e kernel32!WaitForMultipleObjects+0x18
015ffaec 79eec16b mscorwks!DebuggerRCThread::MainLoop+0xcf
015ffb1c 79eec0ae mscorwks!DebuggerRCThread::ThreadProc+0xca
015ffb4c 77563833 mscorwks!DebuggerRCThread::ThreadProcStatic+0x82
015ffb58 77b7a9bd kernel32!BaseThreadInitThunk+0xe
015ffb98 00000000 ntdll!_RtlUserThreadStart+0x23
```

```
   2  Id: dec.1b0 Suspend: 0 Teb: 7ffdd000 Unfrozen
ChildEBP RetAddr
00f7f698 77ba0690 ntdll!KiFastSystemCallRet
00f7f69c 77567e09 ntdll!ZwWaitForMultipleObjects+0xc
00f7f738 77568150 kernel32!WaitForMultipleObjectsEx+0x11d
00f7f754 79f8c8ec kernel32!WaitForMultipleObjects+0x18
00f7f774 79ec547e mscorwks!WKS::WaitForFinalizerEvent+0x7a
00f7f788 79ee171c mscorwks!WKS::GCHeap::FinalizerThreadWorker+0x75
00f7f798 79ee16ba mscorwks!Thread::UserResumeThread+0xfb
00f7f82c 79ee15df mscorwks!Thread::DoADCallBack+0x355
00f7f868 79eed73a mscorwks!Thread::DoADCallBack+0x541
00f7f890 79eed705 mscorwks!ManagedThreadBase_NoADTransition+0x32
00f7f89c 79eeecdb mscorwks!ManagedThreadBase::FinalizerBase+0xb
00f7f8d4 79ee118b mscorwks!WKS::GCHeap::FinalizerThreadStart+0xbb
00f7f96c 77563833 mscorwks!Thread::intermediateThreadProc+0x49
00f7f978 77b7a9bd kernel32!BaseThreadInitThunk+0xe
00f7f9b8 00000000 ntdll!_RtlUserThreadStart+0x23

   3  Id: dec.8ac Suspend: 0 Teb: 7ffdc000 Unfrozen
ChildEBP RetAddr
03a0f8a8 77ba0690 ntdll!KiFastSystemCallRet
03a0f8ac 77567e09 ntdll!ZwWaitForMultipleObjects+0xc
03a0f948 7720c4af kernel32!WaitForMultipleObjectsEx+0x11d
03a0f99c 77208b7b user32!RealMsgWaitForMultipleObjectsEx+0x13c
03a0f9b8 74c71965 user32!MsgWaitForMultipleObjects+0x1f
03a0fa04 77563833 GdiPlus!BackgroundThreadProc+0x59
03a0fa10 77b7a9bd kernel32!BaseThreadInitThunk+0xe
03a0fa50 00000000 ntdll!_RtlUserThreadStart+0x23
```

Note: We see that all threads except the last one have **mscorwks** module on their stack traces (newer version of .NET 4.x uses **clr** module).

7. Let's see what **!analyze -v** command says:

```
                                /* start of WinDbg output */

0:000> !analyze -v
*******************************************************************************
*                                                                             *
*                        Exception Analysis                                   *
*                                                                             *
*******************************************************************************

*** WARNING: Unable to verify checksum for ApplicationA.exe
*** ERROR: Module load completed but symbols could not be loaded for ApplicationA.exe
*** WARNING: Unable to verify checksum for mscorlib.ni.dll
GetPageUrlData failed, server returned HTTP status 404
URL requested:
http://watson.microsoft.com/StageOne/ApplicationA_exe/1_0_0_0/4e80a47e/unknown/0_0_0_0/bbbbbbb4
/80000003/00000000.htm?Retriage=1

FAULTING_IP:
+680b2faf0087de2c
00000000 ??              ???
```

```
EXCEPTION_RECORD:  ffffffff -- (.exr 0xffffffffffffffff)
ExceptionAddress: 00000000
   ExceptionCode: 80000003 (Break instruction exception)
  ExceptionFlags: 00000000
NumberParameters: 0

FAULTING_THREAD:  00000dac

PROCESS_NAME:  ApplicationA.exe

ERROR_CODE: (NTSTATUS) 0x80000003 - {EXCEPTION}  Breakpoint  A breakpoint has been reached.

EXCEPTION_CODE: (HRESULT) 0x80000003 (2147483651) - One or more arguments are invalid

MOD_LIST: <ANALYSIS/>

NTGLOBALFLAG:  0

APPLICATION_VERIFIER_FLAGS:  0

MANAGED_STACK: !dumpstack -EE
OS Thread Id: 0xdac (0)
Current frame:
ChildEBP RetAddr  Caller,Callee

EXCEPTION_OBJECT: !pe 161fe3c
Exception object: 0161fe3c
Exception type: System.NullReferenceException
Message: Object reference not set to an instance of an object.
InnerException: <none>
StackTrace (generated):
    SP       IP       Function

StackTraceString: <none>
HResult: 80004003

MANAGED_OBJECT: !dumpobj 1631624
Name: System.String
MethodTable: 790fc6cc
EEClass: 790fc62c
Size: 124(0x7c) bytes
 (C:\Windows\assembly\GAC_32\mscorlib\2.0.0.0__b77a5c561934e089\mscorlib.dll)
String: Object reference not set to an instance of an object.
Fields:
      MT    Field   Offset                 Type VT     Attr    Value Name
790ff7f0  4000096        4         System.Int32  1 instance       54 m_arrayLength
790ff7f0  4000097        8         System.Int32  1 instance       53 m_stringLength
790fe2dc  4000098        c          System.Char  1 instance       4f m_firstChar
790fc6cc  4000099       10        System.String  0   shared   static Empty
    >> Domain:Value  002056d0:790d7eb4 <<
7913cb00  400009a       14        System.Char[]  0   shared   static WhitespaceChars
    >> Domain:Value  002056d0:016016dc <<

EXCEPTION_MESSAGE:  Object reference not set to an instance of an object.

MANAGED_OBJECT_NAME:  System.NullReferenceException

DEFAULT_BUCKET_ID:  WRONG_SYMBOLS

PRIMARY_PROBLEM_CLASS:  WRONG_SYMBOLS
```

```
BUGCHECK_STR:  APPLICATION_FAULT_WRONG_SYMBOLS__SYSTEM.NULLREFERENCEEXCEPTION

LAST_CONTROL_TRANSFER:  from 7720b5b4 to 77ba0f34

STACK_TEXT:
0017dc74 7720b5b4 7b061e04 8edbe630 79e719d0 ntdll!KiFastSystemCallRet
0017dc78 7b061e04 8edbe630 79e719d0 0017ddc0 user32!NtUserWaitMessage+0xc
0017dd10 7b0620f1 00000000 00000004 0017dd6c System_Windows_Forms_ni+0x91e04
0017dd80 7b062293 0165789c 00000000 01654d5c System_Windows_Forms_ni+0x920f1
0017ddb0 7b2256ae 0165789c 8edbe630 79e719d0 System_Windows_Forms_ni+0x92293
0017de94 7b2251af 7b0c916b 016318ac 0161fe3c System_Windows_Forms_ni+0x2556ae
0017ded4 7b0d1b87 016051c0 7b0d2a70 7b05e61d System_Windows_Forms_ni+0x2551af
0017ed38 77211a10 001202da 00000202 00000000 System_Windows_Forms_ni+0x101b87
0017ed64 77211ae8 00b20caa 001202da 00000202 user32!InternalCallWinProc+0x23
0017eddc 77212a47 00232e4c 00b20caa 001202da user32!UserCallWinProcCheckWow+0x14b
0017ee40 77212a98 00b20caa 00000000 0017ee6c user32!DispatchMessageWorker+0x322
0017ee50 00911046 0017eee0 8edbe630 00000000 user32!DispatchMessageW+0xf
WARNING: Frame IP not in any known module. Following frames may be wrong.
0017ee6c 7b061c9a 0160c07c 00000001 01603480 0x911046
0017ef24 7b0620f1 00000000 ffffffff 7754736a System_Windows_Forms_ni+0x91c9a
0017ef94 7b062293 01606eac 1e2f0005 016061ec System_Windows_Forms_ni+0x920f1
0017efc4 7b0c7a3a 01606eac 0017f00c 01602bc0 System_Windows_Forms_ni+0x92293
0017eff0 79e826bd 0017f0c0 00000000 0017f090 System_Windows_Forms_ni+0xf7a3a
0017f070 79e8451b 0017f0c0 00000000 0017f090 mscorwks!CallDescrWorkerWithHandler+0xa3
0017f1b4 79e84403 00733068 0017f290 0017f244 mscorwks!MethodDesc::CallDescr+0x19c
0017f1cc 79e843e0 00733068 0017f290 0017f244 mscorwks!MethodDesc::CallTargetWorker+0x20
0017f1e0 79ef3e20 0017f244 8edbaaf6 00000000
mscorwks!MethodDescCallSite::CallWithValueTypes+0x18
0017f344 79ef3c19 00733008 00000001 0017f380 mscorwks!ClassLoader::RunMain+0x220
0017f5ac 79ef3aee 00000000 8edbae26 00000000 mscorwks!Assembly::ExecuteMainMethod+0xa6
0017fa7c 79ef3737 00fc0000 00000000 8edba2b2 mscorwks!SystemDomain::ExecuteMainMethod+0x398
0017facc 79ef18d1 00fc0000 8edba36a 00000000 mscorwks!ExecuteEXE+0x59
0017fb14 6e1e55ab 79ef183a 0017fb30 6e187f16 mscorwks!_CorExeMain+0x11b
0017fb20 6e187f16 00000000 6e1e0000 0017fb44 mscoreei!_CorExeMain+0x38
0017fb30 6e184de3 00000000 77563833 7ffd9000 mscoree!ShellShim__CorExeMain+0x99
0017fb38 77563833 7ffd9000 0017fb84 77b7a9bd mscoree!_CorExeMain_Exported+0x8
0017fb44 77b7a9bd 7ffd9000 001785fa 00000000 kernel32!BaseThreadInitThunk+0xe
0017fb84 00000000 00fc35ee 7ffd9000 00000000 ntdll!_RtlUserThreadStart+0x23


FOLLOWUP_IP:
System_Windows_Forms_ni+91e04
7b061e04 8b4dac          mov     ecx,dword ptr [ebp-54h]

SYMBOL_STACK_INDEX:  2

SYMBOL_NAME:  System_Windows_Forms_ni+91e04

FOLLOWUP_NAME:  MachineOwner

MODULE_NAME: System_Windows_Forms_ni

IMAGE_NAME:  System.Windows.Forms.ni.dll

DEBUG_FLR_IMAGE_TIMESTAMP:  4536f34f

STACK_COMMAND:  dt ntdll!LdrpLastDllInitializer BaseDllName ; dt ntdll!LdrpFailureData ; ~0s;
.ecxr ; kb
```

```
FAILURE_BUCKET_ID:  WRONG_SYMBOLS_80000003_System.Windows.Forms.ni.dll!Unknown

BUCKET_ID:
APPLICATION_FAULT_WRONG_SYMBOLS__SYSTEM.NULLREFERENCEEXCEPTION_System_Windows_Forms_ni+91e04

WATSON_STAGEONE_URL:
http://watson.microsoft.com/StageOne/ApplicationA_exe/1_0_0_0/4e80a47e/unknown/0_0_0_0/bbbbbbb4
/80000003/00000000.htm?Retriage=1

Followup: MachineOwner
---------
                                    /* end of WinDbg output */
```

Note: In addition to normal manual dump breakpoint error (in blue) we have .NET diagnostics (in red) pointing to a NULL object reference (similar to a NULL pointer dereference). However we don't see any managed stack trace we see on a dialog above. Now we try to get it.

8. We check if there is a .NET exception on the current thread 0:

```
0:000> !pe
Exception object: 0161fe3c
Exception type: System.NullReferenceException
Message: Object reference not set to an instance of an object.
InnerException: <none>
StackTrace (generated):
    SP       IP       Function

StackTraceString: <none>
HResult: 80004003
```

Note: We see the exception but no stack trace. We also double check that no other threads have exceptions by executing **!pe** command for each thread using **~*e** command:

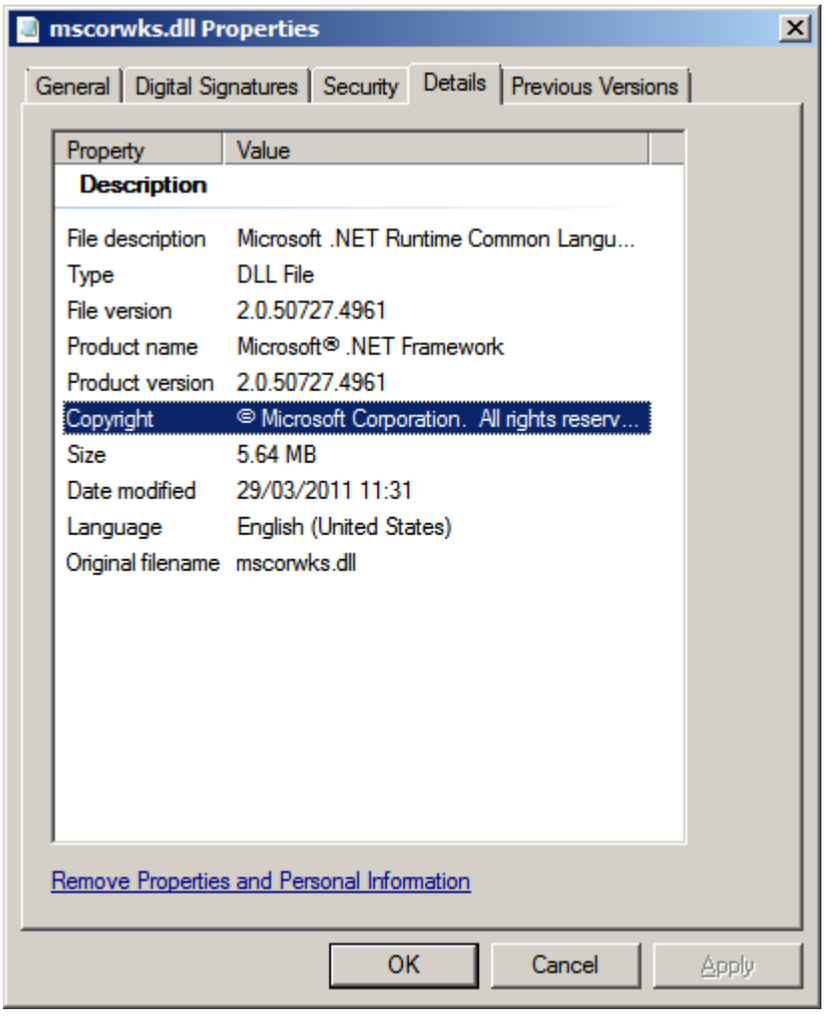
```
0:000> ~*e !pe
Exception object: 0161fe3c
Exception type: System.NullReferenceException
Message: Object reference not set to an instance of an object.
InnerException: <none>
StackTrace (generated):
    SP       IP       Function

StackTraceString: <none>
HResult: 80004003
The current thread is unmanaged
There is no current managed exception on this thread
The current thread is unmanaged
```

9. We now check the version of .NET used when ApplicationA was running:

```
0:000> lmv m mscorwks
start    end        module name
79e70000 7a3d6000   mscorwks   (pdb symbols)
c:\mss\mscorwks.pdb\777BB0057A2341BA8F3A03546FA9EB542\mscorwks.pdb
    Loaded symbol image file: mscorwks.dll
    Image path: C:\Windows\Microsoft.NET\Framework\v2.0.50727\mscorwks.dll
    Image name: mscorwks.dll
```

```
Timestamp:        Thu Oct 19 08:08:07 2006 (45372457)
CheckSum:         00562079
ImageSize:        00566000
File version:     2.0.50727.312
Product version:  2.0.50727.312
File flags:       0 (Mask 3F)
File OS:          4 Unknown Win32
File type:        2.0 Dll
File date:        00000000.00000000
Translations:     0409.04b0
CompanyName:      Microsoft Corporation
ProductName:      Microsoft® .NET Framework
InternalName:     mscorwks.dll
OriginalFilename: mscorwks.dll
ProductVersion:   2.0.50727.312
FileVersion:      2.0.50727.312 (rtmLHS.050727-3100)
FileDescription:  Microsoft .NET Runtime Common Language Runtime - WorkStation
LegalCopyright:   © Microsoft Corporation.  All rights reserved.
Comments:         Flavor=Retail
```

Note: On my analysis system the version is slightly different (C:\Windows\Microsoft.NET\Framework\v2.0.50727):



It has a different .4961 version suffix. The version can also be checked by listing all loaded WinDbg extensions (sos.dll is used for .NET analysis):

```
0:000> .chain
Extension DLL search Path:
[...]
Extension DLL chain:
    C:\Windows\Microsoft.NET\Framework\v2.0.50727\sos: image 2.0.50727.4961, API 1.0.0, built
Fri Mar 25 04:23:32 2011
        [path: C:\Windows\Microsoft.NET\Framework\v2.0.50727\sos.dll]
    dbghelp: image 6.12.0002.633, API 6.1.6, built Mon Feb 01 20:08:26 2010
        [path: C:\Program Files (x86)\Debugging Tools for Windows (x86)\dbghelp.dll]
    ext: image 6.12.0002.633, API 1.0.0, built Mon Feb 01 20:08:31 2010
        [path: C:\Program Files (x86)\Debugging Tools for Windows (x86)\winext\ext.dll]
    exts: image 6.12.0002.633, API 1.0.0, built Mon Feb 01 20:08:24 2010
        [path: C:\Program Files (x86)\Debugging Tools for Windows (x86)\WINXP\exts.dll]
    uext: image 6.12.0002.633, API 1.0.0, built Mon Feb 01 20:08:23 2010
        [path: C:\Program Files (x86)\Debugging Tools for Windows (x86)\winext\uext.dll]
    ntsdexts: image 6.1.7650.0, API 1.0.0, built Mon Feb 01 20:08:08 2010
        [path: C:\Program Files (x86)\Debugging Tools for Windows (x86)\WINXP\ntsdexts.dll]
```

10. In the folder C:\ANETMDA-Dumps\32-bit\Microsoft.NET\Framework\v2.0.50727 we have the correct version .312 of .NET Framework copied from the machine the memory dump came from. So we unload the current SOS extension (**.unload** command) and load the new one (**.load** command):

```
0:000> .unload C:\Windows\Microsoft.NET\Framework\v2.0.50727\sos
Unloading C:\Windows\Microsoft.NET\Framework\v2.0.50727\sos extension DLL

0:000> .chain
Extension DLL search Path:
[...]
Extension DLL chain:
    dbghelp: image 6.12.0002.633, API 6.1.6, built Mon Feb 01 20:08:26 2010
        [path: C:\Program Files (x86)\Debugging Tools for Windows (x86)\dbghelp.dll]
    ext: image 6.12.0002.633, API 1.0.0, built Mon Feb 01 20:08:31 2010
        [path: C:\Program Files (x86)\Debugging Tools for Windows (x86)\winext\ext.dll]
    exts: image 6.12.0002.633, API 1.0.0, built Mon Feb 01 20:08:24 2010
        [path: C:\Program Files (x86)\Debugging Tools for Windows (x86)\WINXP\exts.dll]
    uext: image 6.12.0002.633, API 1.0.0, built Mon Feb 01 20:08:23 2010
        [path: C:\Program Files (x86)\Debugging Tools for Windows (x86)\winext\uext.dll]
    ntsdexts: image 6.1.7650.0, API 1.0.0, built Mon Feb 01 20:08:08 2010
        [path: C:\Program Files (x86)\Debugging Tools for Windows (x86)\WINXP\ntsdexts.dll]

0:000> .load C:\ANETMDA-Dumps\32-bit\Microsoft.NET\Framework\v2.0.50727\sos

0:000> .chain
Extension DLL search Path:
[...]
Extension DLL chain:
    C:\ANETMDA-Dumps\32-bit\Microsoft.NET\Framework\v2.0.50727\sos: image 2.0.50727.312, API
1.0.0, built Thu Oct 19 04:37:43 2006
        [path: C:\ANETMDA-Dumps\32-bit\Microsoft.NET\Framework\v2.0.50727\sos.dll]
    dbghelp: image 6.12.0002.633, API 6.1.6, built Mon Feb 01 20:08:26 2010
        [path: C:\Program Files (x86)\Debugging Tools for Windows (x86)\dbghelp.dll]
    ext: image 6.12.0002.633, API 1.0.0, built Mon Feb 01 20:08:31 2010
        [path: C:\Program Files (x86)\Debugging Tools for Windows (x86)\winext\ext.dll]
    exts: image 6.12.0002.633, API 1.0.0, built Mon Feb 01 20:08:24 2010
```

```
        [path: C:\Program Files (x86)\Debugging Tools for Windows (x86)\WINXP\exts.dll]
    uext: image 6.12.0002.633, API 1.0.0, built Mon Feb 01 20:08:23 2010
        [path: C:\Program Files (x86)\Debugging Tools for Windows (x86)\winext\uext.dll]
    ntsdexts: image 6.1.7650.0, API 1.0.0, built Mon Feb 01 20:08:08 2010
        [path: C:\Program Files (x86)\Debugging Tools for Windows (x86)\WINXP\ntsdexts.dll]
```

11. We now retry **!pe** command:

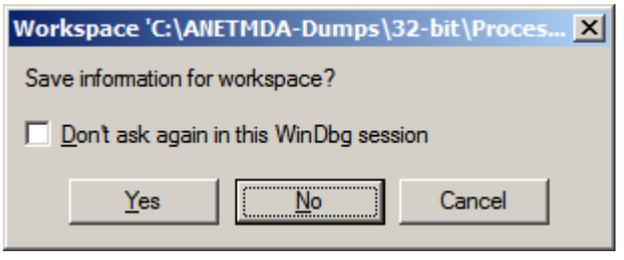
```
0:000> !pe
Exception object: 0161fe3c
Exception type: System.NullReferenceException
Message: Object reference not set to an instance of an object.
InnerException: <none>
StackTrace (generated):
    SP       IP       Function
    0017EB78 00800299 ApplicationA.Form1.button1_Click_1(System.Object, System.EventArgs)
    0017EB80 7B0693EB System.Windows.Forms.Control.OnClick(System.EventArgs)
    0017EB90 7B1065E9 System.Windows.Forms.Button.OnClick(System.EventArgs)
    0017EB9C 7B1066EF
System.Windows.Forms.Button.OnMouseUp(System.Windows.Forms.MouseEventArgs)
    0017EBC0 7B0D16EF System.Windows.Forms.Control.WmMouseUp(System.Windows.Forms.Message
ByRef, System.Windows.Forms.MouseButtons, Int32)
    0017EC1C 7B062A8F System.Windows.Forms.Control.WndProc(System.Windows.Forms.Message ByRef)
    0017EC80 7B08512B System.Windows.Forms.ButtonBase.WndProc(System.Windows.Forms.Message
ByRef)
    0017ECBC 7B08504B System.Windows.Forms.Button.WndProc(System.Windows.Forms.Message ByRef)
    0017ECC4 7B06237D
System.Windows.Forms.Control+ControlNativeWindow.OnMessage(System.Windows.Forms.Message ByRef)
    0017ECC8 7B062466
System.Windows.Forms.Control+ControlNativeWindow.WndProc(System.Windows.Forms.Message ByRef)
    0017ECDC 7B05E605 System.Windows.Forms.NativeWindow.Callback(IntPtr, Int32, IntPtr, IntPtr)

StackTraceString: <none>
HResult: 80004003
```

Note: We see now the correct managed stack trace.

12. We close logging before exiting WinDbg:

```
0:000> .logclose
Closing open log file C:\ANETMDA-Dumps\32-bit\Processes\CLR2\ApplicationA.log
```

If presented by this dialog choose No:



Note: To avoid possible confusion and glitches we recommend to exit WinDbg after each exercise.

Mac OS X Core Dump Analysis Accelerated

Version 2.0

Dmitry Vostokov
Software Diagnostics Services

Published by OpenTask, Republic of Ireland

OpenTask books and magazines are available through booksellers and distributors worldwide. For further information or comments send requests to press@opentask.com.

A CIP catalogue record for this book is available from the British Library.

ISBN-l3: 978-1-908043-71-9 (Paperback)

1st printing, 2014

Contents

Exercise A1 (GDB)

Goal: Learn how to list stack traces, disassemble functions, check their correctness, dump data, compare core dumps with diagnostic reports, get environment

Patterns: Manual Dump, Stack Trace, Stack Trace Collection, Annotated Disassembly, Paratext, Not My Version, Environment Hint

1. Load a core dump core.1394 and App1 executable:

```
$ gdb -c ~/Documents/AMCDA-Dumps/core.1394 -e ~/Documents/AMCDA-
Dumps/Apps/App1/Build/Products/Release/App1
GNU gdb 6.3.50-20050815 (Apple version gdb-1820) (Sat Jun 16 02:40:11 UTC 2012)
Copyright 2004 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty" for details.
This GDB was configured as "x86_64-apple-darwin".
Reading symbols for shared libraries . done
Reading symbols for shared libraries ..... done
#0  0x00007fff8a10ce42 in __semwait_signal ()
```

2. List all threads:

```
(gdb) info threads
  6 0x00007fff8a10ce42 in __semwait_signal ()
  5 0x00007fff8a10ce42 in __semwait_signal ()
  4 0x00007fff8a10ce42 in __semwait_signal ()
  3 0x00007fff8a10ce42 in __semwait_signal ()
  2 0x00007fff8a10ce42 in __semwait_signal ()
* 1 0x00007fff8a10ce42 in __semwait_signal ()
```

3. Get all thread stack traces:

```
(gdb) thread apply all bt

Thread 6 (core thread 5):
#0  0x00007fff8a10ce42 in __semwait_signal ()
#1  0x00007fff84d6edea in nanosleep ()
#2  0x00007fff84d6ec2c in sleep ()
#3  0x00007fff84d6ec08 in sleep ()
#4  0x0000000010390bbb2 in bar_five ()
#5  0x0000000010390bbc9 in foo_five ()
#6  0x0000000010390bbe1 in thread_five ()
#7  0x00007fff84db88bf in _pthread_start ()
#8  0x00007fff84dbbb75 in thread_start ()
```

```
Thread 5 (core thread 4):
#0  0x00007fff8a10ce42 in __semwait_signal ()
#1  0x00007fff84d6edea in nanosleep ()
#2  0x00007fff84d6ec2c in sleep ()
#3  0x00007fff84d6ec08 in sleep ()
#4  0x0000000010390bb52 in bar_four ()
#5  0x0000000010390bb69 in foo_four ()
#6  0x0000000010390bb81 in thread_four ()
#7  0x00007fff84db88bf in _pthread_start ()
#8  0x00007fff84dbbb75 in thread_start ()

Thread 4 (core thread 3):
#0  0x00007fff8a10ce42 in __semwait_signal ()
#1  0x00007fff84d6edea in nanosleep ()
#2  0x00007fff84d6ec2c in sleep ()
#3  0x00007fff84d6ec08 in sleep ()
#4  0x0000000010390baf2 in bar_three ()
#5  0x0000000010390bb09 in foo_three ()
#6  0x0000000010390bb21 in thread_three ()
#7  0x00007fff84db88bf in _pthread_start ()
#8  0x00007fff84dbbb75 in thread_start ()

Thread 3 (core thread 2):
#0  0x00007fff8a10ce42 in __semwait_signal ()
#1  0x00007fff84d6edea in nanosleep ()
#2  0x00007fff84d6ec2c in sleep ()
#3  0x00007fff84d6ec08 in sleep ()
#4  0x0000000010390ba92 in bar_two ()
#5  0x0000000010390baa9 in foo_two ()
#6  0x0000000010390bac1 in thread_two ()
---Type <return> to continue, or q <return> to quit---
#7  0x00007fff84db88bf in _pthread_start ()
#8  0x00007fff84dbbb75 in thread_start ()

Thread 2 (core thread 1):
#0  0x00007fff8a10ce42 in __semwait_signal ()
#1  0x00007fff84d6edea in nanosleep ()
#2  0x00007fff84d6ec2c in sleep ()
#3  0x00007fff84d6ec08 in sleep ()
#4  0x0000000010390ba32 in bar_one ()
#5  0x0000000010390ba49 in foo_one ()
#6  0x0000000010390ba61 in thread_one ()
#7  0x00007fff84db88bf in _pthread_start ()
#8  0x00007fff84dbbb75 in thread_start ()

Thread 1 (core thread 0):
#0  0x00007fff8a10ce42 in __semwait_signal ()
#1  0x00007fff84d6edea in nanosleep ()
#2  0x00007fff84d6ec2c in sleep ()
#3  0x00007fff84d6ec08 in sleep ()
#4  0x0000000010390bcc3 in main ()
```

4. Switch to the thread #3 and get its stack trace:

```
(gdb) thread 3
[Switching to thread 3 (core thread 2)]
0x00007fff8a10ce42 in __semwait_signal ()
```

```
(gdb) bt
#0  0x00007fff8a10ce42 in __semwait_signal ()
#1  0x00007fff84d6edea in nanosleep ()
#2  0x00007fff84d6ec2c in sleep ()
#3  0x00007fff84d6ec08 in sleep ()
#4  0x0000000010390ba92 in bar_two ()
#5  0x0000000010390baa9 in foo_two ()
#6  0x0000000010390bac1 in thread_two ()
#7  0x00007fff84db88bf in _pthread_start ()
#8  0x00007fff84dbbb75 in thread_start ()
```

5. Check that bar_two called sleep function:

```
(gdb) disassemble bar_two
Dump of assembler code for function bar_two:
0x0000000010390ba80 <bar_two+0>:  push   %rbp
0x0000000010390ba81 <bar_two+1>:  mov    %rsp,%rbp
0x0000000010390ba84 <bar_two+4>:  sub    $0x10,%rsp
0x0000000010390ba88 <bar_two+8>:  mov    $0xffffffff,%edi
0x0000000010390ba8d <bar_two+13>: callq  0x10390bce0 <dyld_stub_sleep>
0x0000000010390ba92 <bar_two+18>: mov    %eax,-0x4(%rbp)
0x0000000010390ba95 <bar_two+21>: add    $0x10,%rsp
0x0000000010390ba99 <bar_two+25>: pop    %rbp
0x0000000010390ba9a <bar_two+26>: retq
0x0000000010390ba9b <bar_two+27>: nopl   0x0(%rax,%rax,1)
End of assembler dump.
```

6. Compare with intel disassembly flavor:

```
(gdb) set disassembly-flavor intel

(gdb) disassemble bar_two
Dump of assembler code for function bar_two:
0x0000000010390ba80 <bar_two+0>:  push   rbp
0x0000000010390ba81 <bar_two+1>:  mov    rbp,rsp
0x0000000010390ba84 <bar_two+4>:  sub    rsp,0x10
0x0000000010390ba88 <bar_two+8>:  mov    edi,0xffffffff
0x0000000010390ba8d <bar_two+13>: call   0x10390bce0 <dyld_stub_sleep>
0x0000000010390ba92 <bar_two+18>: mov    DWORD PTR [rbp-0x4],eax
0x0000000010390ba95 <bar_two+21>: add    rsp,0x10
0x0000000010390ba99 <bar_two+25>: pop    rbp
0x0000000010390ba9a <bar_two+26>: ret
0x0000000010390ba9b <bar_two+27>: nop    DWORD PTR [rax+rax+0x0]
End of assembler dump.

(gdb) set disassembly-flavor att
```

7. Follow bar_two to sleep function code:

```
(gdb) disassemble bar_two
Dump of assembler code for function bar_two:
0x000000010390ba80 <bar_two+0>:  push   %rbp
0x000000010390ba81 <bar_two+1>:  mov    %rsp,%rbp
0x000000010390ba84 <bar_two+4>:  sub    $0x10,%rsp
0x000000010390ba88 <bar_two+8>:  mov    $0xffffffff,%edi
0x000000010390ba8d <bar_two+13>: callq  0x10390bce0 <dyld_stub_sleep>
0x000000010390ba92 <bar_two+18>: mov    %eax,-0x4(%rbp)
0x000000010390ba95 <bar_two+21>: add    $0x10,%rsp
0x000000010390ba99 <bar_two+25>: pop    %rbp
0x000000010390ba9a <bar_two+26>: retq
0x000000010390ba9b <bar_two+27>: nopl   0x0(%rax,%rax,1)
End of assembler dump.

(gdb) disassemble dyld_stub_sleep
Dump of assembler code for function dyld_stub_sleep:
0x000000010390bce0 <dyld_stub_sleep+0>: jmpq   *0x362(%rip)        # 0x10390c048
End of assembler dump.
```

8. Dump the annotated value as a memory address interpreting its contents as a symbol and then disassemble it:

```
(gdb) x/a 0x10390c048
0x10390c048: 0x7fff84d6ebef <sleep>


(gdb) disassemble 0x7fff84d6ebef
Dump of assembler code for function sleep:
0x00007fff84d6ebef <sleep+0>:    push   %rbp
0x00007fff84d6ebf0 <sleep+1>:    mov    %rsp,%rbp
0x00007fff84d6ebf3 <sleep+4>:    push   %rbx
0x00007fff84d6ebf4 <sleep+5>:    sub    $0x28,%rsp
0x00007fff84d6ebf8 <sleep+9>:    test   %edi,%edi
0x00007fff84d6ebfa <sleep+11>:   mov    %edi,%ebx
0x00007fff84d6ebfc <sleep+13>:   jns    0x7fff84d6ec11 <sleep+34>
0x00007fff84d6ebfe <sleep+15>:   mov    $0x7fffffff,%edi
0x00007fff84d6ec03 <sleep+20>:   callq  0x7fff84d6ebef <sleep>
0x00007fff84d6ec08 <sleep+25>:   lea    -0x7fffffff(%rbx,%rax,1),%eax
0x00007fff84d6ec0f <sleep+32>:   jmp    0x7fff84d6ec4f <sleep+96>
0x00007fff84d6ec11 <sleep+34>:   mov    %ebx,%eax
0x00007fff84d6ec13 <sleep+36>:   mov    %rax,-0x18(%rbp)
0x00007fff84d6ec17 <sleep+40>:   movq   $0x0,-0x10(%rbp)
0x00007fff84d6ec1f <sleep+48>:   lea    -0x18(%rbp),%rdi
0x00007fff84d6ec23 <sleep+52>:   lea    -0x28(%rbp),%rsi
0x00007fff84d6ec27 <sleep+56>:   callq  0x7fff84d6ed46 <nanosleep>
0x00007fff84d6ec2c <sleep+61>:   cmp    $0xffffffffffffffff,%eax
0x00007fff84d6ec2f <sleep+64>:   je     0x7fff84d6ec37 <sleep+72>
0x00007fff84d6ec31 <sleep+66>:   xor    %ebx,%ebx
0x00007fff84d6ec33 <sleep+68>:   mov    %ebx,%eax
0x00007fff84d6ec35 <sleep+70>:   jmp    0x7fff84d6ec4f <sleep+96>
0x00007fff84d6ec37 <sleep+72>:   callq  0x7fff84e0cc88 <__error>
0x00007fff84d6ec3c <sleep+77>:   cmpl   $0x4,(%rax)
0x00007fff84d6ec3f <sleep+80>:   jne    0x7fff84d6ec33 <sleep+68>
0x00007fff84d6ec41 <sleep+82>:   cmpq   $0x0,-0x20(%rbp)
0x00007fff84d6ec46 <sleep+87>:   setne  %al
0x00007fff84d6ec49 <sleep+90>:   movzbl %al,%eax
0x00007fff84d6ec4c <sleep+93>:   add    -0x28(%rbp),%eax
0x00007fff84d6ec4f <sleep+96>:   add    $0x28,%rsp
```

```
0x00007fff84d6ec53 <sleep+100>:  pop    %rbx
0x00007fff84d6ec54 <sleep+101>:  pop    %rbp
0x00007fff84d6ec55 <sleep+102>:  retq
End of assembler dump.
```

9. Repeat the same with resolving DYLD trampoline stub command:

```
(gdb) disassemble bar_two
Dump of assembler code for function bar_two:
0x0000000010390ba80 <bar_two+0>:  push   %rbp
0x0000000010390ba81 <bar_two+1>:  mov    %rsp,%rbp
0x0000000010390ba84 <bar_two+4>:  sub    $0x10,%rsp
0x0000000010390ba88 <bar_two+8>:  mov    $0xffffffff,%edi
0x0000000010390ba8d <bar_two+13>: callq  0x10390bce0 <dyld_stub_sleep>
0x0000000010390ba92 <bar_two+18>: mov    %eax,-0x4(%rbp)
0x0000000010390ba95 <bar_two+21>: add    $0x10,%rsp
0x0000000010390ba99 <bar_two+25>: pop    %rbp
0x0000000010390ba9a <bar_two+26>: retq
0x0000000010390ba9b <bar_two+27>: nopl   0x0(%rax,%rax,1)
End of assembler dump.

(gdb) info trampoline 0x10390bce0
Function at 0x10390bce0 becomes 0x7fff84d6ebef becomes 0x0
```

10. Compare stack trace for thread #3 (core thread 2) and its module info with the diagnostic report App1_1394.crash:

```
Process:         App1 [1394]
Path:            /Users/USER/Documents/*/App1
Identifier:      App1
Version:         ??? (???)
Code Type:       X86-64 (Native)
Parent Process:  bash [661]

Date/Time:       2012-07-24 00:20:26.078 +0100
OS Version:      Mac OS X 10.7.4 (11E53)
Report Version:  9

Crashed Thread:  0  Dispatch queue: com.apple.main-thread

Exception Type:  EXC_CRASH (SIGABRT)
Exception Codes: 0x0000000000000000, 0x0000000000000000

Thread 0 Crashed:: Dispatch queue: com.apple.main-thread
0   libsystem_kernel.dylib              0x00007fff8a10ce42 __semwait_signal + 10
1   libsystem_c.dylib                   0x00007fff84d6edea nanosleep + 164
2   libsystem_c.dylib                   0x00007fff84d6ec2c sleep + 61
3   libsystem_c.dylib                   0x00007fff84d6ec08 sleep + 25
4   App1                                0x0000000010390bcc3 main + 195
5   App1                                0x0000000010390ba14 start + 52

Thread 1:
0   libsystem_kernel.dylib              0x00007fff8a10ce42 __semwait_signal + 10
1   libsystem_c.dylib                   0x00007fff84d6edea nanosleep + 164
2   libsystem_c.dylib                   0x00007fff84d6ec2c sleep + 61
3   libsystem_c.dylib                   0x00007fff84d6ec08 sleep + 25
4   App1                                0x0000000010390ba32 bar_one + 18
5   App1                                0x0000000010390ba49 foo_one + 9
6   App1                                0x0000000010390ba61 thread_one + 17
7   libsystem_c.dylib                   0x00007fff84db88bf _pthread_start + 335
8   libsystem_c.dylib                   0x00007fff84dbbb75 thread_start + 13
```

```
Thread 2:
0   libsystem_kernel.dylib              0x00007fff8a10ce42 __semwait_signal + 10
1   libsystem_c.dylib                   0x00007fff84d6edea nanosleep + 164
2   libsystem_c.dylib                   0x00007fff84d6ec2c sleep + 61
3   libsystem_c.dylib                   0x00007fff84d6ec08 sleep + 25
4   App1                                0x0000000010390ba92 bar_two + 18
5   App1                                0x0000000010390baa9 foo_two + 9
6   App1                                0x0000000010390bac1 thread_two + 17
7   libsystem_c.dylib                   0x00007fff84db88bf _pthread_start + 335
8   libsystem_c.dylib                   0x00007fff84dbbb75 thread_start + 13

Thread 3:
0   libsystem_kernel.dylib              0x00007fff8a10ce42 __semwait_signal + 10
1   libsystem_c.dylib                   0x00007fff84d6edea nanosleep + 164
2   libsystem_c.dylib                   0x00007fff84d6ec2c sleep + 61
3   libsystem_c.dylib                   0x00007fff84d6ec08 sleep + 25
4   App1                                0x0000000010390baf2 bar_three + 18
5   App1                                0x0000000010390bb09 foo_three + 9
6   App1                                0x0000000010390bb21 thread_three + 17
7   libsystem_c.dylib                   0x00007fff84db88bf _pthread_start + 335
8   libsystem_c.dylib                   0x00007fff84dbbb75 thread_start + 13

Thread 4:
0   libsystem_kernel.dylib              0x00007fff8a10ce42 __semwait_signal + 10
1   libsystem_c.dylib                   0x00007fff84d6edea nanosleep + 164
2   libsystem_c.dylib                   0x00007fff84d6ec2c sleep + 61
3   libsystem_c.dylib                   0x00007fff84d6ec08 sleep + 25
4   App1                                0x0000000010390bb52 bar_four + 18
5   App1                                0x0000000010390bb69 foo_four + 9
6   App1                                0x0000000010390bb81 thread_four + 17
7   libsystem_c.dylib                   0x00007fff84db88bf _pthread_start + 335
8   libsystem_c.dylib                   0x00007fff84dbbb75 thread_start + 13

Thread 5:
0   libsystem_kernel.dylib              0x00007fff8a10ce42 __semwait_signal + 10
1   libsystem_c.dylib                   0x00007fff84d6edea nanosleep + 164
2   libsystem_c.dylib                   0x00007fff84d6ec2c sleep + 61
3   libsystem_c.dylib                   0x00007fff84d6ec08 sleep + 25
4   App1                                0x0000000010390bbb2 bar_five + 18
5   App1                                0x0000000010390bbc9 foo_five + 9
6   App1                                0x0000000010390bbe1 thread_five + 17
7   libsystem_c.dylib                   0x00007fff84db88bf _pthread_start + 335
8   libsystem_c.dylib                   0x00007fff84dbbb75 thread_start + 13

Thread 0 crashed with X86 Thread State (64-bit):
  rax: 0x0000000000000004  rbx: 0x00007fff6350aa08  rcx: 0x00007fff6350a9c8  rdx: 0x0000000000000001
  rdi: 0x0000000000000c03  rsi: 0x0000000000000000  rbp: 0x00007fff6350a9f0  rsp: 0x00007fff6350a9c8
   r8: 0x000000007fffffff   r9: 0x0000000000000000  r10: 0x0000000000000001  r11: 0xffffff80002da8d0
  r12: 0x0000000000000000  r13: 0x0000000000000000  r14: 0x00007fff6350aa18  r15: 0x0000000000000000
  rip: 0x00007fff8a10ce42  rfl: 0x0000000000000247  cr2: 0x0000000103d0b880
Logical CPU: 0

Binary Images:
       0x10390b000 -        0x10390bfff +App1 (??? - ???) <5BC0342F-7E97-3A7D-8EA6-75A0468021EA>
/Users/USER/Documents/*/App1
    0x7fff6350b000 -     0x7fff6353fbaf  dyld (195.6 - ???) <0CD1B35B-A28F-32DA-B72E-452EAD609613> /usr/lib/dyld
    0x7fff849f2000 -     0x7fff84a0ffff  libxpc.dylib (77.19.0 - compatibility 1.0.0) <9F57891B-D7EF-3050-BEDD-
21E7C6668248> /usr/lib/system/libxpc.dylib
    0x7fff84d68000 -     0x7fff84d69ff7  libsystem_blocks.dylib (53.0.0 - compatibility 1.0.0) <8BCA214A-8992-34B2-
A8B9-B74DEACA1869> /usr/lib/system/libsystem_blocks.dylib
    0x7fff84d6a000 -     0x7fff84e47fef  libsystem_c.dylib (763.13.0 - compatibility 1.0.0) <41B43515-2806-3FBC-ACF1-
A16F35B7E290> /usr/lib/system/libsystem_c.dylib
    0x7fff85022000 -     0x7fff85030fff  libdispatch.dylib (187.9.0 - compatibility 1.0.0) <1D5BE322-A9B9-3BCE-8FAC-
076FB07CF54A> /usr/lib/system/libdispatch.dylib
    0x7fff855f0000 -     0x7fff855f1fff  libunc.dylib (24.0.0 - compatibility 1.0.0) <337960EE-0A85-3DD0-A760-
7134CF4C0AFF> /usr/lib/system/libunc.dylib
    0x7fff85ae3000 -     0x7fff85ae4ff7  libremovefile.dylib (21.1.0 - compatibility 1.0.0) <739E6C83-AA52-3C6C-A680-
B37FE2888A04> /usr/lib/system/libremovefile.dylib
    0x7fff89114000 -     0x7fff89118fff  libmathCommon.A.dylib (2026.0.0 - compatibility 1.0.0) <FF83AFF7-42B2-306E-
90AF-D539C51A4542> /usr/lib/system/libmathCommon.A.dylib
    0x7fff89119000 -     0x7fff8911dfff  libdyld.dylib (195.5.0 - compatibility 1.0.0) <380C3F44-0CA7-3514-8080-
46D1C9DF4FCD> /usr/lib/system/libdyld.dylib
    0x7fff89740000 -     0x7fff89741ff7  libsystem_sandbox.dylib (??? - ???) <96D38E74-F18F-3CCB-A20B-E8E3ADC4E166>
/usr/lib/system/libsystem_sandbox.dylib
    0x7fff8a0ef000 -     0x7fff8a0f5fff  libmacho.dylib (800.0.0 - compatibility 1.0.0) <165514D7-1BFA-38EF-A151-
676DCD21FB64> /usr/lib/system/libmacho.dylib
```

```
    0x7fff8a0f6000 -     0x7fff8a116fff  libsystem_kernel.dylib (1699.26.8 - compatibility 1.0.0) <1DDC0B0F-DB2A-34D6-
895D-E5B2B5618946> /usr/lib/system/libsystem_kernel.dylib
    0x7fff8a2ac000 -     0x7fff8a2b4fff  libsystem_dnssd.dylib (??? - ???) <D9BB1F87-A42B-3CBC-9DC2-FC07FCEF0016>
/usr/lib/system/libsystem_dnssd.dylib
    0x7fff8ae26000 -     0x7fff8ae61fff  libsystem_info.dylib (??? - ???) <35F90252-2AE1-32C5-8D34-782C614D9639>
/usr/lib/system/libsystem_info.dylib
    0x7fff8b248000 -     0x7fff8b24afff  libquarantine.dylib (36.6.0 - compatibility 1.0.0) <0EBF714B-4B69-3E1F-9A7D-
6BBC2AACB310> /usr/lib/system/libquarantine.dylib
    0x7fff8b3b4000 -     0x7fff8b3b4fff  libkeymgr.dylib (23.0.0 - compatibility 1.0.0) <61EFED6A-A407-301E-B454-
CD18314F0075> /usr/lib/system/libkeymgr.dylib
    0x7fff8b3dd000 -     0x7fff8b3e2fff  libcompiler_rt.dylib (6.0.0 - compatibility 1.0.0) <98ECD5F6-E85C-32A5-98CD-
8911230CB66A> /usr/lib/system/libcompiler_rt.dylib
    0x7fff8bd1a000 -     0x7fff8bd1bfff  libdnsinfo.dylib (395.11.0 - compatibility 1.0.0) <853BAAA5-270F-3FDC-B025-
D448DB72E1C3> /usr/lib/system/libdnsinfo.dylib
    0x7fff8c528000 -     0x7fff8c52dff7  libsystem_network.dylib (??? - ???) <5DE7024E-1D2D-34A2-80F4-08326331A75B>
/usr/lib/system/libsystem_network.dylib
    0x7fff8cfa3000 -     0x7fff8cfadff7  liblaunch.dylib (392.38.0 - compatibility 1.0.0) <6ECB7F19-B384-32C1-8652-
2463C1CF4815> /usr/lib/system/liblaunch.dylib
    0x7fff8fe02000 -     0x7fff8fe09fff  libcopyfile.dylib (85.1.0 - compatibility 1.0.0) <0AB51EE2-E914-358C-AC19-
47BC024BDAE7> /usr/lib/system/libcopyfile.dylib
    0x7fff8fe4b000 -     0x7fff8fe8dff7  libcommonCrypto.dylib (55010.0.0 - compatibility 1.0.0) <BB770C22-8C57-365A-
8716-4A3C36AE7BFB> /usr/lib/system/libcommonCrypto.dylib
    0x7fff90c0f000 -     0x7fff90c18ff7  libsystem_notify.dylib (80.1.0 - compatibility 1.0.0) <A4D651E3-D1C6-3934-
AD49-7A104FD14596> /usr/lib/system/libsystem_notify.dylib
    0x7fff91376000 -     0x7fff913a3fe7  libSystem.B.dylib (159.1.0 - compatibility 1.0.0) <7BEBB139-50BB-3112-947A-
F4AA168F991C> /usr/lib/libSystem.B.dylib
    0x7fff91489000 -     0x7fff9148fff7  libunwind.dylib (30.0.0 - compatibility 1.0.0) <1E9C6C8C-CBE8-3F4B-A5B5-
E03E3AB53231> /usr/lib/system/libunwind.dylib
    0x7fff91a22000 -     0x7fff91a27fff  libcache.dylib (47.0.0 - compatibility 1.0.0) <1571C3AB-BCB2-38CD-B3B2-
C5FC3F927C6A> /usr/lib/system/libcache.dylib

External Modification Summary:
  Calls made by other processes targeting this process:
    task_for_pid: 2
    thread_create: 0
    thread_set_state: 0
  Calls made by this process:
    task_for_pid: 0
    thread_create: 0
    thread_set_state: 0
  Calls made by all processes on this machine:
    task_for_pid: 2696
    thread_create: 0
    thread_set_state: 0

VM Region Summary:
ReadOnly portion of Libraries: Total=50.2M resident=50.2M(100%) swapped_out_or_unallocated=0K(0%)
Writable regions: Total=38.9M written=10.8M(28%) resident=42.6M(110%) swapped_out=0K(0%)
unallocated=16777216.0T(45221404475392%)

REGION TYPE                      VIRTUAL
===========                      =======
MALLOC                             1220K
Stack                              66.6M
__DATA                              464K
__LINKEDIT                         47.7M
__TEXT                             2484K
shared memory                        12K
===========                      =======
TOTAL                             118.4M
```

11. Get App1 data section from the output of vmmap_1394.log:

```
Virtual Memory Map of process 1394 (App1)
Output report format:  2.2  -- 64-bit process

==== Non-writable regions for process 1394
__TEXT                 000000010390b000-000000010390c000 [    4K] r-x/rwx SM=COW  /Users/DumpAnalysis/Documents/AMCDA-
Dumps/Apps/App1/Build/Products/Release/App1

[...]

==== Writable regions for process 1394
__DATA                 000000010390c000-000000010390d000 [    4K] rw-/rwx SM=PRV  /Users/DumpAnalysis/Documents/AMCDA-
Dumps/Apps/App1/Build/Products/Release/App1

[...]
```

12. Compare with the section information in the core dump:

```
(gdb) maintenance info sections
Exec file:
    `/Users/DumpAnalysis/Documents/AMCDA-Dumps/Apps/App1/Build/Products/Release/App1', file type mach-o-le.
    0x0000000000000000->0x0000000000000000 at 0x00000000: LC_SEGMENT.__PAGEZERO ALLOC LOAD CODE HAS_CONTENTS
    0x0000000100000000->0x0000000100001000 at 0x00000000: LC_SEGMENT.__TEXT ALLOC LOAD CODE HAS_CONTENTS
    0x00000001000009e0->0x0000000100000cd3 at 0x000009e0: LC_SEGMENT.__TEXT.__text ALLOC LOAD READONLY CODE
HAS_CONTENTS
    0x0000000100000cd4->0x0000000100000ce6 at 0x00000cd4: LC_SEGMENT.__TEXT.__stubs ALLOC LOAD CODE HAS_CONTENTS
    0x0000000100000ce8->0x0000000100000d16 at 0x00000ce8: LC_SEGMENT.__TEXT.__stub_helper ALLOC LOAD CODE HAS_CONTENTS
    0x0000000100000d16->0x0000000100000d66 at 0x00000d16: LC_SEGMENT.__TEXT.__unwind_info ALLOC LOAD CODE HAS_CONTENTS
    0x0000000100000d68->0x0000000100001000 at 0x00000d68: LC_SEGMENT.__TEXT.__eh_frame ALLOC LOAD CODE HAS_CONTENTS
    0x0000000100001000->0x0000000100002000 at 0x00001000: LC_SEGMENT.__DATA ALLOC LOAD CODE HAS_CONTENTS
    0x0000000100001000->0x0000000100001028 at 0x00001000: LC_SEGMENT.__DATA.__program_vars ALLOC LOAD CODE
HAS_CONTENTS
    0x0000000100001028->0x0000000100001038 at 0x00001028: LC_SEGMENT.__DATA.__nl_symbol_ptr ALLOC LOAD CODE
HAS_CONTENTS
    0x0000000100001038->0x0000000100001050 at 0x00001038: LC_SEGMENT.__DATA.__la_symbol_ptr ALLOC LOAD CODE
HAS_CONTENTS
    0x0000000100001050->0x0000000100001070 at 0x00000000: LC_SEGMENT.__DATA.__common ALLOC
    0x0000000100002000->0x00000001000023b0 at 0x00002000: LC_SEGMENT.__LINKEDIT ALLOC LOAD CODE HAS_CONTENTS
    0x0000000000000000->0x00000000000001a0 at 0x000020d0: LC_SYMTAB.stabs HAS_CONTENTS
    0x0000000000000000->0x0000000000000120 at 0x00002290: LC_SYMTAB.stabstr HAS_CONTENTS
    0x0000000000000000->0x0000000000000100 at 0x000020d0: LC_DYSYMTAB.localstabs HAS_CONTENTS
    0x0000000000000000->0x00000000000000a0 at 0x000021d0: LC_DYSYMTAB.nonlocalstabs HAS_CONTENTS
    0x0000000000000000->0x0000000000000018 at 0x000004b0: LC_LOAD_DYLINKER HAS_CONTENTS
    0x0000000000000000->0x00000000000000a8 at 0x00000500: LC_THREAD.x86_THREAD_STATE64.0 HAS_CONTENTS
    0x0000000000000000->0x0000000000000030 at 0x000005b0: LC_LOAD_DYLIB HAS_CONTENTS
Core file:
    `/Users/DumpAnalysis/Documents/AMCDA-Dumps/core.1394', file type mach-o-le.
    0x000000010390b000->0x000000010390c000 at 0x00002000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x000000010390c000->0x000000010390d000 at 0x00003000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x000000010390d000->0x000000010390e000 at 0x00004000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x000000010390e000->0x000000010390f000 at 0x00005000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x000000010390f000->0x0000000103910000 at 0x00006000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x0000000103910000->0x0000000103911000 at 0x00007000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x0000000103911000->0x0000000103926000 at 0x00008000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x0000000103926000->0x0000000103927000 at 0x0001d000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x0000000103927000->0x0000000103928000 at 0x0001e000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x0000000103928000->0x000000010393d000 at 0x0001f000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x000000010393d000->0x000000010393e000 at 0x00034000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x000000010393e000->0x000000010393f000 at 0x00035000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x000000010393f000->0x0000000103940000 at 0x00036000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x0000000103940000->0x00000001039c2000 at 0x00037000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x0000000103a00000->0x0000000103b00000 at 0x000b9000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x0000000103b00000->0x0000000103b01000 at 0x001b9000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x0000000103b01000->0x0000000103b83000 at 0x001ba000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x0000000103b83000->0x0000000103b84000 at 0x0023c000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x0000000103b84000->0x0000000103c06000 at 0x0023d000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x0000000103c06000->0x0000000103c07000 at 0x002bf000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x0000000103c07000->0x0000000103c89000 at 0x002c0000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x0000000103c89000->0x0000000103c8a000 at 0x00342000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
    0x0000000103c8a000->0x0000000103d0c000 at 0x00343000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
```

```
0x00007fff5f50b000->0x00007fff62d0b000 at 0x003c5000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff62d0b000->0x00007fff6350a000 at 0x03bc5000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff6350a000->0x00007fff6350b000 at 0x043c4000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff6350b000->0x00007fff63540000 at 0x043c5000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff63540000->0x00007fff63542000 at 0x043fa000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff63542000->0x00007fff6357c000 at 0x043fc000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff6357c000->0x00007fff6358f000 at 0x04436000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff749b8000->0x00007fff74a00000 at 0x04449000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff74a00000->0x00007fff74c00000 at 0x04491000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff74c00000->0x00007fff74e00000 at 0x04691000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff74e00000->0x00007fff75000000 at 0x04891000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff75000000->0x00007fff75200000 at 0x04a91000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff75200000->0x00007fff75400000 at 0x04c91000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff75400000->0x00007fff75600000 at 0x04e91000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff75600000->0x00007fff75800000 at 0x05091000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff75800000->0x00007fff75a00000 at 0x05291000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff75a00000->0x00007fff75c00000 at 0x05491000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff75c00000->0x00007fff75e00000 at 0x05691000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff75e00000->0x00007fff76200000 at 0x05891000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff76200000->0x00007fff76400000 at 0x05c91000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff76400000->0x00007fff764ac000 at 0x05e91000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff849b8000->0x00007fff91a28000 at 0x05f3d000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fff91a28000->0x00007fff94b30000 at 0x12fad000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x00007fffffe00000->0x00007fffffe02000 at 0x160b5000: LC_SEGMENT. ALLOC LOAD CODE HAS_CONTENTS
0x0000000000000000->0x00000000000000b0 at 0x00000d68: LC_THREAD.x86_THREAD_STATE.0 HAS_CONTENTS
0x0000000000000000->0x0000000000000214 at 0x00000e20: LC_THREAD.x86_FLOAT_STATE.0 HAS_CONTENTS
0x0000000000000000->0x0000000000000018 at 0x0000103c: LC_THREAD.x86_EXCEPTION_STATE.0 HAS_CONTENTS
0x0000000000000000->0x00000000000000b0 at 0x00001064: LC_THREAD.x86_THREAD_STATE.1 HAS_CONTENTS
0x0000000000000000->0x0000000000000214 at 0x0000111c: LC_THREAD.x86_FLOAT_STATE.1 HAS_CONTENTS
0x0000000000000000->0x0000000000000018 at 0x00001338: LC_THREAD.x86_EXCEPTION_STATE.1 HAS_CONTENTS
0x0000000000000000->0x00000000000000b0 at 0x00001360: LC_THREAD.x86_THREAD_STATE.2 HAS_CONTENTS
0x0000000000000000->0x0000000000000214 at 0x00001418: LC_THREAD.x86_FLOAT_STATE.2 HAS_CONTENTS
0x0000000000000000->0x0000000000000018 at 0x00001634: LC_THREAD.x86_EXCEPTION_STATE.2 HAS_CONTENTS
0x0000000000000000->0x00000000000000b0 at 0x0000165c: LC_THREAD.x86_THREAD_STATE.3 HAS_CONTENTS
0x0000000000000000->0x0000000000000214 at 0x00001714: LC_THREAD.x86_FLOAT_STATE.3 HAS_CONTENTS
0x0000000000000000->0x0000000000000018 at 0x00001930: LC_THREAD.x86_EXCEPTION_STATE.3 HAS_CONTENTS
0x0000000000000000->0x00000000000000b0 at 0x00001958: LC_THREAD.x86_THREAD_STATE.4 HAS_CONTENTS
0x0000000000000000->0x0000000000000214 at 0x00001a10: LC_THREAD.x86_FLOAT_STATE.4 HAS_CONTENTS
0x0000000000000000->0x0000000000000018 at 0x00001c2c: LC_THREAD.x86_EXCEPTION_STATE.4 HAS_CONTENTS
0x0000000000000000->0x00000000000000b0 at 0x00001c54: LC_THREAD.x86_THREAD_STATE.5 HAS_CONTENTS
0x0000000000000000->0x0000000000000214 at 0x00001d0c: LC_THREAD.x86_FLOAT_STATE.5 HAS_CONTENTS
0x0000000000000000->0x0000000000000018 at 0x00001f28: LC_THREAD.x86_EXCEPTION_STATE.5 HAS_CONTENTS
```

13. Dump data with possible symbolic information:

```
 (gdb) x/512a 0x000000010390c000
0x10390c000: 0x10390b000  0x10390c050 <NXArgc>
0x10390c010: 0x10390c058 <NXArgv>      0x10390c060 <environ>
0x10390c020: 0x10390c068 <__progname>  0x7fff8911a6a0 <dyld_stub_binder>
0x10390c030: 0x7fff63546d80      0x10390bcf8
0x10390c040: 0x7fff84dbab01 <pthread_create>  0x7fff84d6ebef <sleep>
0x10390c050 <NXArgc>:     0x1    0x7fff6350aaf0
0x10390c060 <environ>:    0x7fff6350ab00      0x7fff6350ac73
0x10390c070: 0x0    0x0
0x10390c080: 0x0    0x0
0x10390c090: 0x0    0x0
0x10390c0a0: 0x0    0x0
0x10390c0b0: 0x0    0x0
0x10390c0c0: 0x0    0x0
0x10390c0d0: 0x0    0x0
0x10390c0e0: 0x0    0x0
0x10390c0f0: 0x0    0x0
0x10390c100: 0x0    0x0
0x10390c110: 0x0    0x0
0x10390c120: 0x0    0x0
0x10390c130: 0x0    0x0
0x10390c140: 0x0    0x0
```

```
0x10390c150: 0x0    0x0
0x10390c160: 0x0    0x0
0x10390c170: 0x0    0x0
0x10390c180: 0x0    0x0
0x10390c190: 0x0    0x0
0x10390c1a0: 0x0    0x0
0x10390c1b0: 0x0    0x0
0x10390c1c0: 0x0    0x0
0x10390c1d0: 0x0    0x0
0x10390c1e0: 0x0    0x0
0x10390c1f0: 0x0    0x0
0x10390c200: 0x0    0x0
0x10390c210: 0x0    0x0
0x10390c220: 0x0    0x0
0x10390c230: 0x0    0x0
0x10390c240: 0x0    0x0
0x10390c250: 0x0    0x0
0x10390c260: 0x0    0x0
0x10390c270: 0x0    0x0
0x10390c280: 0x0    0x0
0x10390c290: 0x0    0x0
---Type <return> to continue, or q <return> to quit---q
Quit
```

14. Dump the contents of memory pointed to by environ variable in null-terminated string format:

```
(gdb) x/100s 0x7fff6350ab00
[...]
0x7fff6350abd5:      ""
0x7fff6350abd6:      ""
0x7fff6350abd7:      ""
0x7fff6350abd8:      "/Users/DumpAnalysis/Documents/AMCDA-
Dumps/Apps/App1/Build/Products/Release/App1"
0x7fff6350ac28:      "/Users/DumpAnalysis/Documents/AMCDA-
Dumps/Apps/App1/Build/Products/Release/App1"
0x7fff6350ac78:      "TERM_PROGRAM=Apple_Terminal"
0x7fff6350ac94:      "TERM=xterm-256color"
0x7fff6350aca8:      "SHELL=/bin/bash"
0x7fff6350acb8:      "TMPDIR=/var/folders/ww/rmtqfhl93yj4213dnl2rqy6w0000gn/T/"
0x7fff6350acf1:      "Apple_PubSub_Socket_Render=/tmp/launch-mYEvtN/Render"
0x7fff6350ad26:      "TERM_PROGRAM_VERSION=303.2"
0x7fff6350ad41:      "TERM_SESSION_ID=2B039506-8384-4620-B354-120BE31AEA84"
0x7fff6350ad76:      "USER=DumpAnalysis"
0x7fff6350ad88:      "COMMAND_MODE=unix2003"
0x7fff6350ad9e:      "SSH_AUTH_SOCK=/tmp/launch-9sm7dH/Listeners"
0x7fff6350adc9:      "__CF_USER_TEXT_ENCODING=0x1F5:0:0"
0x7fff6350adeb:      "Apple_Ubiquity_Message=/tmp/launch-tWsFs8/Apple_Ubiquity_Message"
0x7fff6350ae2c:
"PATH=/Applications/Xcode.app/Contents/Developer/usr/bin/:/usr/bin:/bin:/usr/sbin:/sbin:/usr/lo
cal/bin:/usr/X11/bin"
0x7fff6350ae9f:      "PWD=/Users/DumpAnalysis"
0x7fff6350aeb7:      "LANG=en_IE.UTF-8"
---Type <return> to continue, or q <return> to quit---
0x7fff6350aec8:      "SHLVL=1"
0x7fff6350aed0:      "HOME=/Users/DumpAnalysis"
0x7fff6350aee9:      "LOGNAME=DumpAnalysis"
0x7fff6350aefe:      "DISPLAY=/tmp/launch-M8cgb1/org.x:0"
0x7fff6350af21:      "SECURITYSESSIONID=186af"
```

```
0x7fff6350af39:      "_=/Users/DumpAnalysis/Documents/AMCDA-
Dumps/Apps/App1/Build/Products/Release/App1"
0x7fff6350af8b:      "OLDPWD=/usr/share/man/man1"
0x7fff6350afa6:      ""
0x7fff6350afa7:      ""
0x7fff6350afa8:      "stack_guard=0x74843dc6068699c3"
0x7fff6350afc7:      "malloc_entropy=0x7406669509034332,0x71e4e2253a6d22b0"
0x7fff6350affc:      ""
0x7fff6350affd:      ""
```

15. Get the list of loaded modules:

```
(gdb) info sharedlibrary
The DYLD shared library state has been initialized from the executable's shared library information.  All symbols should be present, but the addresses of some
symbols may move when the program is executed, as DYLD may relocate library load addresses if necessary.
                                  Requested State Current State
Num Basename             Type Address         Reason | | Source
  | |                       | |                   | | | |
  1 App1                   - 0x10390b000        exec Y Y /Users/DumpAnalysis/Documents/AMCDA-Dumps/Apps/App1/Build/Products/Release/App1 at 0x10390b000
(offset 0x390b000)
                                              (objfile is) [memory object "/Users/DumpAnalysis/Documents/AMCDA-Dumps/Apps/App1/Build/Products/Release/App1" at
0x10390b000]
  2 dyld                   - 0x7fff6350b000        dyld Y Y /usr/lib/dyld at 0x7fff6350b000 (offset 0x7fff6350b001) with prefix "__dyld_"
                                              (objfile is) [memory object "/usr/lib/dyld" at 0x7fff6350b000]
  3 libSystem.B.dylib      - 0x7fff91376000        dyld Y Y /usr/lib/libSystem.B.dylib at 0x7fff91376000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/libSystem.B.dylib" at 0x7fff91376000]
  4 libcache.dylib         - 0x7fff91a22000        dyld Y Y /usr/lib/system/libcache.dylib at 0x7fff91a22000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libcache.dylib" at 0x7fff91a22000]
  5 libcommonCrypto.dylib  - 0x7fff8fe4b000        dyld Y Y /usr/lib/system/libcommonCrypto.dylib at 0x7fff8fe4b000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libcommonCrypto.dylib" at 0x7fff8fe4b000]
  6 libcompiler_rt.dylib   - 0x7fff8b3dd000        dyld Y Y /usr/lib/system/libcompiler_rt.dylib at 0x7fff8b3dd000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libcompiler_rt.dylib" at 0x7fff8b3dd000]
  7 libcopyfile.dylib      - 0x7fff8fe02000        dyld Y Y /usr/lib/system/libcopyfile.dylib at 0x7fff8fe02000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libcopyfile.dylib" at 0x7fff8fe02000]
  8 libdispatch.dylib      - 0x7fff85022000        dyld Y Y /usr/lib/system/libdispatch.dylib at 0x7fff85022000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libdispatch.dylib" at 0x7fff85022000]
  9 libdnsinfo.dylib       - 0x7fff8bd1a000        dyld Y Y /usr/lib/system/libdnsinfo.dylib at 0x7fff8bd1a000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libdnsinfo.dylib" at 0x7fff8bd1a000]
 10 libdyld.dylib          - 0x7fff89119000        dyld Y Y /usr/lib/system/libdyld.dylib at 0x7fff89119000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libdyld.dylib" at 0x7fff89119000]
 11 libkeymgr.dylib        - 0x7fff8b3b4000        dyld Y Y /usr/lib/system/libkeymgr.dylib at 0x7fff8b3b4000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libkeymgr.dylib" at 0x7fff8b3b4000]
 12 liblaunch.dylib        - 0x7fff8cfa3000        dyld Y Y /usr/lib/system/liblaunch.dylib at 0x7fff8cfa3000 (offset 0x49b800---Type <return> to continue,
or q <return> to quit---
0)
                                              (objfile is) [memory object "/usr/lib/system/liblaunch.dylib" at 0x7fff8cfa3000]
 13 libmacho.dylib         - 0x7fff8a0ef000        dyld Y Y /usr/lib/system/libmacho.dylib at 0x7fff8a0ef000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libmacho.dylib" at 0x7fff8a0ef000]
 14 libmathCommon.A.dylib  - 0x7fff89114000        dyld Y Y /usr/lib/system/libmathCommon.A.dylib at 0x7fff89114000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libmathCommon.A.dylib" at 0x7fff89114000]
 15 libquarantine.dylib    - 0x7fff8b248000        dyld Y Y /usr/lib/system/libquarantine.dylib at 0x7fff8b248000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libquarantine.dylib" at 0x7fff8b248000]
 16 libremovefile.dylib    - 0x7fff85ae3000        dyld Y Y /usr/lib/system/libremovefile.dylib at 0x7fff85ae3000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libremovefile.dylib" at 0x7fff85ae3000]
 17 libsystem_blocks.dylib - 0x7fff84d68000        dyld Y Y /usr/lib/system/libsystem_blocks.dylib at 0x7fff84d68000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libsystem_blocks.dylib" at 0x7fff84d68000]
 18 libsystem_c.dylib      - 0x7fff84d6a000        dyld Y Y /usr/lib/system/libsystem_c.dylib at 0x7fff84d6a000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libsystem_c.dylib" at 0x7fff84d6a000]
 19 libsystem_dnssd.dylib  - 0x7fff8a2ac000        dyld Y Y /usr/lib/system/libsystem_dnssd.dylib at 0x7fff8a2ac000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libsystem_dnssd.dylib" at 0x7fff8a2ac000]
 20 libsystem_info.dylib   - 0x7fff8ae26000        dyld Y Y /usr/lib/system/libsystem_info.dylib at 0x7fff8ae26000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libsystem_info.dylib" at 0x7fff8ae26000]
 21 libsystem_kernel.dylib - 0x7fff8a0f6000        dyld Y Y /usr/lib/system/libsystem_kernel.dylib at 0x7fff8a0f6000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libsystem_kernel.dylib" at 0x7fff8a0f6000]
 22 libsystem_network.dylib - 0x7fff8c528000       dyld Y Y /usr/lib/system/libsystem_network.dylib at 0x7fff8c528000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libsystem_network.dylib" at 0x7fff8c528000]
 23 libsystem_notify.dylib - 0x7fff90c0f000        dyld Y Y /usr/lib/system/libsystem_notify.dylib at 0x7fff90c0f000 (offset 0x49b8000)
---Type <return> to continue, or q <return> to quit---
                                              (objfile is) [memory object "/usr/lib/system/libsystem_notify.dylib" at 0x7fff90c0f000]
 24 libsystem_sandbox.dylib - 0x7fff89740000       dyld Y Y /usr/lib/system/libsystem_sandbox.dylib at 0x7fff89740000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libsystem_sandbox.dylib" at 0x7fff89740000]
 25 libunc.dylib           - 0x7fff855f0000        dyld Y Y /usr/lib/system/libunc.dylib at 0x7fff855f0000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libunc.dylib" at 0x7fff855f0000]
 26 libunwind.dylib        - 0x7fff91489000        dyld Y Y /usr/lib/system/libunwind.dylib at 0x7fff91489000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libunwind.dylib" at 0x7fff91489000]
 27 libxpc.dylib           - 0x7fff849f2000        dyld Y Y /usr/lib/system/libxpc.dylib at 0x7fff849f2000 (offset 0x49b8000)
                                              (objfile is) [memory object "/usr/lib/system/libxpc.dylib" at 0x7fff849f2000]
```

Exercise A1 (LLDB)

Goal: Learn how to list stack traces, disassemble functions, check their correctness, dump data, compare core dumps with diagnostic reports, get environment

Patterns: Manual Dump, Stack Trace, Stack Trace Collection, Annotated Disassembly, Paratext, Not My Version, Environment Hint

1. Load a core dump core.1394 and App1 executable:

```
$ lldb -c ~/Documents/AMCDA-Dumps/core.1394 -f ~/Documents/AMCDA-
Dumps/Apps/App1/Build/Products/Release/App1
error: core.1394 is a corrupt mach-o file: load command 46 LC_SEGMENT_64 has a fileoff +
filesize (0x160b7000) that extends beyond the end of the file (0x160b5000), the segment will
be truncated
Core file '/Users/DumpAnalysis/Documents/AMCDA-Dumps/core.1394' (x86_64) was loaded.
Process 0 stopped
* thread #1: tid = 0x0000, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
    frame #0: 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10
libsystem_kernel.dylib`__semwait_signal + 10:
-> 0x7fff8a10ce42:  jae    0x7fff8a10ce49            ; __semwait_signal + 17
   0x7fff8a10ce44:  jmpq   0x7fff8a10dffc            ; cerror
   0x7fff8a10ce49:  ret
   0x7fff8a10ce4a:  nop
  thread #2: tid = 0x0001, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
    frame #0: 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10
libsystem_kernel.dylib`__semwait_signal + 10:
-> 0x7fff8a10ce42:  jae    0x7fff8a10ce49            ; __semwait_signal + 17
   0x7fff8a10ce44:  jmpq   0x7fff8a10dffc            ; cerror
   0x7fff8a10ce49:  ret
   0x7fff8a10ce4a:  nop
  thread #3: tid = 0x0002, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
    frame #0: 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10
libsystem_kernel.dylib`__semwait_signal + 10:
-> 0x7fff8a10ce42:  jae    0x7fff8a10ce49            ; __semwait_signal + 17
   0x7fff8a10ce44:  jmpq   0x7fff8a10dffc            ; cerror
   0x7fff8a10ce49:  ret
   0x7fff8a10ce4a:  nop
  thread #4: tid = 0x0003, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
    frame #0: 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10
libsystem_kernel.dylib`__semwait_signal + 10:
-> 0x7fff8a10ce42:  jae    0x7fff8a10ce49            ; __semwait_signal + 17
   0x7fff8a10ce44:  jmpq   0x7fff8a10dffc            ; cerror
   0x7fff8a10ce49:  ret
   0x7fff8a10ce4a:  nop
```

```
  thread #5: tid = 0x0004, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
    frame #0: 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10
libsystem_kernel.dylib`__semwait_signal + 10:
-> 0x7fff8a10ce42:  jae    0x7fff8a10ce49            ; __semwait_signal + 17
   0x7fff8a10ce44:  jmpq   0x7fff8a10dffc            ; cerror
   0x7fff8a10ce49:  ret
   0x7fff8a10ce4a:  nop
  thread #6: tid = 0x0005, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
    frame #0: 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10
libsystem_kernel.dylib`__semwait_signal + 10:
-> 0x7fff8a10ce42:  jae    0x7fff8a10ce49            ; __semwait_signal + 17
   0x7fff8a10ce44:  jmpq   0x7fff8a10dffc            ; cerror
   0x7fff8a10ce49:  ret
   0x7fff8a10ce4a:  nop
(lldb)
```

Note: We see LLDB listed 6 threads with their TIDs numbered from 0. Also we have code disassembly starting from the next instruction that was to be executed if dump wasn't saved. The nice feature is annotated disassembly that shows symbolic names for jump and call destinations.

2. List all threads:

```
(lldb) thread list
Process 0 stopped
* thread #1: tid = 0x0000, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
  thread #2: tid = 0x0001, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
  thread #3: tid = 0x0002, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
  thread #4: tid = 0x0003, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
  thread #5: tid = 0x0004, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
  thread #6: tid = 0x0005, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
```

Note: Compared to GDB here threads are listed according to increasing thread number order.

3. Get all thread stack traces:

```
(lldb) thread backtrace all

* thread #1: tid = 0x0000, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
    frame #0: 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10
    frame #1: 0x00007fff84d6edea libsystem_c.dylib`nanosleep + 164
    frame #2: 0x00007fff84d6ec2c libsystem_c.dylib`sleep + 61
    frame #3: 0x00007fff84d6ec08 libsystem_c.dylib`sleep + 25
    frame #4: 0x000000010390bcc3 App1`main + 195
    frame #5: 0x000000010390ba14 App1`start + 52
```

```
  thread #2: tid = 0x0001, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
    frame #0: 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10
    frame #1: 0x00007fff84d6edea libsystem_c.dylib`nanosleep + 164
    frame #2: 0x00007fff84d6ec2c libsystem_c.dylib`sleep + 61
    frame #3: 0x00007fff84d6ec08 libsystem_c.dylib`sleep + 25
    frame #4: 0x000000010390ba32 App1`bar_one + 18
    frame #5: 0x000000010390ba49 App1`foo_one + 9
    frame #6: 0x000000010390ba61 App1`thread_one + 17
    frame #7: 0x00007fff84db88bf libsystem_c.dylib`_pthread_start + 335
    frame #8: 0x00007fff84dbbb75 libsystem_c.dylib`thread_start + 13

  thread #3: tid = 0x0002, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
    frame #0: 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10
    frame #1: 0x00007fff84d6edea libsystem_c.dylib`nanosleep + 164
    frame #2: 0x00007fff84d6ec2c libsystem_c.dylib`sleep + 61
    frame #3: 0x00007fff84d6ec08 libsystem_c.dylib`sleep + 25
    frame #4: 0x000000010390ba92 App1`bar_two + 18
    frame #5: 0x000000010390baa9 App1`foo_two + 9
    frame #6: 0x000000010390bac1 App1`thread_two + 17
    frame #7: 0x00007fff84db88bf libsystem_c.dylib`_pthread_start + 335
    frame #8: 0x00007fff84dbbb75 libsystem_c.dylib`thread_start + 13

  thread #4: tid = 0x0003, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
    frame #0: 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10
    frame #1: 0x00007fff84d6edea libsystem_c.dylib`nanosleep + 164
    frame #2: 0x00007fff84d6ec2c libsystem_c.dylib`sleep + 61
    frame #3: 0x00007fff84d6ec08 libsystem_c.dylib`sleep + 25
    frame #4: 0x000000010390baf2 App1`bar_three + 18
    frame #5: 0x000000010390bb09 App1`foo_three + 9
    frame #6: 0x000000010390bb21 App1`thread_three + 17
    frame #7: 0x00007fff84db88bf libsystem_c.dylib`_pthread_start + 335
    frame #8: 0x00007fff84dbbb75 libsystem_c.dylib`thread_start + 13

  thread #5: tid = 0x0004, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
    frame #0: 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10
    frame #1: 0x00007fff84d6edea libsystem_c.dylib`nanosleep + 164
    frame #2: 0x00007fff84d6ec2c libsystem_c.dylib`sleep + 61
    frame #3: 0x00007fff84d6ec08 libsystem_c.dylib`sleep + 25
    frame #4: 0x000000010390bb52 App1`bar_four + 18
    frame #5: 0x000000010390bb69 App1`foo_four + 9
    frame #6: 0x000000010390bb81 App1`thread_four + 17
    frame #7: 0x00007fff84db88bf libsystem_c.dylib`_pthread_start + 335
    frame #8: 0x00007fff84dbbb75 libsystem_c.dylib`thread_start + 13

  thread #6: tid = 0x0005, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
    frame #0: 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10
    frame #1: 0x00007fff84d6edea libsystem_c.dylib`nanosleep + 164
    frame #2: 0x00007fff84d6ec2c libsystem_c.dylib`sleep + 61
    frame #3: 0x00007fff84d6ec08 libsystem_c.dylib`sleep + 25
    frame #4: 0x000000010390bbb2 App1`bar_five + 18
    frame #5: 0x000000010390bbc9 App1`foo_five + 9
    frame #6: 0x000000010390bbe1 App1`thread_five + 17
    frame #7: 0x00007fff84db88bf libsystem_c.dylib`_pthread_start + 335
    frame #8: 0x00007fff84dbbb75 libsystem_c.dylib`thread_start + 13
```

4. Switch to the thread #3 and get its stack trace:

```
(lldb) thread select 3
* thread #3: tid = 0x0002, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
    frame #0: 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10
libsystem_kernel.dylib`__semwait_signal + 10:
-> 0x7fff8a10ce42:  jae    0x7fff8a10ce49            ; __semwait_signal + 17
   0x7fff8a10ce44:  jmpq   0x7fff8a10dffc            ; cerror
   0x7fff8a10ce49:  ret
   0x7fff8a10ce4a:  nop

(lldb) bt
* thread #3: tid = 0x0002, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
    frame #0: 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10
    frame #1: 0x00007fff84d6edea libsystem_c.dylib`nanosleep + 164
    frame #2: 0x00007fff84d6ec2c libsystem_c.dylib`sleep + 61
    frame #3: 0x00007fff84d6ec08 libsystem_c.dylib`sleep + 25
    frame #4: 0x000000010390ba92 App1`bar_two + 18
    frame #5: 0x000000010390baa9 App1`foo_two + 9
    frame #6: 0x000000010390bac1 App1`thread_two + 17
    frame #7: 0x00007fff84db88bf libsystem_c.dylib`_pthread_start + 335
    frame #8: 0x00007fff84dbbb75 libsystem_c.dylib`thread_start + 13
```

Note: We can also list any thread stack trace without switching to it:

```
(lldb) thread backtrace 4
  thread #4: tid = 0x0003, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
    frame #0: 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10
    frame #1: 0x00007fff84d6edea libsystem_c.dylib`nanosleep + 164
    frame #2: 0x00007fff84d6ec2c libsystem_c.dylib`sleep + 61
    frame #3: 0x00007fff84d6ec08 libsystem_c.dylib`sleep + 25
    frame #4: 0x000000010390baf2 App1`bar_three + 18
    frame #5: 0x000000010390bb09 App1`foo_three + 9
    frame #6: 0x000000010390bb21 App1`thread_three + 17
    frame #7: 0x00007fff84db88bf libsystem_c.dylib`_pthread_start + 335
    frame #8: 0x00007fff84dbbb75 libsystem_c.dylib`thread_start + 13
```

5. Check that *bar_two* called *sleep* function:

```
(lldb) di -n bar_two
App1`bar_two:
   0x10390ba80:  pushq  %rbp
   0x10390ba81:  movq   %rsp, %rbp
   0x10390ba84:  subq   $16, %rsp
   0x10390ba88:  movl   $4294967295, %edi
   0x10390ba8d:  callq  0x10390bce0               ; symbol stub for: sleep
   0x10390ba92:  movl   %eax, -4(%rbp)
   0x10390ba95:  addq   $16, %rsp
   0x10390ba99:  popq   %rbp
   0x10390ba9a:  ret
   0x10390ba9b:  nopl   (%rax,%rax)
```

```
(lldb) bt
* thread #3: tid = 0x0002, 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10,
stop reason = signal SIGSTOP
    frame #0: 0x00007fff8a10ce42 libsystem_kernel.dylib`__semwait_signal + 10
    frame #1: 0x00007fff84d6edea libsystem_c.dylib`nanosleep + 164
    frame #2: 0x00007fff84d6ec2c libsystem_c.dylib`sleep + 61
    frame #3: 0x00007fff84d6ec08 libsystem_c.dylib`sleep + 25
    frame #4: 0x000000010390ba92 App1`bar_two + 18
    frame #5: 0x000000010390baa9 App1`foo_two + 9
    frame #6: 0x000000010390bac1 App1`thread_two + 17
    frame #7: 0x00007fff84db88bf libsystem_c.dylib`_pthread_start + 335
    frame #8: 0x00007fff84dbbb75 libsystem_c.dylib`thread_start + 13
```

6. Compare with Intel disassembly flavor:

```
(lldb) settings set target.x86-disassembly-flavor intel

(lldb) di -n bar_two
App1`bar_two:
   0x10390ba80:  push   RBP
   0x10390ba81:  mov    RBP, RSP
   0x10390ba84:  sub    RSP, 16
   0x10390ba88:  mov    EDI, 4294967295
   0x10390ba8d:  call   0x10390bce0               ; symbol stub for: sleep
   0x10390ba92:  mov    DWORD PTR [RBP - 4], EAX
   0x10390ba95:  add    RSP, 16
   0x10390ba99:  pop    RBP
   0x10390ba9a:  ret
   0x10390ba9b:  nop    DWORD PTR [RAX + RAX]

(lldb) set disassembly-flavor att
(lldb)
```

7. Follow *bar_two* function to *sleep* function code:

```
(lldb) di -n bar_two
App1`bar_two:
   0x10390ba80:  pushq  %rbp
   0x10390ba81:  movq   %rsp, %rbp
   0x10390ba84:  subq   $16, %rsp
   0x10390ba88:  movl   $4294967295, %edi
   0x10390ba8d:  callq  0x10390bce0               ; symbol stub for: sleep
   0x10390ba92:  movl   %eax, -4(%rbp)
   0x10390ba95:  addq   $16, %rsp
   0x10390ba99:  popq   %rbp
   0x10390ba9a:  ret
   0x10390ba9b:  nopl   (%rax,%rax)

(lldb) di -a 0x10390bce0
App1`symbol stub for: sleep:
   0x10390bce0:  jmpq   *866(%rip)               ; (void *)0x00007fff84d6ebef: sleep
```

8. Disassemble the annotated value:

```
(lldb) di -a 0x00007fff84d6ebef
libsystem_c.dylib`sleep:
   0x7fff84d6ebef:  pushq  %rbp
   0x7fff84d6ebf0:  movq   %rsp, %rbp
   0x7fff84d6ebf3:  pushq  %rbx
   0x7fff84d6ebf4:  subq   $40, %rsp
   0x7fff84d6ebf8:  testl  %edi, %edi
   0x7fff84d6ebfa:  movl   %edi, %ebx
   0x7fff84d6ebfc:  jns    0x7fff84d6ec11            ; sleep + 34
   0x7fff84d6ebfe:  movl   $2147483647, %edi
   0x7fff84d6ec03:  callq  0x7fff84d6ebef            ; sleep
   0x7fff84d6ec08:  leal   -2147483647(%rbx,%rax), %eax
   0x7fff84d6ec0f:  jmp    0x7fff84d6ec4f            ; sleep + 96
   0x7fff84d6ec11:  movl   %ebx, %eax
   0x7fff84d6ec13:  movq   %rax, -24(%rbp)
   0x7fff84d6ec17:  movq   $0, -16(%rbp)
   0x7fff84d6ec1f:  leaq   -24(%rbp), %rdi
   0x7fff84d6ec23:  leaq   -40(%rbp), %rsi
   0x7fff84d6ec27:  callq  0x7fff84d6ed46            ; nanosleep
   0x7fff84d6ec2c:  cmpl   $-1, %eax
   0x7fff84d6ec2f:  je     0x7fff84d6ec37            ; sleep + 72
   0x7fff84d6ec31:  xorl   %ebx, %ebx
   0x7fff84d6ec33:  movl   %ebx, %eax
   0x7fff84d6ec35:  jmp    0x7fff84d6ec4f            ; sleep + 96
   0x7fff84d6ec37:  callq  0x7fff84e0cc88            ; __error
   0x7fff84d6ec3c:  cmpl   $4, (%rax)
   0x7fff84d6ec3f:  jne    0x7fff84d6ec33            ; sleep + 68
   0x7fff84d6ec41:  cmpq   $0, -32(%rbp)
   0x7fff84d6ec46:  setne  %al
   0x7fff84d6ec49:  movzbl %al, %eax
   0x7fff84d6ec4c:  addl   -40(%rbp), %eax
   0x7fff84d6ec4f:  addq   $40, %rsp
   0x7fff84d6ec53:  popq   %rbx
   0x7fff84d6ec54:  popq   %rbp
```

9. Compare stack trace for thread #3 (core thread 2) and its module info with the diagnostic report App1_1394.crash:

```
Process:         App1 [1394]
Path:            /Users/USER/Documents/*/App1
Identifier:      App1
Version:         ??? (???)
Code Type:       X86-64 (Native)
Parent Process:  bash [661]

Date/Time:       2012-07-24 00:20:26.078 +0100
OS Version:      Mac OS X 10.7.4 (11E53)
Report Version:  9

Crashed Thread:  0  Dispatch queue: com.apple.main-thread

Exception Type:  EXC_CRASH (SIGABRT)
Exception Codes: 0x0000000000000000, 0x0000000000000000

Thread 0 Crashed:: Dispatch queue: com.apple.main-thread
0   libsystem_kernel.dylib              0x00007fff8a10ce42 __semwait_signal + 10
1   libsystem_c.dylib                   0x00007fff84d6edea nanosleep + 164
2   libsystem_c.dylib                   0x00007fff84d6ec2c sleep + 61
3   libsystem_c.dylib                   0x00007fff84d6ec08 sleep + 25
4   App1                                0x000000010390bcc3 main + 195
5   App1                                0x000000010390ba14 start + 52
```

```
Thread 1:
0   libsystem_kernel.dylib              0x00007fff8a10ce42 __semwait_signal + 10
1   libsystem_c.dylib                   0x00007fff84d6edea nanosleep + 164
2   libsystem_c.dylib                   0x00007fff84d6ec2c sleep + 61
3   libsystem_c.dylib                   0x00007fff84d6ec08 sleep + 25
4   App1                                0x0000000010390ba32 bar_one + 18
5   App1                                0x0000000010390ba49 foo_one + 9
6   App1                                0x0000000010390ba61 thread_one + 17
7   libsystem_c.dylib                   0x00007fff84db88bf _pthread_start + 335
8   libsystem_c.dylib                   0x00007fff84dbbb75 thread_start + 13

Thread 2:
0   libsystem_kernel.dylib              0x00007fff8a10ce42 __semwait_signal + 10
1   libsystem_c.dylib                   0x00007fff84d6edea nanosleep + 164
2   libsystem_c.dylib                   0x00007fff84d6ec2c sleep + 61
3   libsystem_c.dylib                   0x00007fff84d6ec08 sleep + 25
4   App1                                0x0000000010390ba92 bar_two + 18
5   App1                                0x0000000010390baa9 foo_two + 9
6   App1                                0x0000000010390bac1 thread_two + 17
7   libsystem_c.dylib                   0x00007fff84db88bf _pthread_start + 335
8   libsystem_c.dylib                   0x00007fff84dbbb75 thread_start + 13

Thread 3:
0   libsystem_kernel.dylib              0x00007fff8a10ce42 __semwait_signal + 10
1   libsystem_c.dylib                   0x00007fff84d6edea nanosleep + 164
2   libsystem_c.dylib                   0x00007fff84d6ec2c sleep + 61
3   libsystem_c.dylib                   0x00007fff84d6ec08 sleep + 25
4   App1                                0x0000000010390baf2 bar_three + 18
5   App1                                0x0000000010390bb09 foo_three + 9
6   App1                                0x0000000010390bb21 thread_three + 17
7   libsystem_c.dylib                   0x00007fff84db88bf _pthread_start + 335
8   libsystem_c.dylib                   0x00007fff84dbbb75 thread_start + 13

Thread 4:
0   libsystem_kernel.dylib              0x00007fff8a10ce42 __semwait_signal + 10
1   libsystem_c.dylib                   0x00007fff84d6edea nanosleep + 164
2   libsystem_c.dylib                   0x00007fff84d6ec2c sleep + 61
3   libsystem_c.dylib                   0x00007fff84d6ec08 sleep + 25
4   App1                                0x0000000010390bb52 bar_four + 18
5   App1                                0x0000000010390bb69 foo_four + 9
6   App1                                0x0000000010390bb81 thread_four + 17
7   libsystem_c.dylib                   0x00007fff84db88bf _pthread_start + 335
8   libsystem_c.dylib                   0x00007fff84dbbb75 thread_start + 13

Thread 5:
0   libsystem_kernel.dylib              0x00007fff8a10ce42 __semwait_signal + 10
1   libsystem_c.dylib                   0x00007fff84d6edea nanosleep + 164
2   libsystem_c.dylib                   0x00007fff84d6ec2c sleep + 61
3   libsystem_c.dylib                   0x00007fff84d6ec08 sleep + 25
4   App1                                0x0000000010390bbb2 bar_five + 18
5   App1                                0x0000000010390bbc9 foo_five + 9
6   App1                                0x0000000010390bbe1 thread_five + 17
7   libsystem_c.dylib                   0x00007fff84db88bf _pthread_start + 335
8   libsystem_c.dylib                   0x00007fff84dbbb75 thread_start + 13

Thread 0 crashed with X86 Thread State (64-bit):
  rax: 0x0000000000000004  rbx: 0x00007fff6350aa08  rcx: 0x00007fff6350a9c8  rdx: 0x0000000000000001
  rdi: 0x0000000000000c03  rsi: 0x0000000000000000  rbp: 0x00007fff6350a9f0  rsp: 0x00007fff6350a9c8
   r8: 0x000000007fffffff   r9: 0x0000000000000000  r10: 0x0000000000000001  r11: 0xffffff80002da8d0
  r12: 0x0000000000000000  r13: 0x0000000000000000  r14: 0x00007fff6350aa18  r15: 0x0000000000000000
  rip: 0x00007fff8a10ce42  rfl: 0x0000000000000247  cr2: 0x0000000103d0b880
Logical CPU: 0
```

```
Binary Images:
       0x10390b000 -        0x10390bfff +App1 (??? - ???) <5BC0342F-7E97-3A7D-8EA6-75A0468021EA>
/Users/USER/Documents/*/App1
    0x7fff6350b000 -     0x7fff6353fbaf  dyld (195.6 - ???) <0CD1B35B-A28F-32DA-B72E-452EAD609613> /usr/lib/dyld
    0x7fff849f2000 -     0x7fff84a0ffff  libxpc.dylib (77.19.0 - compatibility 1.0.0) <9F57891B-D7EF-3050-BEDD-
21E7C6668248> /usr/lib/system/libxpc.dylib
    0x7fff84d68000 -     0x7fff84d69ff7  libsystem_blocks.dylib (53.0.0 - compatibility 1.0.0) <8BCA214A-8992-34B2-
A8B9-B74DEACA1869> /usr/lib/system/libsystem_blocks.dylib
    0x7fff84d6a000 -     0x7fff84e47fef  libsystem_c.dylib (763.13.0 - compatibility 1.0.0) <41B43515-2806-3FBC-ACF1-
A16F35B7E290> /usr/lib/system/libsystem_c.dylib
    0x7fff85022000 -     0x7fff85030fff  libdispatch.dylib (187.9.0 - compatibility 1.0.0) <1D5BE322-A9B9-3BCE-8FAC-
076FB07CF54A> /usr/lib/system/libdispatch.dylib
    0x7fff855f0000 -     0x7fff855f1fff  libunc.dylib (24.0.0 - compatibility 1.0.0) <337960EE-0A85-3DD0-A760-
7134CF4C0AFF> /usr/lib/system/libunc.dylib
    0x7fff85ae3000 -     0x7fff85ae4ff7  libremovefile.dylib (21.1.0 - compatibility 1.0.0) <739E6C83-AA52-3C6C-A680-
B37FE2888A04> /usr/lib/system/libremovefile.dylib
    0x7fff89114000 -     0x7fff89118fff  libmathCommon.A.dylib (2026.0.0 - compatibility 1.0.0) <FF83AFF7-42B2-306E-
90AF-D539C51A4542> /usr/lib/system/libmathCommon.A.dylib
    0x7fff89119000 -     0x7fff8911dfff  libdyld.dylib (195.5.0 - compatibility 1.0.0) <380C3F44-0CA7-3514-8080-
46D1C9DF4FCD> /usr/lib/system/libdyld.dylib
    0x7fff89740000 -     0x7fff89741ff7  libsystem_sandbox.dylib (??? - ???) <96D38E74-F18F-3CCB-A20B-E8E3ADC4E166>
/usr/lib/system/libsystem_sandbox.dylib
    0x7fff8a0ef000 -     0x7fff8a0f5fff  libmacho.dylib (800.0.0 - compatibility 1.0.0) <165514D7-1BFA-38EF-A151-
676DCD21FB64> /usr/lib/system/libmacho.dylib
    0x7fff8a0f6000 -     0x7fff8a116fff  libsystem_kernel.dylib (1699.26.8 - compatibility 1.0.0) <1DDC0B0F-DB2A-34D6-
895D-E5B2B5618946> /usr/lib/system/libsystem_kernel.dylib
    0x7fff8a2ac000 -     0x7fff8a2b4fff  libsystem_dnssd.dylib (??? - ???) <D9BB1F87-A42B-3CBC-9DC2-FC07FCEF0016>
/usr/lib/system/libsystem_dnssd.dylib
    0x7fff8ae26000 -     0x7fff8ae61fff  libsystem_info.dylib (??? - ???) <35F90252-2AE1-32C5-8D34-782C614D9639>
/usr/lib/system/libsystem_info.dylib
    0x7fff8b248000 -     0x7fff8b24afff  libquarantine.dylib (36.6.0 - compatibility 1.0.0) <0EBF714B-4B69-3E1F-9A7D-
6BBC2AACB310> /usr/lib/system/libquarantine.dylib
    0x7fff8b3b4000 -     0x7fff8b3b4fff  libkeymgr.dylib (23.0.0 - compatibility 1.0.0) <61EFED6A-A407-301E-B454-
CD18314F0075> /usr/lib/system/libkeymgr.dylib
    0x7fff8b3dd000 -     0x7fff8b3e2fff  libcompiler_rt.dylib (6.0.0 - compatibility 1.0.0) <98ECD5F6-E85C-32A5-98CD-
8911230CB66A> /usr/lib/system/libcompiler_rt.dylib
    0x7fff8bd1a000 -     0x7fff8bd1bfff  libdnsinfo.dylib (395.11.0 - compatibility 1.0.0) <853BAAA5-270F-3FDC-B025-
D448DB72E1C3> /usr/lib/system/libdnsinfo.dylib
    0x7fff8c528000 -     0x7fff8c52dff7  libsystem_network.dylib (??? - ???) <5DE7024E-1D2D-34A2-80F4-08326331A75B>
/usr/lib/system/libsystem_network.dylib
    0x7fff8cfa3000 -     0x7fff8cfadff7  liblaunch.dylib (392.38.0 - compatibility 1.0.0) <6ECB7F19-B384-32C1-8652-
2463C1CF4815> /usr/lib/system/liblaunch.dylib
    0x7fff8fe02000 -     0x7fff8fe09fff  libcopyfile.dylib (85.1.0 - compatibility 1.0.0) <0AB51EE2-E914-358C-AC19-
47BC024BDAE7> /usr/lib/system/libcopyfile.dylib
    0x7fff8fe4b000 -     0x7fff8fe8dff7  libcommonCrypto.dylib (55010.0.0 - compatibility 1.0.0) <BB770C22-8C57-365A-
8716-4A3C36AE7BFB> /usr/lib/system/libcommonCrypto.dylib
    0x7fff90c0f000 -     0x7fff90c18ff7  libsystem_notify.dylib (80.1.0 - compatibility 1.0.0) <A4D651E3-D1C6-3934-
AD49-7A104FD14596> /usr/lib/system/libsystem_notify.dylib
    0x7fff91376000 -     0x7fff913a3fe7  libSystem.B.dylib (159.1.0 - compatibility 1.0.0) <7BEBB139-50BB-3112-947A-
F4AA168F991C> /usr/lib/libSystem.B.dylib
    0x7fff91489000 -     0x7fff9148fff7  libunwind.dylib (30.0.0 - compatibility 1.0.0) <1E9C6C8C-CBE8-3F4B-A5B5-
E03E3AB53231> /usr/lib/system/libunwind.dylib
    0x7fff91a22000 -     0x7fff91a27fff  libcache.dylib (47.0.0 - compatibility 1.0.0) <1571C3AB-BCB2-38CD-B3B2-
C5FC3F927C6A> /usr/lib/system/libcache.dylib

External Modification Summary:
  Calls made by other processes targeting this process:
    task_for_pid: 2
    thread_create: 0
    thread_set_state: 0
  Calls made by this process:
    task_for_pid: 0
    thread_create: 0
    thread_set_state: 0
  Calls made by all processes on this machine:
    task_for_pid: 2696
    thread_create: 0
    thread_set_state: 0
```

```
VM Region Summary:
ReadOnly portion of Libraries: Total=50.2M resident=50.2M(100%) swapped_out_or_unallocated=0K(0%)
Writable regions: Total=38.9M written=10.8M(28%) resident=42.6M(110%) swapped_out=0K(0%)
unallocated=16777216.0T(45221404475392%)

REGION TYPE                      VIRTUAL
===========                      =======
MALLOC                             1220K
Stack                              66.6M
__DATA                              464K
__LINKEDIT                         47.7M
__TEXT                             2484K
shared memory                        12K
===========                      =======
TOTAL                             118.4M
```

10. Get App1 data section from the output of vmmap_1394.log:

```
Virtual Memory Map of process 1394 (App1)
Output report format:  2.2  -- 64-bit process

==== Non-writable regions for process 1394
__TEXT                 000000010390b000-000000010390c000 [    4K] r-x/rwx SM=COW  /Users/DumpAnalysis/Documents/AMCDA-
Dumps/Apps/App1/Build/Products/Release/App1

[...]

==== Writable regions for process 1394
__DATA                 000000010390c000-000000010390d000 [    4K] rw-/rwx SM=PRV  /Users/DumpAnalysis/Documents/AMCDA-
Dumps/Apps/App1/Build/Products/Release/App1

[...]
```

11. Compare with the section information in the core dump:

```
(lldb) image dump sections App1
Sections for '/Users/DumpAnalysis/Documents/AMCDA-Dumps/Apps/App1/Build/Products/Release/App1' (x86_64):
  SectID     Type             Load Address                                  File Off.  File Size  Flags      Section Name
  ---------- ---------------- --------------------------------------------  ---------- ---------- ---------- ----------------------------
  0x00000100 container        [0x0000000000000000-0x0000000100000000)* 0x00000000 0x00000000 0x00000000 App1.__PAGEZERO
  0x00000200 container        [0x000000010390b000-0x000000010390c000)  0x00000000 0x00001000 0x00000000 App1.__TEXT
  0x00000001 code             [0x000000010390b9e0-0x000000010390bcd3)  0x000009e0 0x000002f3 0x80000400 App1.__TEXT.__text
  0x00000002 code             [0x000000010390bcd4-0x000000010390bce6)  0x00000cd4 0x00000012 0x80000408 App1.__TEXT.__stubs
  0x00000003 code             [0x000000010390bce8-0x000000010390bd16)  0x00000ce8 0x0000002e 0x80000400 App1.__TEXT.__stub_helper
  0x00000004 code             [0x000000010390bd16-0x000000010390bd66)  0x00000d16 0x00000050 0x00000000 App1.__TEXT.__unwind_info
  0x00000005 eh-frame         [0x000000010390bd68-0x000000010390c000)  0x00000d68 0x00000298 0x00000000 App1.__TEXT.__eh_frame
  0x00000300 container        [0x000000010390c000-0x000000010390d000)  0x00001000 0x00001000 0x00000000 App1.__DATA
  0x00000006 data             [0x000000010390c000-0x000000010390c028)  0x00001000 0x00000028 0x00000000 App1.__DATA.__program_vars
  0x00000007 data-ptrs        [0x000000010390c028-0x000000010390c038)  0x00001028 0x00000010 0x00000006 App1.__DATA.__nl_symbol_ptr
  0x00000008 data-ptrs        [0x000000010390c038-0x000000010390c050)  0x00001038 0x00000018 0x00000007 App1.__DATA.__la_symbol_ptr
  0x00000009 zero-fill        [0x000000010390c050-0x000000010390c070)  0x00000000 0x00000000 0x00000001 App1.__DATA.__common
  0x00000400 container        [0x000000010390d000-0x000000010390d3b0)  0x00002000 0x000003b0 0x00000000 App1.__LINKEDIT
```

12. Dump data with possible symbolic information:

```
(lldb) x/512a 0x000000010390c000
error: Normally, 'memory read' will not read over 1024 bytes of data.
error: Please use --force to override this restriction just once.
error: or set target.max-memory-read-size if you will often need a larger limit.
```

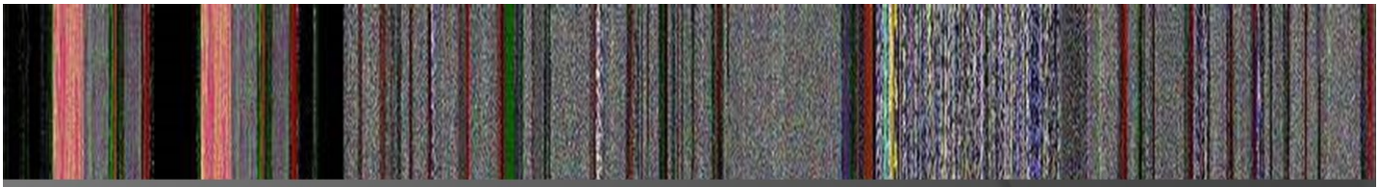
```
(lldb) x/512a 0x000000010390c000 --force
0x10390c000: 0x000000010390b000
0x10390c008: 0x000000010390c050 App1`NXArgc
0x10390c010: 0x000000010390c058 App1`NXArgv
0x10390c018: 0x000000010390c060 App1`environ
0x10390c020: 0x000000010390c068
0x10390c028: 0x00007fff8911a6a0 libdyld.dylib`dyld_stub_binder
0x10390c030: 0x00007fff63546d80 dyld`initialPoolContent + 2128
0x10390c038: 0x000000010390bcf8
0x10390c040: 0x00007fff84dbab01 libsystem_c.dylib`pthread_create
0x10390c048: 0x00007fff84d6ebef libsystem_c.dylib`sleep
0x10390c050: 0x0000000000000001
0x10390c058: 0x00007fff6350aaf0
0x10390c060: 0x00007fff6350ab00
0x10390c068: 0x00007fff6350ac73
0x10390c070: 0x0000000000000000
0x10390c078: 0x0000000000000000
0x10390c080: 0x0000000000000000
0x10390c088: 0x0000000000000000
0x10390c090: 0x0000000000000000
[...]
```

13. Dump the contents of memory pointed to by *environ* variable in null-terminated string format:

```
(lldb) x/100s 0x00007fff6350ab00
[...]
0x7fff6350abd5:  ""
0x7fff6350abd6:  ""
0x7fff6350abd7:  ""
0x7fff6350abd8:  "/Users/DumpAnalysis/Documents/AMCDA-Dumps/Apps/App1/Build/Products/Release/App1"
0x7fff6350ac28:  "/Users/DumpAnalysis/Documents/AMCDA-Dumps/Apps/App1/Build/Products/Release/App1"
0x7fff6350ac78:  "TERM_PROGRAM=Apple_Terminal"
0x7fff6350ac94:  "TERM=xterm-256color"
0x7fff6350aca8:  "SHELL=/bin/bash"
0x7fff6350acb8:  "TMPDIR=/var/folders/ww/rmtqfhl93yj4213dnl2rqy6w0000gn/T/"
0x7fff6350acf1:  "Apple_PubSub_Socket_Render=/tmp/launch-mYEvtN/Render"
0x7fff6350ad26:  "TERM_PROGRAM_VERSION=303.2"
0x7fff6350ad41:  "TERM_SESSION_ID=2B039506-8384-4620-B354-120BE31AEA84"
0x7fff6350ad76:  "USER=DumpAnalysis"
0x7fff6350ad88:  "COMMAND_MODE=unix2003"
0x7fff6350ad9e:  "SSH_AUTH_SOCK=/tmp/launch-9sm7dH/Listeners"
0x7fff6350adc9:  "__CF_USER_TEXT_ENCODING=0x1F5:0:0"
0x7fff6350adeb:  "Apple_Ubiquity_Message=/tmp/launch-tWsFs8/Apple_Ubiquity_Message"
0x7fff6350ae2c:
"PATH=/Applications/Xcode.app/Contents/Developer/usr/bin/:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/X11/bin"
0x7fff6350ae9f:  "PWD=/Users/DumpAnalysis"
0x7fff6350aeb7:  "LANG=en_IE.UTF-8"
---Type <return> to continue, or q <return> to quit---
0x7fff6350aec8:  "SHLVL=1"
0x7fff6350aed0:  "HOME=/Users/DumpAnalysis"
0x7fff6350aee9:  "LOGNAME=DumpAnalysis"
0x7fff6350aefe:  "DISPLAY=/tmp/launch-M8cgb1/org.x:0"
0x7fff6350af21:  "SECURITYSESSIONID=186af"
0x7fff6350af39:  "_=/Users/DumpAnalysis/Documents/AMCDA-Dumps/Apps/App1/Build/Products/Release/App1"
0x7fff6350af8b:  "OLDPWD=/usr/share/man/man1"
0x7fff6350afa6:  ""
0x7fff6350afa7:  ""
0x7fff6350afa8:  "stack_guard=0x74843dc6068699c3"
0x7fff6350afc7:  "malloc_entropy=0x7406669509034332,0x71e4e2253a6d22b0"
0x7fff6350affc:  ""
0x7fff6350affd:  ""
```

14. Get the list of loaded modules:

```
(lldb) image list
[  0] 5BC0342F-7E97-3A7D-8EA6-75A0468021EA 0x000000010390b000 /Users/DumpAnalysis/Documents/AMCDA-
Dumps/Apps/App1/Build/Products/Release/App1
[  1] 7BEBB139-50BB-3112-947A-F4AA168F991C 0x00007fff91376000 /usr/lib/libSystem.B.dylib (0x00007fff91376000)
[  2] 1571C3AB-BCB2-38CD-B3B2-C5FC3F927C6A 0x00007fff91a22000 /usr/lib/system/libcache.dylib (0x00007fff91a22000)
[  3] BB770C22-8C57-365A-8716-4A3C36AE7BFB 0x00007fff8fe4b000 /usr/lib/system/libcommonCrypto.dylib (0x00007fff8fe4b000)
[  4] 98ECD5F6-E85C-32A5-98CD-8911230CB66A 0x00007fff8b3dd000 /usr/lib/system/libcompiler_rt.dylib (0x00007fff8b3dd000)
[  5] 0AB51EE2-E914-358C-AC19-47BC024BDAE7 0x00007fff8fe02000 /usr/lib/system/libcopyfile.dylib (0x00007fff8fe02000)
[  6] 1D5BE322-A9B9-3BCE-8FAC-076FB07CF54A 0x00007fff85022000 /usr/lib/system/libdispatch.dylib (0x00007fff85022000)
[  7] 853BAAA5-270F-3FDC-B025-D448DB72E1C3 0x00007fff8bd1a000 /usr/lib/system/libdnsinfo.dylib (0x00007fff8bd1a000)
[  8] 380C3F44-0CA7-3514-8080-46D1C9DF4FCD 0x00007fff89119000 /usr/lib/system/libdyld.dylib (0x00007fff89119000)
[  9] 61EFED6A-A407-301E-B454-CD18314F0075 0x00007fff8b3b4000 /usr/lib/system/libkeymgr.dylib (0x00007fff8b3b4000)
[ 10] 6ECB7F19-B384-32C1-8652-2463C1CF4815 0x00007fff8cfa3000 /usr/lib/system/liblaunch.dylib (0x00007fff8cfa3000)
[ 11] 165514D7-1BFA-38EF-A151-676DCD21FB64 0x00007fff8a0ef000 /usr/lib/system/libmacho.dylib (0x00007fff8a0ef000)
[ 12] FF83AFF7-42B2-306E-90AF-D539C51A4542 0x00007fff89114000 /usr/lib/system/libmathCommon.A.dylib (0x00007fff89114000)
[ 13] 0EBF714B-4B69-3E1F-9A7D-6BBC2AACB310 0x00007fff8b248000 /usr/lib/system/libquarantine.dylib (0x00007fff8b248000)
[ 14] 739E6C83-AA52-3C6C-A680-B37FE2888A04 0x00007fff85ae3000 /usr/lib/system/libremovefile.dylib (0x00007fff85ae3000)
[ 15] 8BCA214A-8992-34B2-A8B9-B74DEACA1869 0x00007fff84d68000 /usr/lib/system/libsystem_blocks.dylib (0x00007fff84d68000)
[ 16] 41B43515-2806-3FBC-ACF1-A16F35B7E290 0x00007fff84d6a000 /usr/lib/system/libsystem_c.dylib (0x00007fff84d6a000)
[ 17] D9BB1F87-A42B-3CBC-9DC2-FC07FCEF0016 0x00007fff8a2ac000 /usr/lib/system/libsystem_dnssd.dylib (0x00007fff8a2ac000)
[ 18] 35F90252-2AE1-32C5-8D34-782C614D9639 0x00007fff8ae26000 /usr/lib/system/libsystem_info.dylib (0x00007fff8ae26000)
[ 19] 1DDC0B0F-DB2A-34D6-895D-E5B2B5618946 0x00007fff8a0f6000 /usr/lib/system/libsystem_kernel.dylib (0x00007fff8a0f6000)
[ 20] 5DE7024E-1D2D-34A2-80F4-08326331A75B 0x00007fff8c528000 /usr/lib/system/libsystem_network.dylib (0x00007fff8c528000)
[ 21] A4D651E3-D1C6-3934-AD49-7A104FD14596 0x00007fff90c0f000 /usr/lib/system/libsystem_notify.dylib (0x00007fff90c0f000)
[ 22] 96D38E74-F18F-3CCB-A20B-E8E3ADC4E166 0x00007fff89740000 /usr/lib/system/libsystem_sandbox.dylib (0x00007fff89740000)
[ 23] 337960EE-0A85-3DD0-A760-7134CF4C0AFF 0x00007fff855f0000 /usr/lib/system/libunc.dylib (0x00007fff855f0000)
[ 24] 1E9C6C8C-CBE8-3F4B-A5B5-E03E3AB53231 0x00007fff91489000 /usr/lib/system/libunwind.dylib (0x00007fff91489000)
[ 25] 9F57891B-D7EF-3050-BEDD-21E7C6668248 0x00007fff849f2000 /usr/lib/system/libxpc.dylib (0x00007fff849f2000)
[ 26] 0CD1B35B-A28F-32DA-B72E-452EAD609613 0x00007fff6350b000 /usr/lib/dyld (0x00007fff6350b000)
(lldb)
```



Windows Memory Dump Analysis

Advanced

with Data Structures

Version 2.0

Dmitry Vostokov
Software Diagnostics Services

Published by OpenTask, Republic of Ireland

OpenTask books and magazines are available through booksellers and distributors worldwide. For further information or comments send requests to press@opentask.com.

A CIP catalogue record for this book is available from the British Library.

ISBN-l3: 978-0-9558328-8-8 (Paperback)

Version 2, 2014

Contents

Exercise C1: Stack Trace Collection (64-bit)

Goal: Learn how to get stack traces related to sessions, processes and threads; diagnose different thread types; get stack traces from WOW64 processes.

Patterns: Stack Trace Collection; Passive Thread; Coupled Processes; Virtualized Process

1. Launch WinDbg from Windows Kits \ Debugging Tools for Windows (X64).

2. Open \AdvMDA-Dumps\64-bit\Complete\MEMORY-Normal.DMP

3. We get the dump file loaded:

```
Microsoft (R) Windows Debugger Version 6.3.9600.16384 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.


Loading Dump File [C:\AdvWMDA-Dumps\64-bit\Complete\MEMORY-Normal.DMP]
Kernel Complete Dump File: Full address space is available

Symbol search path is: *** Invalid ***
************************************************************************
* Symbol loading may be unreliable without a symbol search path.       *
* Use .symfix to have the debugger choose a symbol path.               *
* After setting your symbol path, use .reload to refresh symbol locations. *
************************************************************************
Executable search path is:
*****************************************************************
* Symbols can not be loaded because symbol path is not initialized. *
*                                                               *
* The Symbol Path can be set by:                                *
*   using the _NT_SYMBOL_PATH environment variable.             *
*   using the -y <symbol_path> argument when starting the debugger. *
*   using .sympath and .sympath+                                *
*****************************************************************
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for ntkrnlmp.exe -
Windows Vista Kernel Version 6000 MP (2 procs) Free x64
Product: WinNt, suite: TerminalServer SingleUserTS Personal
Built by: 6000.16386.amd64fre.vista_rtm.061101-2205
Machine Name:
Kernel base = 0xfffff800`01800000 PsLoadedModuleList = 0xfffff800`01999e90
Debug session time: Tue Jul 12 16:18:12.325 2011 (UTC + 0:00)
System Uptime: 0 days 0:11:03.409
*****************************************************************
* Symbols can not be loaded because symbol path is not initialized. *
*                                                               *
* The Symbol Path can be set by:                                *
*   using the _NT_SYMBOL_PATH environment variable.             *
*   using the -y <symbol_path> argument when starting the debugger. *
*   using .sympath and .sympath+                                *
*****************************************************************
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for ntkrnlmp.exe -
Loading Kernel Symbols
...............................................................
...............................................................
.........
```

```
Loading User Symbols
.................
Loading unloaded module list
.........*** ERROR: Symbol file could not be found.  Defaulted to export symbols for ntdll.dll
-


************* Symbol Loading Error Summary **************
Module name            Error
ntkrnlmp               The system cannot find the file specified
ntdll                  The system cannot find the file specified

You can troubleshoot most symbol related issues by turning on symbol loading diagnostics (!sym
noisy) and repeating the command that caused symbols to be loaded.
You should also verify that your symbol search path (.sympath) is correct.
*******************************************************************************
*                                                                             *
*                        Bugcheck Analysis                                    *
*                                                                             *
*******************************************************************************

Use !analyze -v to get detailed debugging information.

BugCheck D1, {fffff88002c04800, 2, 0, fffff9800e6e117a}

*** ERROR: Module load completed but symbols could not be loaded for myfault.sys
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for kernel32.dll -
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for USER32.dll -
***** Kernel symbols are WRONG. Please fix symbols to do analysis.

*******************************************************************************
***                                                                         ***
***                                                                         ***
***    Either you specified an unqualified symbol, or your debugger         ***
***    doesn't have full symbol information.  Unqualified symbol            ***
***    resolution is turned off by default. Please either specify a         ***
***    fully qualified symbol module!symbolname, or enable resolution       ***
***    of unqualified symbols by typing ".symopt- 100". Note that   ***
***    enabling unqualified symbol resolution with network symbol           ***
***    server shares in the symbol path may cause the debugger to           ***
***    appear to hang for long periods of time when an incorrect            ***
***    symbol name is typed or the network symbol server is down.           ***
***                                                                         ***
***    For some commands to work properly, your symbol path                 ***
***    must point to .pdb files that have full type information.            ***
***                                                                         ***
***    Certain .pdb files (such as the public OS symbols) do not            ***
***    contain the required information.  Contact the group that            ***
***    provided you with these symbols if you need this command to          ***
***    work.                                                                ***
***                                                                         ***
***    Type referenced: nt!_KPRCB                                           ***
***                                                                         ***
*******************************************************************************
*******************************************************************************
***                                                                         ***
***                                                                         ***
***    Either you specified an unqualified symbol, or your debugger         ***
***    doesn't have full symbol information.  Unqualified symbol            ***
***    resolution is turned off by default. Please either specify a         ***
```

```
***    fully qualified symbol module!symbolname, or enable resolution ***
***    of unqualified symbols by typing ".symopt- 100". Note that   ***
***    enabling unqualified symbol resolution with network symbol     ***
***    server shares in the symbol path may cause the debugger to     ***
***    appear to hang for long periods of time when an incorrect      ***
***    symbol name is typed or the network symbol server is down.     ***
***                                                                   ***
***    For some commands to work properly, your symbol path           ***
***    must point to .pdb files that have full type information.      ***
***                                                                   ***
***    Certain .pdb files (such as the public OS symbols) do not      ***
***    contain the required information.  Contact the group that      ***
***    provided you with these symbols if you need this command to    ***
***    work.                                                          ***
***                                                                   ***
***    Type referenced: nt!KPRCB                                      ***
***                                                                   ***
*************************************************************************
*************************************************************************
***                                                                   ***
***                                                                   ***
***    Either you specified an unqualified symbol, or your debugger   ***
***    doesn't have full symbol information.  Unqualified symbol      ***
***    resolution is turned off by default. Please either specify a   ***
***    fully qualified symbol module!symbolname, or enable resolution ***
***    of unqualified symbols by typing ".symopt- 100". Note that   ***
***    enabling unqualified symbol resolution with network symbol     ***
***    server shares in the symbol path may cause the debugger to     ***
***    appear to hang for long periods of time when an incorrect      ***
***    symbol name is typed or the network symbol server is down.     ***
***                                                                   ***
***    For some commands to work properly, your symbol path           ***
***    must point to .pdb files that have full type information.      ***
***                                                                   ***
***    Certain .pdb files (such as the public OS symbols) do not      ***
***    contain the required information.  Contact the group that      ***
***    provided you with these symbols if you need this command to    ***
***    work.                                                          ***
***                                                                   ***
***    Type referenced: nt!_KPRCB                                     ***
***                                                                   ***
*************************************************************************
*************************************************************************
***                                                                   ***
***                                                                   ***
***    Either you specified an unqualified symbol, or your debugger   ***
***    doesn't have full symbol information.  Unqualified symbol      ***
***    resolution is turned off by default. Please either specify a   ***
***    fully qualified symbol module!symbolname, or enable resolution ***
***    of unqualified symbols by typing ".symopt- 100". Note that   ***
***    enabling unqualified symbol resolution with network symbol     ***
***    server shares in the symbol path may cause the debugger to     ***
***    appear to hang for long periods of time when an incorrect      ***
***    symbol name is typed or the network symbol server is down.     ***
***                                                                   ***
***    For some commands to work properly, your symbol path           ***
***    must point to .pdb files that have full type information.      ***
***                                                                   ***
***    Certain .pdb files (such as the public OS symbols) do not      ***
***    contain the required information.  Contact the group that      ***
```

```
***    provided you with these symbols if you need this command to    ***
***    work.                                                          ***
***                                                                   ***
***    Type referenced: nt!KPRCB                                      ***
***                                                                   ***
*************************************************************************
*************************************************************************
***                                                                   ***
***                                                                   ***
***    Either you specified an unqualified symbol, or your debugger   ***
***    doesn't have full symbol information.  Unqualified symbol      ***
***    resolution is turned off by default. Please either specify a   ***
***    fully qualified symbol module!symbolname, or enable resolution ***
***    of unqualified symbols by typing ".symopt- 100". Note that   ***
***    enabling unqualified symbol resolution with network symbol     ***
***    server shares in the symbol path may cause the debugger to     ***
***    appear to hang for long periods of time when an incorrect      ***
***    symbol name is typed or the network symbol server is down.     ***
***                                                                   ***
***    For some commands to work properly, your symbol path           ***
***    must point to .pdb files that have full type information.      ***
***                                                                   ***
***    Certain .pdb files (such as the public OS symbols) do not      ***
***    contain the required information.  Contact the group that      ***
***    provided you with these symbols if you need this command to    ***
***    work.                                                          ***
***                                                                   ***
***    Type referenced: nt!_KPRCB                                     ***
***                                                                   ***
*************************************************************************
*************************************************************************
***                                                                   ***
***                                                                   ***
***    Either you specified an unqualified symbol, or your debugger   ***
***    doesn't have full symbol information.  Unqualified symbol      ***
***    resolution is turned off by default. Please either specify a   ***
***    fully qualified symbol module!symbolname, or enable resolution ***
***    of unqualified symbols by typing ".symopt- 100". Note that   ***
***    enabling unqualified symbol resolution with network symbol     ***
***    server shares in the symbol path may cause the debugger to     ***
***    appear to hang for long periods of time when an incorrect      ***
***    symbol name is typed or the network symbol server is down.     ***
***                                                                   ***
***    For some commands to work properly, your symbol path           ***
***    must point to .pdb files that have full type information.      ***
***                                                                   ***
***    Certain .pdb files (such as the public OS symbols) do not      ***
***    contain the required information.  Contact the group that      ***
***    provided you with these symbols if you need this command to    ***
***    work.                                                          ***
***                                                                   ***
***    Type referenced: nt!_KPRCB                                     ***
***                                                                   ***
*************************************************************************
*************************************************************************
***                                                                   ***
***                                                                   ***
***    Either you specified an unqualified symbol, or your debugger   ***
***    doesn't have full symbol information.  Unqualified symbol      ***
***    resolution is turned off by default. Please either specify a   ***
```

```
***    fully qualified symbol module!symbolname, or enable resolution ***
***    of unqualified symbols by typing ".symopt- 100". Note that   ***
***    enabling unqualified symbol resolution with network symbol     ***
***    server shares in the symbol path may cause the debugger to     ***
***    appear to hang for long periods of time when an incorrect      ***
***    symbol name is typed or the network symbol server is down.     ***
***                                                                   ***
***    For some commands to work properly, your symbol path           ***
***    must point to .pdb files that have full type information.      ***
***                                                                   ***
***    Certain .pdb files (such as the public OS symbols) do not      ***
***    contain the required information.  Contact the group that      ***
***    provided you with these symbols if you need this command to    ***
***    work.                                                          ***
***                                                                   ***
***    Type referenced: nt!_KPRCB                                     ***
***                                                                   ***
*************************************************************************
*************************************************************************
***                                                                   ***
***                                                                   ***
***    Either you specified an unqualified symbol, or your debugger   ***
***    doesn't have full symbol information.  Unqualified symbol      ***
***    resolution is turned off by default. Please either specify a   ***
***    fully qualified symbol module!symbolname, or enable resolution ***
***    of unqualified symbols by typing ".symopt- 100". Note that   ***
***    enabling unqualified symbol resolution with network symbol     ***
***    server shares in the symbol path may cause the debugger to     ***
***    appear to hang for long periods of time when an incorrect      ***
***    symbol name is typed or the network symbol server is down.     ***
***                                                                   ***
***    For some commands to work properly, your symbol path           ***
***    must point to .pdb files that have full type information.      ***
***                                                                   ***
***    Certain .pdb files (such as the public OS symbols) do not      ***
***    contain the required information.  Contact the group that      ***
***    provided you with these symbols if you need this command to    ***
***    work.                                                          ***
***                                                                   ***
***    Type referenced: nt!_KPRCB                                     ***
***                                                                   ***
*************************************************************************
Probably caused by : myfault.sys ( myfault+117a )

Followup: MachineOwner
---------
```

Note: Probably caused by myfault.sys. We used NotMyFault tool from Windows Internals

http://technet.microsoft.com/en-us/sysinternals/bb963901

http://download.sysinternals.com/files/NotMyFault.zip

4. We open a log file, set up symbols and reload them:

```
1: kd> .logopen C:\AdvWMDA-Dumps\64-bit\Complete\C1.log
Opened log file 'C:\AdvWMDA-Dumps\64-bit\Complete\C1.log'

1: kd> .symfix c:\mss
```

```
1: kd> .reload
Loading Kernel Symbols
...............................................................
...............................................................
.........
Loading User Symbols
.................
Loading unloaded module list
.........Unable to enumerate user-mode unloaded modules, NTSTATUS 0xC0000147
```

5. We list running sessions:

```
1: kd> !session
Sessions on machine: 2
Valid Sessions: 0 1
Current Session 1
```

6. We check the current process:

```
1: kd> !process
PROCESS fffffa8001412040
    SessionId: 1  Cid: 03bc    Peb: 7fffffd4000  ParentCid: 0734
    DirBase: 16e27000  ObjectTable: fffff8800327b5e0  HandleCount:  49.
    Image: NotMyfault.exe
    VadRoot fffffa80039b22c0 Vads 50 Clone 0 Private 354. Modified 4. Locked 0.
    DeviceMap fffff88001a8a970
    Token                             fffff88002ed5060
    ElapsedTime                       00:00:02.973
    UserTime                          00:00:00.000
    KernelTime                        00:00:00.000
    QuotaPoolUsage[PagedPool]         134936
    QuotaPoolUsage[NonPagedPool]      4704
    Working Set Sizes (now,min,max)  (1296, 50, 345) (5184KB, 200KB, 1380KB)
    PeakWorkingSetSize                1296
    VirtualSize                       69 Mb
    PeakVirtualSize                   69 Mb
    PageFaultCount                    1310
    MemoryPriority                    BACKGROUND
    BasePriority                      8
    CommitCharge                      478

        THREAD fffffa8001220060  Cid 03bc.0b34  Teb: 000007fffffde000 Win32Thread:
fffff900c30cd460 RUNNING on processor 1
```

7. We set the current session 0 and examine its implicit process:

```
1: kd> !session -s 0
Sessions on machine: 2
Implicit process is now fffffa80`023cba80
Using session 0
```

```
1: kd> !process fffffa80`023cba80 3f
PROCESS fffffa80023cba80
    SessionId: 0  Cid: 01a4    Peb: 7fffffd6000  ParentCid: 0198
    DirBase: 263d2000  ObjectTable: fffff880014e74e0  HandleCount: 527.
    Image: csrss.exe
    VadRoot fffffa8002437b20 Vads 108 Clone 0 Private 393. Modified 321. Locked 0.
    DeviceMap fffff88000007790
    Token                             fffff880014e7640
    ElapsedTime                       00:10:17.528
    UserTime                          00:00:00.046
    KernelTime                        00:00:01.187
    QuotaPoolUsage[PagedPool]         262912
    QuotaPoolUsage[NonPagedPool]      10400
    Working Set Sizes (now,min,max)  (1151, 50, 345) (4604KB, 200KB, 1380KB)
    PeakWorkingSetSize                1499
    VirtualSize                       109 Mb
    PeakVirtualSize                   116 Mb
    PageFaultCount                    3547
    MemoryPriority                    BACKGROUND
    BasePriority                      13
    CommitCharge                      590

    PEB at 000007fffffd6000
    InheritedAddressSpace:    No
    ReadImageFileExecOptions: No
    BeingDebugged:            No
    ImageBaseAddress:         0000000049740000
    Ldr                       00000000777af980
    Ldr.Initialized:          Yes
    Ldr.InInitializationOrderModuleList: 00000000002d20d0 . 00000000002face0
    Ldr.InLoadOrderModuleList:           00000000002d1fe0 . 00000000002facc0
    Ldr.InMemoryOrderModuleList:         00000000002d1ff0 . 00000000002facd0
            Base TimeStamp                     Module
        49740000 4549b4cc Nov 02 09:05:16 2006 C:\Windows\system32\csrss.exe
        776a0000 4549d372 Nov 02 11:16:02 2006 C:\Windows\system32\ntdll.dll
     7fefdd40000 4549d353 Nov 02 11:15:31 2006 C:\Windows\system32\CSRSRV.dll
     7fefdd20000 4549d24d Nov 02 11:11:09 2006 C:\Windows\system32\basesrv.dll
     7fefdcb0000 4549d37e Nov 02 11:16:14 2006 C:\Windows\system32\winsrv.dll
        775d0000 4549d334 Nov 02 11:15:00 2006 C:\Windows\system32\USER32.dll
        77490000 4549d328 Nov 02 11:14:48 2006 C:\Windows\system32\KERNEL32.dll
     7fefe8f0000 4549d273 Nov 02 11:11:47 2006 C:\Windows\system32\GDI32.dll
     7fefe260000 4549d267 Nov 02 11:11:35 2006 C:\Windows\system32\ADVAPI32.dll
     7fefeb90000 4549d31a Nov 02 11:14:34 2006 C:\Windows\system32\RPCRT4.dll
     7fefeb80000 4549d2a1 Nov 02 11:12:33 2006 C:\Windows\system32\LPK.DLL
     7fefe380000 4549d337 Nov 02 11:15:03 2006 C:\Windows\system32\USP10.dll
     7fefead0000 4549d2e1 Nov 02 11:13:37 2006 C:\Windows\system32\msvcrt.dll
     7fefda80000 4549d32f Nov 02 11:14:55 2006 C:\Windows\system32\sxs.dll
    SubSystemData:     0000000000000000
    ProcessHeap:       00000000002d0000
    ProcessParameters: 00000000002d16d0
    CurrentDirectory:  'C:\Windows\system32\'
    WindowTitle:  '< Name not readable >'
    ImageFile:    'C:\Windows\system32\csrss.exe'
    CommandLine:  'C:\Windows\system32\csrss.exe ObjectDirectory=\Windows
SharedSection=1024,20480,768 Windows=On SubSystemType=Windows ServerDll=basesrv,1
ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=winsrv:ConServerDllInitialization,2
ProfileControl=Off MaxRequestThreads=16'
    DllPath:      'C:\Windows\system32;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem'
    Environment:  00000000002d1310
        ComSpec=C:\Windows\system32\cmd.exe
```

```
        FP_NO_HOST_CHECK=NO
        NUMBER_OF_PROCESSORS=2
        OS=Windows_NT
        Path=C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem
        PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
        PROCESSOR_ARCHITECTURE=AMD64
        PROCESSOR_IDENTIFIER=EM64T Family 6 Model 15 Stepping 11, GenuineIntel
        PROCESSOR_LEVEL=6
        PROCESSOR_REVISION=0f0b
        SystemDrive=C:
        SystemRoot=C:\Windows
        TEMP=C:\Windows\TEMP
        TMP=C:\Windows\TEMP
        USERNAME=SYSTEM
        windir=C:\Windows


        THREAD fffffa800243f710  Cid 01a4.01b4  Teb: 000007fffffdc000 Win32Thread:
fffff900c07e2b20 WAIT: (WrLpcReply) UserMode Non-Alertable
            fffffa800243faa0  Semaphore Limit 0x1
        Waiting for reply to ALPC Message fffff88001b41ad0
        Not impersonating
        DeviceMap                 fffff88000007790
        Owning Process            fffffa80023cba80       Image:         csrss.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      6897           Ticks: 35561 (0:00:09:15.640)
        Context Switch Count      7              IdealProcessor: 0                 LargeStack
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address winsrv!TerminalServerRequestThread (0x000007fefdcb9ac0)
        Stack Init fffff9800e0f3db0 Current fffff9800e0f3760
        Base fffff9800e0f4000 Limit fffff9800e0ee000 Call 0
        Priority 15 BasePriority 15 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Kernel stack not resident.
        Child-SP          RetAddr           Call Site
        fffff980`0e0f37a0 fffff800`0185d695 nt!KiSwapContext+0x84
        fffff980`0e0f38e0 fffff800`0185d3dd nt!KiSwapThread+0x125
        fffff980`0e0f3940 fffff800`0188fd77 nt!KeWaitForSingleObject+0x5f5
        fffff980`0e0f39c0 fffff800`01abed94 nt!AlpcpSignalAndWait+0x97
        fffff980`0e0f3a00 fffff800`01a9ab3a nt!AlpcpReceiveSynchronousReply+0x44
        fffff980`0e0f3a60 fffff800`01aa6ac2 nt!AlpcpProcessSynchronousRequest+0x257
        fffff980`0e0f3b80 fffff800`01a9f1cd nt!LpcpRequestWaitReplyPort+0x91
        fffff980`0e0f3be0 fffff800`0184dcf3 nt!NtRequestWaitReplyPort+0x6d
        fffff980`0e0f3c20 00000000`776f049a nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @
fffff980`0e0f3c20)
        00000000`0144fa88 000007fe`fdcb9c30 ntdll!NtRequestWaitReplyPort+0xa
        00000000`0144fa90 00000000`776cb332 winsrv!TerminalServerRequestThread+0x256
        00000000`0144fca0 00000000`00000000 ntdll!RtlUserThreadStart+0x29
```

```
        THREAD fffffa800243bbb0  Cid 01a4.01b8  Teb: 000007fffffda000 Win32Thread:
fffff900c0784350 WAIT: (UserRequest) UserMode Alertable
            fffffa80023cb8d0  SynchronizationEvent
            fffffa8002413d40  SynchronizationEvent
            fffffa80023cb930  SynchronizationEvent
        Not impersonating
        DeviceMap                 fffff88000007790
        Owning Process            fffffa80023cba80       Image:         csrss.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      10122          Ticks: 32336 (0:00:08:25.250)
        Context Switch Count      7              IdealProcessor: 1                 LargeStack
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address winsrv!NotificationThread (0x000007fefdcb9e10)
        Stack Init fffff98007f7fdb0 Current fffff98007f7f260
        Base fffff98007f80000 Limit fffff98007f7a000 Call 0
        Priority 14 BasePriority 13 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Kernel stack not resident.
        Child-SP          RetAddr           Call Site
        fffff980`07f7f2a0 fffff800`0185d695 nt!KiSwapContext+0x84
        fffff980`07f7f3e0 fffff800`0185ad2f nt!KiSwapThread+0x125
        fffff980`07f7f440 fffff800`01ac1813 nt!KeWaitForMultipleObjects+0x703
        fffff980`07f7f4b0 fffff800`01ac1a03 nt!ObpWaitForMultipleObjects+0x216
        fffff980`07f7f960 fffff800`0184dcf3 nt!NtWaitForMultipleObjects+0xe2
        fffff980`07f7fbb0 00000000`776f082a nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @
fffff980`07f7fc20)
        00000000`014ef708 000007fe`fdcba003 ntdll!NtWaitForMultipleObjects+0xa
        00000000`014ef710 00000000`776cb332 winsrv!NotificationThread+0x1ee
        00000000`014efa20 00000000`00000000 ntdll!RtlUserThreadStart+0x29

        THREAD fffffa80024437c0  Cid 01a4.01bc  Teb: 000007fffffd8000 Win32Thread:
fffff900c0096d60 WAIT: (WrLpcReceive) UserMode Non-Alertable
            fffffa8002443b50  Semaphore Limit 0x1
        Not impersonating
        DeviceMap                 fffff88000007790
        Owning Process            fffffa80023cba80       Image:         csrss.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      42400          Ticks: 58 (0:00:00:00.906)
        Context Switch Count      1272           IdealProcessor: 0                 LargeStack
        UserTime                  00:00:00.218
        KernelTime                00:00:00.468
        Win32 Start Address CSRSRV!CsrApiRequestThread (0x000007fefdd45f8c)
        Stack Init fffff98004e79db0 Current fffff98004e797a0
        Base fffff98004e7a000 Limit fffff98004e72000 Call 0
        Priority 14 BasePriority 13 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Child-SP          RetAddr           Call Site
        fffff980`04e797e0 fffff800`0185d695 nt!KiSwapContext+0x84
        fffff980`04e79920 fffff800`0185d3dd nt!KiSwapThread+0x125
        fffff980`04e79980 fffff800`01ab3a98 nt!KeWaitForSingleObject+0x5f5
        fffff980`04e79a00 fffff800`01ab41ea nt!AlpcpReceiveMessagePort+0x298
        fffff980`04e79a60 fffff800`01ab74ea nt!AlpcpReceiveMessage+0x246
        fffff980`04e79b00 fffff800`0184dcf3 nt!NtAlpcSendWaitReceivePort+0x1da
        fffff980`04e79bb0 00000000`776f0aca nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @
fffff980`04e79c20)
        00000000`0153f6d8 000007fe`fdd460d2 ntdll!NtAlpcSendWaitReceivePort+0xa
        00000000`0153f6e0 00000000`776cb332 CSRSRV!CsrApiRequestThread+0x146
        00000000`0153f9e0 00000000`00000000 ntdll!RtlUserThreadStart+0x29
```

```
        THREAD fffffa8002445bb0  Cid 01a4.01c0  Teb: 000007fffffd4000 Win32Thread:
0000000000000000 WAIT: (WrLpcReceive) UserMode Non-Alertable
            fffffa8002445f40  Semaphore Limit 0x1
        Not impersonating
        DeviceMap                 fffff88000007790
        Owning Process            fffffa80023cba80       Image:         csrss.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      3088           Ticks: 39370 (0:00:10:15.156)
        Context Switch Count      3              IdealProcessor: 1
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address CSRSRV!CsrSbApiRequestThread (0x000007fefdd4525c)
        Stack Init fffff980049f8db0 Current fffff980049f87f0
        Base fffff980049f9000 Limit fffff980049f3000 Call 0
        Priority 14 BasePriority 13 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Kernel stack not resident.
        Child-SP          RetAddr           Call Site
        fffff980`049f8830 fffff800`0185d695 nt!KiSwapContext+0x84
        fffff980`049f8970 fffff800`0185d3dd nt!KiSwapThread+0x125
        fffff980`049f89d0 fffff800`01ab3a98 nt!KeWaitForSingleObject+0x5f5
        fffff980`049f8a50 fffff800`01a80743 nt!AlpcpReceiveMessagePort+0x298
        fffff980`049f8ab0 fffff800`01a75722 nt!AlpcpReceiveLegacyMessage+0x122
        fffff980`049f8b50 fffff800`01a73b0f nt!NtReplyWaitReceivePortEx+0xc1
        fffff980`049f8be0 fffff800`0184dcf3 nt!NtReplyWaitReceivePort+0xf
        fffff980`049f8c20 00000000`776f032a nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @
fffff980`049f8c20)
        00000000`00a5fd08 000007fe`fdd452a9 ntdll!NtReplyWaitReceivePort+0xa
        00000000`00a5fd10 00000000`776cb332 CSRSRV!CsrSbApiRequestThread+0x4d
        00000000`00a5fe90 00000000`00000000 ntdll!RtlUserThreadStart+0x29

        THREAD fffffa80023ef650  Cid 01a4.01d0  Teb: 000007fffffde000 Win32Thread:
fffff900c06b6580 WAIT: (WrLpcReceive) UserMode Non-Alertable
            fffffa80023ef9e0  Semaphore Limit 0x1
        Not impersonating
        DeviceMap                 fffff88000007790
        Owning Process            fffffa80023cba80       Image:         csrss.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      42400          Ticks: 58 (0:00:00:00.906)
        Context Switch Count      1153           IdealProcessor: 0                 LargeStack
        UserTime                  00:00:00.093
        KernelTime                00:00:00.187
        Win32 Start Address CSRSRV!CsrApiRequestThread (0x000007fefdd45f8c)
        Stack Init fffff98007efadb0 Current fffff98007efa7a0
        Base fffff98007efb000 Limit fffff98007ef3000 Call 0
        Priority 15 BasePriority 13 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Child-SP          RetAddr           Call Site
        fffff980`07efa7e0 fffff800`0185d695 nt!KiSwapContext+0x84
        fffff980`07efa920 fffff800`0185d3dd nt!KiSwapThread+0x125
        fffff980`07efa980 fffff800`01ab3a98 nt!KeWaitForSingleObject+0x5f5
        fffff980`07efaa00 fffff800`01ab41ea nt!AlpcpReceiveMessagePort+0x298
        fffff980`07efaa60 fffff800`01ab74ea nt!AlpcpReceiveMessage+0x246
        fffff980`07efab00 fffff800`0184dcf3 nt!NtAlpcSendWaitReceivePort+0x1da
        fffff980`07efabb0 00000000`776f0aca nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @
fffff980`07efac20)
        00000000`0157f758 000007fe`fdd460d2 ntdll!NtAlpcSendWaitReceivePort+0xa
        00000000`0157f760 00000000`776cb332 CSRSRV!CsrApiRequestThread+0x146
        00000000`0157fa60 00000000`00000000 ntdll!RtlUserThreadStart+0x29
```

```
        THREAD fffffa800246d420  Cid 01a4.0204  Teb: 000007fffffae000 Win32Thread:
fffff900c00df570 WAIT: (WrUserRequest) KernelMode Alertable
            fffffa800248d260  SynchronizationEvent
            fffffa800246b290  NotificationTimer
            fffffa800246b240  SynchronizationTimer
            fffff8000197a920  NotificationEvent
        Not impersonating
        DeviceMap                 fffff88000007790
        Owning Process            fffffa80023cba80       Image:         csrss.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      42264          Ticks: 194 (0:00:00:03.031)
        Context Switch Count      136            IdealProcessor: 1                 LargeStack
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address winsrv!StartCreateSystemThreads (0x000007fefdcbc640)
        Stack Init fffff98007e88db0 Current fffff98007e888b0
        Base fffff98007e89000 Limit fffff98007e83000 Call 0
        Priority 15 BasePriority 13 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Child-SP          RetAddr           Call Site
        fffff980`07e888f0 fffff800`0185d695 nt!KiSwapContext+0x84
        fffff980`07e88a30 fffff800`0185ad2f nt!KiSwapThread+0x125
        fffff980`07e88a90 fffff960`000c17f1 nt!KeWaitForMultipleObjects+0x703
        fffff980`07e88b00 fffff960`00056838 win32k!RawInputThread+0x681
        fffff980`07e88bc0 fffff960`000d1c80 win32k!xxxCreateSystemThreads+0x58
        fffff980`07e88bf0 fffff800`0184dcf3 win32k!NtUserCallNoParam+0x20
        fffff980`07e88c20 000007fe`fdcbe01a nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @
fffff980`07e88c20)
        00000000`014afe18 000007fe`fdcbc659 winsrv!NtUserCallNoParam+0xa
        00000000`014afe20 00000000`776cb332 winsrv!StartCreateSystemThreads+0x19
        00000000`014afe50 00000000`00000000 ntdll!RtlUserThreadStart+0x29

        THREAD fffffa800245f780  Cid 01a4.0208  Teb: 000007fffffac000 Win32Thread:
fffff900c00df2c0 WAIT: (WrUserRequest) UserMode Non-Alertable
            fffffa8002427910  SynchronizationEvent
            fffffa800245f470  SynchronizationEvent
        Not impersonating
        DeviceMap                 fffff88000007790
        Owning Process            fffffa80023cba80       Image:         csrss.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      38691          Ticks: 3767 (0:00:00:58.859)
        Context Switch Count      16             IdealProcessor: 0                 LargeStack
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address winsrv!StartCreateSystemThreads (0x000007fefdcbc640)
        Stack Init fffff98007e9bdb0 Current fffff98007e9b880
        Base fffff98007e9c000 Limit fffff98007e96000 Call 0
        Priority 15 BasePriority 13 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Kernel stack not resident.
        Child-SP          RetAddr           Call Site
        fffff980`07e9b8c0 fffff800`0185d695 nt!KiSwapContext+0x84
        fffff980`07e9ba00 fffff800`0185ad2f nt!KiSwapThread+0x125
        fffff980`07e9ba60 fffff960`00099673 nt!KeWaitForMultipleObjects+0x703
        fffff980`07e9bad0 fffff960`0009a4e1 win32k!xxxMsgWaitForMultipleObjects+0xf3
        fffff980`07e9bb50 fffff960`00056844 win32k!xxxDesktopThread+0x212
        fffff980`07e9bbc0 fffff960`000d1c80 win32k!xxxCreateSystemThreads+0x64
        fffff980`07e9bbf0 fffff800`0184dcf3 win32k!NtUserCallNoParam+0x20
        fffff980`07e9bc20 000007fe`fdcbe01a nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @
fffff980`07e9bc20)
        00000000`0018f958 000007fe`fdcbc659 winsrv!NtUserCallNoParam+0xa
        00000000`0018f960 00000000`776cb332 winsrv!StartCreateSystemThreads+0x19
```

```
        00000000`0018f990 00000000`00000000 ntdll!RtlUserThreadStart+0x29

        THREAD fffffa80024d1bb0  Cid 01a4.0240  Teb: 000007fffffaa000 Win32Thread:
fffff900c06b81b0 WAIT: (WrLpcReceive) UserMode Non-Alertable
            fffffa80024d1f40  Semaphore Limit 0x1
        Not impersonating
        DeviceMap                 fffff88000007790
        Owning Process            fffffa80023cba80       Image:         csrss.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      42403          Ticks: 55 (0:00:00:00.859)
        Context Switch Count      1132           IdealProcessor: 1                 LargeStack
        UserTime                  00:00:00.109
        KernelTime                00:00:00.203
        Win32 Start Address CSRSRV!CsrApiRequestThread (0x000007fefdd45f8c)
        Stack Init fffff98007f20db0 Current fffff98007f207a0
        Base fffff98007f21000 Limit fffff98007f1a000 Call 0
        Priority 15 BasePriority 13 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Child-SP          RetAddr           Call Site
        fffff980`07f207e0 fffff800`0185d695 nt!KiSwapContext+0x84
        fffff980`07f20920 fffff800`0185d3dd nt!KiSwapThread+0x125
        fffff980`07f20980 fffff800`01ab3a98 nt!KeWaitForSingleObject+0x5f5
        fffff980`07f20a00 fffff800`01ab41ea nt!AlpcpReceiveMessagePort+0x298
        fffff980`07f20a60 fffff800`01ab74ea nt!AlpcpReceiveMessage+0x246
        fffff980`07f20b00 fffff800`0184dcf3 nt!NtAlpcSendWaitReceivePort+0x1da
        fffff980`07f20bb0 00000000`776f0aca nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @
fffff980`07f20c20)
        00000000`05d8fb78 000007fe`fdd460d2 ntdll!NtAlpcSendWaitReceivePort+0xa
        00000000`05d8fb80 00000000`776cb332 CSRSRV!CsrApiRequestThread+0x146
        00000000`05d8fe80 00000000`00000000 ntdll!RtlUserThreadStart+0x29

        THREAD fffffa80024d5bb0  Cid 01a4.0244  Teb: 000007fffffa8000 Win32Thread:
fffff900c06b6160 WAIT: (WrUserRequest) UserMode Non-Alertable
            fffffa80024cf6e0  SynchronizationEvent
        Not impersonating
        DeviceMap                 fffff88000007790
        Owning Process            fffffa80023cba80       Image:         csrss.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      11494          Ticks: 30964 (0:00:08:03.812)
        Context Switch Count      5              IdealProcessor: 0                 LargeStack
        UserTime                  00:00:00.000
        KernelTime                00:00:00.015
        Win32 Start Address winsrv!StartCreateSystemThreads (0x000007fefdcbc640)
        Stack Init fffff98007f0ddb0 Current fffff98007f0d880
        Base fffff98007f0e000 Limit fffff98007f08000 Call 0
        Priority 15 BasePriority 13 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Kernel stack not resident.
        Child-SP          RetAddr           Call Site
        fffff980`07f0d8c0 fffff800`0185d695 nt!KiSwapContext+0x84
        fffff980`07f0da00 fffff800`0185ad2f nt!KiSwapThread+0x125
        fffff980`07f0da60 fffff960`00099673 nt!KeWaitForMultipleObjects+0x703
        fffff980`07f0dad0 fffff960`0009a4e1 win32k!xxxMsgWaitForMultipleObjects+0xf3
        fffff980`07f0db50 fffff960`00056844 win32k!xxxDesktopThread+0x212
        fffff980`07f0dbc0 fffff960`000d1c80 win32k!xxxCreateSystemThreads+0x64
        fffff980`07f0dbf0 fffff800`0184dcf3 win32k!NtUserCallNoParam+0x20
        fffff980`07f0dc20 000007fe`fdcbe01a nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @
fffff980`07f0dc20)
        00000000`05d3fea8 000007fe`fdcbc659 winsrv!NtUserCallNoParam+0xa
        00000000`05d3feb0 00000000`776cb332 winsrv!StartCreateSystemThreads+0x19
        00000000`05d3fee0 00000000`00000000 ntdll!RtlUserThreadStart+0x29
```

```
        THREAD fffffa80024e5bb0  Cid 01a4.0258  Teb: 000007fffffa6000 Win32Thread:
fffff900c06ba5f0 WAIT: (WrUserRequest) UserMode Non-Alertable
            fffffa80024e36b0  SynchronizationEvent
        Not impersonating
        DeviceMap                 fffff88000007790
        Owning Process            fffffa80023cba80       Image:         csrss.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      3291           Ticks: 39167 (0:00:10:11.984)
        Context Switch Count      6              IdealProcessor: 1                 LargeStack
        UserTime                  00:00:00.000
        KernelTime                00:00:00.015
        Win32 Start Address winsrv!ConsoleInputThread (0x000007fefdcb3450)
        Stack Init fffff98007f33db0 Current fffff98007f33740
        Base fffff98007f34000 Limit fffff98007f2c000 Call 0
        Priority 15 BasePriority 13 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Kernel stack not resident.
        Child-SP          RetAddr           Call Site
        fffff980`07f33780 fffff800`0185d695 nt!KiSwapContext+0x84
        fffff980`07f338c0 fffff800`0185d3dd nt!KiSwapThread+0x125
        fffff980`07f33920 fffff960`000c95f8 nt!KeWaitForSingleObject+0x5f5
        fffff980`07f339a0 fffff960`000c9686 win32k!xxxRealSleepThread+0x278
        fffff980`07f33a40 fffff960`000c7dde win32k!xxxSleepThread+0x56
        fffff980`07f33a70 fffff960`000c7ee5 win32k!xxxRealInternalGetMessage+0x72e
        fffff980`07f33b50 fffff960`000c97a4 win32k!xxxInternalGetMessage+0x35
        fffff980`07f33b90 fffff800`0184dcf3 win32k!NtUserGetMessage+0x64
        fffff980`07f33c20 00000000`775ee6ba nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @
fffff980`07f33c20)
        00000000`05ddfc88 00000000`775ee6fa USER32!ZwUserGetMessage+0xa
        00000000`05ddfc90 000007fe`fdcb3556 USER32!GetMessageW+0x34
        00000000`05ddfcc0 00000000`776cb332 winsrv!ConsoleInputThread+0x315
        00000000`05ddfda0 00000000`00000000 ntdll!RtlUserThreadStart+0x29

        THREAD fffffa8003b86b40  Cid 01a4.077c  Teb: 000007fffffa4000 Win32Thread:
fffff900c07e1010 WAIT: (WrUserRequest) UserMode Non-Alertable
            fffffa8003b44fe0  SynchronizationEvent
        Not impersonating
        DeviceMap                 fffff88000007790
        Owning Process            fffffa80023cba80       Image:         csrss.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      10169          Ticks: 32289 (0:00:08:24.515)
        Context Switch Count      10             IdealProcessor: 0                 LargeStack
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address winsrv!ConsoleInputThread (0x000007fefdcb3450)
        Stack Init fffff9800e06edb0 Current fffff9800e06e740
        Base fffff9800e06f000 Limit fffff9800e067000 Call 0
        Priority 15 BasePriority 13 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Kernel stack not resident.
        Child-SP          RetAddr           Call Site
        fffff980`0e06e780 fffff800`0185d695 nt!KiSwapContext+0x84
        fffff980`0e06e8c0 fffff800`0185d3dd nt!KiSwapThread+0x125
        fffff980`0e06e920 fffff960`000c95f8 nt!KeWaitForSingleObject+0x5f5
        fffff980`0e06e9a0 fffff960`000c9686 win32k!xxxRealSleepThread+0x278
        fffff980`0e06ea40 fffff960`000c7dde win32k!xxxSleepThread+0x56
        fffff980`0e06ea70 fffff960`000c7ee5 win32k!xxxRealInternalGetMessage+0x72e
        fffff980`0e06eb50 fffff960`000c97a4 win32k!xxxInternalGetMessage+0x35
        fffff980`0e06eb90 fffff800`0184dcf3 win32k!NtUserGetMessage+0x64
        fffff980`0e06ec20 00000000`775ee6ba nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @
fffff980`0e06ec20)
        00000000`066cfb88 00000000`775ee6fa USER32!ZwUserGetMessage+0xa
```

```
00000000`066cfb90 000007fe`fdcb3556 USER32!GetMessageW+0x34
00000000`066cfbc0 00000000`776cb332 winsrv!ConsoleInputThread+0x315
00000000`066cfca0 00000000`00000000 ntdll!RtlUserThreadStart+0x29
```

Note: We see that the current process has changed. We also notice passive threads that lead to a weekly coupled process. We also specified **3f** flags to have the process context changed to that of csrss.exe during the execution of **!process** command.

8. Now we list processes and threads from the session 1:

```
1: kd> !sprocess 1 3f
Dumping Session 1

_MM_SESSION_SPACE fffff98001481000
_MMSESSION        fffff98001481b40
PROCESS fffffa8002453c10
    SessionId: 1  Cid: 01dc    Peb: 7fffffde000  ParentCid: 01d4
    DirBase: 24c71000  ObjectTable: fffff8800159b610  HandleCount: 356.
    Image: csrss.exe
    VadRoot fffffa8002465800 Vads 96 Clone 0 Private 446. Modified 3194. Locked 0.
    DeviceMap fffff88000007790
    Token                             fffff8800159cc00
    ElapsedTime                       00:10:15.403
    UserTime                          00:00:00.000
    KernelTime                        00:00:01.656
    QuotaPoolUsage[PagedPool]         248152
    QuotaPoolUsage[NonPagedPool]      12480
    Working Set Sizes (now,min,max)  (1489, 50, 345) (5956KB, 200KB, 1380KB)
    PeakWorkingSetSize                2119
    VirtualSize                       114 Mb
    PeakVirtualSize                   165 Mb
    PageFaultCount                    11534
    MemoryPriority                    BACKGROUND
    BasePriority                      13
    CommitCharge                      1598

    PEB at 000007fffffde000
    InheritedAddressSpace:    No
    ReadImageFileExecOptions: No
    BeingDebugged:            No
    ImageBaseAddress:         0000000049740000
    Ldr                       00000000777af980
    Ldr.Initialized:          Yes
    Ldr.InInitializationOrderModuleList: 00000000001320d0 . 000000000015acb0
    Ldr.InLoadOrderModuleList:           0000000000131fe0 . 000000000015ac90
    Ldr.InMemoryOrderModuleList:         0000000000131ff0 . 000000000015aca0
            Base TimeStamp                     Module
        49740000 4549b4cc Nov 02 09:05:16 2006 C:\Windows\system32\csrss.exe
        776a0000 4549d372 Nov 02 11:16:02 2006 C:\Windows\system32\ntdll.dll
     7fefdd40000 4549d353 Nov 02 11:15:31 2006 C:\Windows\system32\CSRSRV.dll
     7fefdd20000 4549d24d Nov 02 11:11:09 2006 C:\Windows\system32\basesrv.dll
     7fefdcb0000 4549d37e Nov 02 11:16:14 2006 C:\Windows\system32\winsrv.dll
        775d0000 4549d334 Nov 02 11:15:00 2006 C:\Windows\system32\USER32.dll
        77490000 4549d328 Nov 02 11:14:48 2006 C:\Windows\system32\KERNEL32.dll
     7fefe8f0000 4549d273 Nov 02 11:11:47 2006 C:\Windows\system32\GDI32.dll
     7fefe260000 4549d267 Nov 02 11:11:35 2006 C:\Windows\system32\ADVAPI32.dll
     7fefeb90000 4549d31a Nov 02 11:14:34 2006 C:\Windows\system32\RPCRT4.dll
```

```
     7fefeb80000 4549d2a1 Nov 02 11:12:33 2006 C:\Windows\system32\LPK.DLL
     7fefe380000 4549d337 Nov 02 11:15:03 2006 C:\Windows\system32\USP10.dll
     7fefead0000 4549d2e1 Nov 02 11:13:37 2006 C:\Windows\system32\msvcrt.dll
     7fefda80000 4549d32f Nov 02 11:14:55 2006 C:\Windows\system32\sxs.dll
    SubSystemData:     0000000000000000
    ProcessHeap:       0000000000130000
    ProcessParameters: 00000000001316d0
    CurrentDirectory:  'C:\Windows\system32\'
    WindowTitle:  '< Name not readable >'
    ImageFile:    'C:\Windows\system32\csrss.exe'
    CommandLine:  'C:\Windows\system32\csrss.exe ObjectDirectory=\Windows
SharedSection=1024,20480,768 Windows=On SubSystemType=Windows ServerDll=basesrv,1
ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=winsrv:ConServerDllInitialization,2
ProfileControl=Off MaxRequestThreads=16'
    DllPath:      'C:\Windows\system32;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem'
    Environment:  0000000000131310
        ComSpec=C:\Windows\system32\cmd.exe
        FP_NO_HOST_CHECK=NO
        NUMBER_OF_PROCESSORS=2
        OS=Windows_NT
        Path=C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem
        PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
        PROCESSOR_ARCHITECTURE=AMD64
        PROCESSOR_IDENTIFIER=EM64T Family 6 Model 15 Stepping 11, GenuineIntel
        PROCESSOR_LEVEL=6
        PROCESSOR_REVISION=0f0b
        SystemDrive=C:
        SystemRoot=C:\Windows
        TEMP=C:\Windows\TEMP
        TMP=C:\Windows\TEMP
        USERNAME=SYSTEM
        windir=C:\Windows


        THREAD fffffa80024714f0  Cid 01dc.01f0  Teb: 0000000000000000 Win32Thread:
0000000000000000 WAIT: (Executive) KernelMode Non-Alertable
            fffffa8002413d70  SynchronizationEvent
            fffffa80024715a8  NotificationTimer
        Not impersonating
        DeviceMap                 fffff88000007790
        Owning Process            fffffa8002453c10       Image:         csrss.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      42458          Ticks: 0
        Context Switch Count      28049          IdealProcessor: 1
        UserTime                  00:00:00.000
        KernelTime                00:00:02.703
        Win32 Start Address cdd!PresentWorkerThread (0xfffff96000603c38)
        Stack Init fffff980049cedb0 Current fffff980049ce9e0
        Base fffff980049cf000 Limit fffff980049c9000 Call 0
        Priority 15 BasePriority 15 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Child-SP          RetAddr           Call Site
        fffff980`049cea20 fffff800`0185d695 nt!KiSwapContext+0x84
        fffff980`049ceb60 fffff800`0185d3dd nt!KiSwapThread+0x125
        fffff980`049cebc0 fffff960`006040ae nt!KeWaitForSingleObject+0x5f5
        fffff980`049cec40 fffff800`01ae199b cdd!PresentWorkerThread+0x476
        fffff980`049ced50 fffff800`01834b86 nt!PspSystemThreadStartup+0x5b
        fffff980`049ced80 00000000`00000000 nt!KiStartSystemThread+0x16
```

```
        THREAD fffffa800247b4f0  Cid 01dc.01f4  Teb: 000007fffffda000 Win32Thread:
fffff900c06ba910 WAIT: (WrLpcReply) UserMode Non-Alertable
            fffffa800247b880  Semaphore Limit 0x1
        Waiting for reply to ALPC Message fffff88001eaa030
        Not impersonating
        DeviceMap                 fffff88000007790
        Owning Process            fffffa8002453c10       Image:         csrss.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      12306          Ticks: 30152 (0:00:07:51.125)
        Context Switch Count      714            IdealProcessor: 0                 LargeStack
        UserTime                  00:00:00.000
        KernelTime                00:00:00.031
        Win32 Start Address winsrv!TerminalServerRequestThread (0x000007fefdcb9ac0)
        Stack Init fffff9800cce4db0 Current fffff9800cce4760
        Base fffff9800cce5000 Limit fffff9800ccdf000 Call 0
        Priority 15 BasePriority 15 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Kernel stack not resident.
        Child-SP          RetAddr           Call Site
        fffff980`0cce47a0 fffff800`0185d695 nt!KiSwapContext+0x84
        fffff980`0cce48e0 fffff800`0185d3dd nt!KiSwapThread+0x125
        fffff980`0cce4940 fffff800`0188fd77 nt!KeWaitForSingleObject+0x5f5
        fffff980`0cce49c0 fffff800`01abed94 nt!AlpcpSignalAndWait+0x97
        fffff980`0cce4a00 fffff800`01a9ab3a nt!AlpcpReceiveSynchronousReply+0x44
        fffff980`0cce4a60 fffff800`01aa6ac2 nt!AlpcpProcessSynchronousRequest+0x257
        fffff980`0cce4b80 fffff800`01a9f1cd nt!LpcpRequestWaitReplyPort+0x91
        fffff980`0cce4be0 fffff800`0184dcf3 nt!NtRequestWaitReplyPort+0x6d
        fffff980`0cce4c20 00000000`776f049a nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @
fffff980`0cce4c20)
        00000000`0083fb18 000007fe`fdcb9c30 ntdll!NtRequestWaitReplyPort+0xa
        00000000`0083fb20 00000000`776cb332 winsrv!TerminalServerRequestThread+0x256
        00000000`0083fd30 00000000`00000000 ntdll!RtlUserThreadStart+0x29

[...]

PROCESS fffffa80040d2040
    SessionId: 1  Cid: 05b0    Peb: 7efdf000  ParentCid: 05f0
    DirBase: 03d42000  ObjectTable: fffff88002740010  HandleCount: 336.
    Image: iexplore.exe
    VadRoot fffffa80013f4970 Vads 275 Clone 0 Private 1536. Modified 56. Locked 0.
    DeviceMap fffff88001697690
    Token                             fffff88002dd0a70
    ElapsedTime                       00:00:28.052
    UserTime                          00:00:00.156
    KernelTime                        00:00:00.265
    QuotaPoolUsage[PagedPool]         205248
    QuotaPoolUsage[NonPagedPool]      28032
    Working Set Sizes (now,min,max)  (5635, 50, 345) (22540KB, 200KB, 1380KB)
    PeakWorkingSetSize                5694
    VirtualSize                       113 Mb
    PeakVirtualSize                   128 Mb
    PageFaultCount                    6778
    MemoryPriority                    BACKGROUND
    BasePriority                      8
    CommitCharge                      3462

    PEB at 000000007efdf000
    InheritedAddressSpace:    No
    ReadImageFileExecOptions: No
    BeingDebugged:            No
    ImageBaseAddress:         0000000000e00000
```

```
    Ldr                       00000000777af980
    Ldr.Initialized:          Yes
    Ldr.InInitializationOrderModuleList: 00000000000826e0 . 0000000000082a40
    Ldr.InLoadOrderModuleList:           00000000000825f0 . 0000000000082ba0
    Ldr.InMemoryOrderModuleList:         0000000000082600 . 0000000000082bb0
            Base TimeStamp                     Module
          e00000 4549b133 Nov 02 08:49:55 2006 C:\Program Files (x86)\Internet
Explorer\iexplore.exe
        776a0000 4549d372 Nov 02 11:16:02 2006 C:\Windows\system32\ntdll.dll
        74fb0000 4549d371 Nov 02 11:16:01 2006 C:\Windows\system32\wow64.dll
        75580000 4549d374 Nov 02 11:16:04 2006 C:\Windows\system32\wow64win.dll
        759e0000 4549d372 Nov 02 11:16:02 2006 C:\Windows\system32\wow64cpu.dll
    SubSystemData:     0000000000000000
    ProcessHeap:       0000000000080000
    ProcessParameters: 0000000000081d00
    CurrentDirectory:  'C:\Windows\system32\'
    WindowTitle:  'C:\Program Files (x86)\Internet Explorer\iexplore.exe'
    ImageFile:    'C:\Program Files (x86)\Internet Explorer\iexplore.exe'
    CommandLine:  '"C:\Program Files (x86)\Internet Explorer\iexplore.exe" '
    DllPath:      'C:\Program Files (x86)\Internet
Explorer;C:\Windows\system32;C:\Windows\system;C:\Windows;.;C:\Program Files\Internet
Explorer;;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem'
    Environment:  0000000000081310
        =::=::\
        ALLUSERSPROFILE=C:\ProgramData
        APPDATA=C:\Users\Training\AppData\Roaming
        CommonProgramFiles=C:\Program Files\Common Files
        CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
        COMPUTERNAME=LH-UZIYU07F029O
        ComSpec=C:\Windows\system32\cmd.exe
        FP_NO_HOST_CHECK=NO
        HKCU_S=\REGISTRY\CUSER\Software
        HKLM_S=\REGISTRY\MACHINE\Software
        HOMEDRIVE=C:
        HOMEPATH=\Users\Training
        LOCALAPPDATA=C:\Users\Training\AppData\Local
        LOGONSERVER=\\LH-UZIYU07F029O
        NUMBER_OF_PROCESSORS=2
        OS=Windows_NT
        Path=C:\Program Files\Internet
Explorer;;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem
        PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
        PROCESSOR_ARCHITECTURE=AMD64
        PROCESSOR_IDENTIFIER=EM64T Family 6 Model 15 Stepping 11, GenuineIntel
        PROCESSOR_LEVEL=6
        PROCESSOR_REVISION=0f0b
        ProgramData=C:\ProgramData
        ProgramFiles=C:\Program Files
        ProgramFiles(x86)=C:\Program Files (x86)
        PUBLIC=C:\Users\Public
        SESSIONNAME=Console
        SystemDrive=C:
        SystemRoot=C:\Windows
        TEMP=C:\Users\Training\AppData\Local\Temp
        TMP=C:\Users\Training\AppData\Local\Temp
        USERDOMAIN=LH-UZIYU07F029O
        USERNAME=Training
        USERPROFILE=C:\Users\Training
        windir=C:\Windows
```

```
        THREAD fffffa800141dbb0  Cid 05b0.0890  Teb: 000000007efdb000 Win32Thread:
fffff900c228d1d0 WAIT: (UserRequest) UserMode Non-Alertable
            fffffa80040d2490  SynchronizationEvent
            fffffa800121c180  SynchronizationEvent
        Not impersonating
        DeviceMap                 fffff88001697690
        Owning Process            fffffa80040d2040       Image:         iexplore.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      41157          Ticks: 1301 (0:00:00:20.328)
        Context Switch Count      2106           IdealProcessor: 1                 LargeStack
        UserTime                  00:00:00.359
        KernelTime                00:00:01.375
        Win32 Start Address iexplore!wWinMainCRTStartup (0x0000000000e02d61)
        Stack Init fffff9800db54db0 Current fffff9800db54260
        Base fffff9800db55000 Limit fffff9800db4a000 Call 0
        Priority 10 BasePriority 8 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Child-SP          RetAddr           Call Site
        fffff980`0db542a0 fffff800`0185d695 nt!KiSwapContext+0x84
        fffff980`0db543e0 fffff800`0185ad2f nt!KiSwapThread+0x125
        fffff980`0db54440 fffff800`01ac1813 nt!KeWaitForMultipleObjects+0x703
        fffff980`0db544b0 fffff800`01ba477a nt!ObpWaitForMultipleObjects+0x216
        fffff980`0db54960 fffff800`0184dcf3 nt!NtWaitForMultipleObjects32+0xd9
        fffff980`0db54bb0 00000000`759e373f nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @
fffff980`0db54c20)
        00000000`0022ea18 00000000`74fbabfe wow64cpu!WaitForMultipleObjects32+0x3a
        00000000`0022eac0 00000000`74fba202 wow64!RunCpuSimulation+0xa
        00000000`0022eaf0 00000000`776de23d wow64!Wow64LdrpInitialize+0x492
        00000000`0022f050 00000000`7774e974 ntdll!LdrpInitializeProcess+0x1333
        00000000`0022f2e0 00000000`776ec4ee ntdll! ?? ::FNODOBFM::`string'+0x1d641
        00000000`0022f380 00000000`00000000 ntdll!LdrInitializeThunk+0xe

[...]
```

Note: We see that iexplorer.exe is a virtualized WOW64 process.

9. Suppose, we are interested in the first iexplore.exe thread (here we need **/w** switch):

```
1: kd> .load wow64exts

1: kd> .process /r /p fffffa80040d2040
Implicit process is now fffffa80`040d2040
Loading User Symbols
.....

************* Symbol Loading Error Summary **************
Module name            Error
myfault                The system cannot find the file specified

You can troubleshoot most symbol related issues by turning on symbol loading diagnostics (!sym
noisy) and repeating the command that caused symbols to be loaded.
You should also verify that your symbol search path (.sympath) is correct.

1: kd> .thread /w fffffa800141dbb0
Implicit thread is now fffffa80`0141dbb0
The context is partially valid. Only x86 user-mode context is available.
x86 context set
```

```
1: kd:x86> k
  *** Stack trace for last set context - .thread/.cxr resets it
ChildEBP          RetAddr
WARNING: Frame IP not in any known module. Following frames may be wrong.
0039e628 770e944c 0x7789aec5
0039e67c 72f8605c 0x770e944c
0039e69c 72f8634e 0x72f8605c
0039e6c4 72f86178 0x72f8634e
0039e700 730798d8 0x72f86178
0039e70c 7307a977 0x730798d8
0039e730 7307a8c5 0x7307a977
0039f7a0 7307a773 0x7307a8c5
0039f9d8 00e014bf 0x7307a773
0039fe18 00e0131a iexplore!wWinMain+0x28a
0039feac 76b619f1 iexplore!_initterm_e+0x1b1
0039feb8 778cd109 0x76b619f1
0039fef8 00000000 0x778cd109

1: kd:x86> .reload
Loading Kernel Symbols
...............................................................
...............................................................
.........
Loading User Symbols
.....
Loading unloaded module list
.........Unable to enumerate user-mode unloaded modules, NTSTATUS 0xC0000147
Loading Wow64 Symbols
...............................................................
.......................

1: kd:x86> k
  *** Stack trace for last set context - .thread/.cxr resets it
ChildEBP          RetAddr
0039e58c 76afedb5 ntdll_77850000!ZwWaitForMultipleObjects+0x15
0039e628 770e944c kernel32!WaitForMultipleObjectsEx+0x11d
0039e67c 72f8605c USER32!RealMsgWaitForMultipleObjectsEx+0x14d
0039e69c 72f8634e IEUI!CoreSC::Wait+0x49
0039e6c4 72f86178 IEUI!CoreSC::WaitMessage+0x54
0039e6d0 7308867d IEUI!WaitMessageEx+0x33
0039e700 730798d8 IEFRAME!CBrowserFrame::FrameMessagePump+0x199
0039e70c 7307a977 IEFRAME!BrowserThreadProc+0x3f
0039e730 7307a8c5 IEFRAME!BrowserNewThreadProc+0x7b
0039f7a0 7307a773 IEFRAME!SHOpenFolderWindow+0x188
0039f9d8 00e014bf IEFRAME!IEWinMain+0x336
0039fe18 00e0131a iexplore!wWinMain+0x28a
0039feac 76b619f1 iexplore!_initterm_e+0x1b1
0039feb8 778cd109 kernel32!BaseThreadInitThunk+0xe
0039fef8 00000000 ntdll_77850000!_RtlUserThreadStart+0x23
```

Note: To switch back to our native processor architecture we use **.effmach** or **!sw** commands:

```
1: kd:x86> .effmach AMD64
Effective machine: x64 (AMD64)

1: kd> !sw
Switched to 32bit mode
```

```
1: kd:x86> !sw
Switched to 64bit mode
```

10. Another way to list all stack traces is to use **!for_each_thread** command where we can customize stack trace output:

```
1: kd> !for_each_thread ".thread /r /p @#Thread; kv"
Implicit thread is now fffffa80`00c5bb20
Implicit process is now fffffa80`00c360b0
Loading User Symbols

  *** Stack trace for last set context - .thread/.cxr resets it
Child-SP          RetAddr           : Args to Child                                                   : Call Site
fffff980`00a0ca70 fffff800`0185d695 : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : nt!KiSwapContext+0x84
fffff980`00a0cbb0 fffff800`0185d3dd : fffff980`00000001 fffff800`0188e5bb ffffffff`ffffffff fffff980`00db8180 : nt!KiSwapThread+0x125
fffff980`00a0cc10 fffff800`0188814d : 00000000`00000000 00000000`00000008 00000000`00000000 fffffa80`00c5bb00 : nt!KeWaitForSingleObject+0x5f5
fffff980`00a0cc90 fffff800`01bfe99e : fffff800`00000000 fffffa80`00000040 00000000`0000003f fffffa80`00c5bb20 : nt!MmZeroPageThread+0x180
fffff980`00a0cd20 fffff800`01ae199b : 01cc4002`3f5f377e fffff800`01834b79 00000000`00000010 00000000`00000286 : nt!Phase1Initialization+0xe
fffff980`00a0cd50 fffff800`01834b86 : fffff800`01949880 fffffa80`00c5bb20 fffff800`0194eb80 fffff800`0080e5f8 : nt!PspSystemThreadStartup+0x5b
fffff980`00a0cd80 00000000`00000000 : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : nt!KiStartSystemThread+0x16

[...]

Implicit thread is now fffffa80`01220060
Implicit process is now fffffa80`01412040
Loading User Symbols
...............

************* Symbol Loading Error Summary **************
Module name            Error
pci                    The system cannot find the file specified
vmci                   The system cannot find the file specified
spsys                  The system cannot find the file specified
vmmemctl               The system cannot find the file specified

You can troubleshoot most symbol related issues by turning on symbol loading diagnostics (!sym noisy) and repeating the command that caused symbols to be
loaded.
You should also verify that your symbol search path (.sympath) is correct.
  *** Stack trace for last set context - .thread/.cxr resets it
Child-SP          RetAddr           : Args to Child                                                   : Call Site
fffff980`11da0568 fffff800`0184dff3 : 00000000`0000000a fffff880`02c04800 00000000`00000002 00000000`00000000 : nt!KeBugCheckEx
fffff980`11da0570 fffff800`0184cecb : 00000000`00000000 fffffa80`033f15a0 fffff980`11da0800 fffff980`11da07bc : nt!KiBugCheckDispatch+0x73
fffff980`11da06b0 fffff980`0e6e117a : 00000000`00000001 fffff980`11da0ca0 fffffa80`00c2b090 00000000`656e6f4e : nt!KiPageFault+0x20b (TrapFrame @
fffff980`11da06b0)
*** ERROR: Module load completed but symbols could not be loaded for myfault.sys
fffff980`11da0840 fffff980`0e6e1397 : fffffa80`033f1600 fffff880`02ef3230 fffffa80`01412040 fffff900`c30cd460 : myfault+0x117a
fffff980`11da09a0 fffff800`01a8e217 : fffffa80`033f1600 fffffa80`014c81d0 00000000`00000001 00000000`00000000 : myfault+0x1397
fffff980`11da0a00 fffff800`01a94466 : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : nt!IopXxxControlFile+0x626
fffff980`11da0b40 fffff800`0184dcf3 : 00000000`000a03cc fffff960`000d1a78 fffff900`c30cd460 fffff980`11da0ca0 : nt!NtDeviceIoControlFile+0x56
fffff980`11da0bb0 00000000`776f02ea : 00000000`774b21da 00000000`0020025c 00000000`775fdf00 00000000`00000000 : nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @
fffff980`11da0c20)
00000000`0012f5c8 00000000`774b21da : 00000000`0020025c 00000000`775fdf00 00000000`00000000 00000000`000000f0 : ntdll!ZwDeviceIoControlFile+0xa
*** ERROR: Module load completed but symbols could not be loaded for NotMyfault.exe
00000000`0012f5d0 00000001`40001cc9 : 00000000`83360018 00000000`00000004 00000000`00000000 00000000`00000013 : kernel32!DeviceIoControl+0xaa
00000000`0012f650 00000000`775ee26a : 00000000`00000070 ffffffff`ffffffff ffffffff`ffffffff 00000000`775ee1ef : NotMyfault+0x1cc9
00000000`0012f830 00000000`775eb7d9 : 00000000`00b34a50 00000001`400019a0 00000000`00b34a78 00000001`400019a0 : USER32!UserCallWinProcCheckWow+0x1ad
00000000`0012f8f0 00000000`775eb86d : 00000000`000c01da 00000000`0012f900 00000000`00000001 00000000`00000111 : USER32!SendMessageWorker+0x64a
00000000`0012f980 00000000`77611ad4 : 00000000`00afd420 00000000`000a03cc 00000000`00000000 00000000`77612820 : USER32!SendMessageW+0x5b
00000000`0012f9d0 00000000`7760080b : 00000000`00000000 00000000`00afd420 00000000`0012fab0 00000000`00000040 : USER32!xxxBNReleaseCapture+0x444
00000000`0012fa40 00000000`77612886 : 00000000`00000202 00000000`00000000 00000000`0178016d ffffffff`ffffffff : USER32!ButtonWndProcWorker+0x1490
00000000`0012fb90 00000000`775ee26a : 00000000`00000000 00000000`00000001 00000000`00000000 00000000`774c8a36 : USER32!ButtonWndProcA+0x66
00000000`0012fbd0 00000000`775eec9f : 00000000`0012fe70 00000000`77612820 00000000`00b34a50 00000000`00afd420 : USER32!UserCallWinProcCheckWow+0x1ad
00000000`0012fc90 00000000`775e0e63 : 00000000`00000000 00000000`00000000 00000000`77612820 00000000`00000001 : USER32!DispatchMessageWorker+0x389
00000000`0012fd10 00000000`776276ba : 00000000`000c01da 00000000`00000000 00000000`00000000 00000000`00000000 : USER32!IsDialogMessageW+0x14f
00000000`0012fda0 00000001`40001ed1 : 00000000`000c01da 00000000`00000001 00000001`40000000 00000000`0018f688 : USER32!IsDialogMessageA+0x7a
00000000`0012fdd0 00000001`400026b5 : 00000000`00000000 00000000`00000000 00000001`40000000 00000000`00000001 : NotMyfault+0x1ed1
00000000`0012feb0 00000000`774ccdcd : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : NotMyfault+0x26b5
00000000`0012ff60 00000000`776ec6e1 : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : kernel32!BaseThreadInitThunk+0xd
00000000`0012ff90 00000000`00000000 : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : ntdll!RtlUserThreadStart+0x1d
```

Note: We can use this script to list all processes and threads:

```
1: kd> !for_each_thread "!thread @#Thread 1f;.thread /w @#Thread; .reload; kb 256; .effmach AMD64"
THREAD fffffa8000c5bb20  Cid 0004.0008  Teb: 0000000000000000 Win32Thread: 0000000000000000 WAIT: (WrFreePage)
KernelMode Non-Alertable
    fffff8000199ad00  NotificationEvent
Not impersonating
DeviceMap                 fffff88000007790
Owning Process            fffffa8000c360b0       Image:         System
Attached Process          N/A            Image:         N/A
Wait Start TickCount      42438          Ticks: 20 (0:00:00:00.312)
Context Switch Count      8288           IdealProcessor: 0
UserTime                  00:00:00.000
KernelTime                00:00:15.578
Win32 Start Address nt!Phase1Initialization (0xfffff80001bfe990)
Stack Init fffff98000a0cdb0 Current fffff98000a0ca30
```

```
Base fffff98000a0d000 Limit fffff98000a07000 Call 0
Priority 0 BasePriority 0 PriorityDecrement 0 IoPriority 2 PagePriority 5
Child-SP          RetAddr           Call Site
fffff980`00a0ca70 fffff800`0185d695 nt!KiSwapContext+0x84
fffff980`00a0cbb0 fffff800`0185d3dd nt!KiSwapThread+0x125
fffff980`00a0cc10 fffff800`0188814d nt!KeWaitForSingleObject+0x5f5
fffff980`00a0cc90 fffff800`01bfe99e nt!MmZeroPageThread+0x180
fffff980`00a0cd20 fffff800`01ae199b nt!Phase1Initialization+0xe
fffff980`00a0cd50 fffff800`01834b86 nt!PspSystemThreadStartup+0x5b
fffff980`00a0cd80 00000000`00000000 nt!KiStartSystemThread+0x16

Implicit thread is now fffffa80`00c5bb20
WARNING: WOW context retrieval requires
switching to the thread's process context.
Use .process /p fffffa80`01412040 to switch back.
Implicit process is now fffffa80`00c360b0
The context is partially valid. Only x86 user-mode context is available.
x86 context set
Loading Kernel Symbols
...............................................................
...............................................................
Loading User Symbols

Loading unloaded module list
.........
ChildEBP          RetAddr           Args to Child
WARNING: Frame IP not in any known module. Following frames may be wrong.
00db8180 00000000 00000000 00000000 00000000 0x1850374
Effective machine: x64 (AMD64)

[...]

Effective machine: x64 (AMD64)
THREAD fffffa80012b7bb0  Cid 05b0.086c  Teb: 000000007ef98000 Win32Thread: fffff900c30c7ad0 WAIT: (UserRequest)
UserMode Non-Alertable
    fffffa80039c6dc0  SynchronizationEvent
    fffffa80012b7c68  NotificationTimer
Not impersonating
DeviceMap                 fffff88001697690
Owning Process            fffffa80040d2040       Image:         iexplore.exe
Attached Process          N/A            Image:         N/A
Wait Start TickCount      40954          Ticks: 1504 (0:00:00:23.500)
Context Switch Count      83             IdealProcessor: 1                 LargeStack
UserTime                  00:00:00.093
KernelTime                00:00:00.062
Win32 Start Address mshtml!CExecFT::StaticThreadProc (0x000000007230dacf)
Stack Init fffff98010367db0 Current fffff98010367960
Base fffff98010368000 Limit fffff98010360000 Call 0
Priority 8 BasePriority 8 PriorityDecrement 0 IoPriority 2 PagePriority 5
Child-SP          RetAddr           Call Site
fffff980`103679a0 fffff800`0185d695 nt!KiSwapContext+0x84
fffff980`10367ae0 fffff800`0185d3dd nt!KiSwapThread+0x125
fffff980`10367b40 fffff800`01a8b27b nt!KeWaitForSingleObject+0x5f5
fffff980`10367bc0 fffff800`0184dcf3 nt!NtWaitForSingleObject+0x9b
fffff980`10367c20 00000000`759e3cf9 nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff980`10367c20)
00000000`0374e9b8 00000000`759e3af6 wow64cpu!CpupSyscallStub+0x9
00000000`0374e9c0 00000000`74fbabfe wow64cpu!Thunk0ArgReloadState+0x1a
00000000`0374ea30 00000000`74fba202 wow64!RunCpuSimulation+0xa
00000000`0374ea60 00000000`776c894c wow64!Wow64LdrpInitialize+0x492
00000000`0374efc0 00000000`776ec4ee ntdll! ?? ::FNODOBFM::`string'+0x1d777
00000000`0374f060 00000000`00000000 ntdll!LdrInitializeThunk+0xe

Implicit thread is now fffffa80`012b7bb0
WARNING: WOW context retrieval requires
switching to the thread's process context.
Use .process /p fffffa80`01412040 to switch back.
Implicit process is now fffffa80`040d2040
x86 context set
Loading Kernel Symbols
...............................................................
...............................................................
.........
Loading User Symbols
.....
Loading unloaded module list
.........Unable to enumerate user-mode unloaded modules, NTSTATUS 0xC0000147
```

```
Loading Wow64 Symbols
...............................................................
.......................
ChildEBP          RetAddr           Args to Child
043bf9e0 76af1220 00000550 00000000 043bfa28 ntdll_77850000!ZwWaitForSingleObject+0x15
043bfa50 76af1188 00000550 000927c0 00000000 kernel32!WaitForSingleObjectEx+0xbe
043bfa64 7238f57d 00000550 000927c0 722f0000 kernel32!WaitForSingleObject+0x12
043bfa7c 72386c92 00000000 00000000 7230dadc mshtml!CDwnTaskExec::ThreadExec+0x127
043bfa88 7230dadc 043bfa9c 76b619f1 03d85e88 mshtml!CExecFT::ThreadProc+0x3c
043bfa90 76b619f1 03d85e88 043bfadc 778cd109 mshtml!CExecFT::StaticThreadProc+0xd
043bfa9c 778cd109 03d85e88 043bdb05 00000000 kernel32!BaseThreadInitThunk+0xe
043bfadc 00000000 7230dacf 03d85e88 00000000 ntdll_77850000!_RtlUserThreadStart+0x23

[...]

Effective machine: x64 (AMD64)
THREAD fffffa8001220060  Cid 03bc.0b34  Teb: 000007fffffde000 Win32Thread: fffff900c30cd460 RUNNING on processor 1
IRP List:
    fffffa80033f15a0: (0006,0118) Flags: 00060000  Mdl: 00000000
Not impersonating
DeviceMap                 fffff88001a8a970
Owning Process            fffffa8001412040       Image:         NotMyfault.exe
Attached Process          N/A            Image:         N/A
Wait Start TickCount      42458          Ticks: 0
Context Switch Count      240            IdealProcessor: 1                 LargeStack
UserTime                  00:00:00.015
KernelTime                00:00:00.093
*** ERROR: Module load completed but symbols could not be loaded for NotMyfault.exe
Win32 Start Address NotMyfault (0x0000000140002708)
Stack Init fffff98011da0db0 Current fffff98011da0740
Base fffff98011da1000 Limit fffff98011d98000 Call 0
Priority 13 BasePriority 8 PriorityDecrement 5 IoPriority 2 PagePriority 5
*** ERROR: Module load completed but symbols could not be loaded for myfault.sys
Child-SP          RetAddr           Call Site
fffff980`11da0568 fffff800`0184dff3 nt!KeBugCheckEx
fffff980`11da0570 fffff800`0184cecb nt!KiBugCheckDispatch+0x73
fffff980`11da06b0 fffff980`0e6e117a nt!KiPageFault+0x20b (TrapFrame @ fffff980`11da06b0)
fffff980`11da0840 fffff980`0e6e1397 myfault+0x117a
fffff980`11da09a0 fffff800`01a8e217 myfault+0x1397
fffff980`11da0a00 fffff800`01a94466 nt!IopXxxControlFile+0x626
fffff980`11da0b40 fffff800`0184dcf3 nt!NtDeviceIoControlFile+0x56
fffff980`11da0bb0 00000000`776f02ea nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff980`11da0c20)
00000000`0012f5c8 00000000`774b21da ntdll!ZwDeviceIoControlFile+0xa
00000000`0012f5d0 00000001`40001cc9 kernel32!DeviceIoControl+0xaa
00000000`0012f650 00000000`775ee26a NotMyfault+0x1cc9
00000000`0012f830 00000000`775eb7d9 USER32!UserCallWinProcCheckWow+0x1ad
00000000`0012f8f0 00000000`775eb86d USER32!SendMessageWorker+0x64a
00000000`0012f980 00000000`77611ad4 USER32!SendMessageW+0x5b
00000000`0012f9d0 00000000`7760080b USER32!xxxBNReleaseCapture+0x444
00000000`0012fa40 00000000`77612886 USER32!ButtonWndProcWorker+0x1490
00000000`0012fb90 00000000`775ee26a USER32!ButtonWndProcA+0x66
00000000`0012fbd0 00000000`775eec9f USER32!UserCallWinProcCheckWow+0x1ad
00000000`0012fc90 00000000`775e0e63 USER32!DispatchMessageWorker+0x389
00000000`0012fd10 00000000`776276ba USER32!IsDialogMessageW+0x14f
00000000`0012fda0 00000001`40001ed1 USER32!IsDialogMessageA+0x7a
00000000`0012fdd0 00000001`400026b5 NotMyfault+0x1ed1
00000000`0012feb0 00000000`774ccdcd NotMyfault+0x26b5
00000000`0012ff60 00000000`776ec6e1 kernel32!BaseThreadInitThunk+0xd
00000000`0012ff90 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

Implicit thread is now fffffa80`01220060
The context is partially valid. Only x86 user-mode context is available.
x86 context set
Loading Kernel Symbols
...............................................................
...............................................................
Loading User Symbols
.................
Loading unloaded module list
.........Unable to enumerate user-mode unloaded modules, NTSTATUS 0xC0000147
ChildEBP          RetAddr           Args to Child
WARNING: Frame IP not in any known module. Following frames may be wrong.
00000000 00000000 00000000 00000000 00000000 0x0
Effective machine: x64 (AMD64)
```

11. Yet another way is to use **!stacks** command (the default version omits paged out stacks):

```
1: kd> !stacks
Proc.Thread  .Thread  Ticks   ThreadState Blocker
*** ERROR: Module load completed but symbols could not be loaded for intelppm.sys
                           [fffff8000194f300 Idle]
   0.000000  fffff8000194eb80 ffff5a4e RUNNING    intelppm+0x36d2
   0.000000  fffff98000dc1c40 0000000 RUNNING    nt!KiIdleLoop+0xf6
                           [fffffa8000c360b0 System]
   4.00002c  fffffa8000c85bb0 fffffb8d Blocked    msiscsi!iSpPersistentLoginsWorker+0x8b
   4.000060  fffffa8000cad040 ffffa5fe Blocked    nt!MiModifiedPageWriter+0x5e
   4.000070  fffffa8000cae600 ffff5a65 Blocked    nt!CcQueueLazyWriteScanThread+0x73
   4.000080  fffffa8000c4fbb0 ffff5a4e Blocked    nt!EtwpLogger+0xdd
   4.000084  fffffa8000c58040 ffff5a4e Blocked    nt!EtwpLogger+0xdd
   4.000088  fffffa80010fc040 ffff5a60 Blocked    nt!EtwpLogger+0xdd
   4.00008c  fffffa80010fc7f0 ffff5a2f Blocked    nt!EtwpLogger+0xdd
   4.000090  fffffa8001145040 ffffff8a Blocked    nt!EtwpLogger+0x84
   4.000094  fffffa8001145670 ffffff8a Blocked    nt!EtwpLogger+0x84
   4.000098  fffffa8001186bb0 ffffff8a Blocked    nt!EtwpLogger+0x84
   4.0000a0  fffffa80019de040 ffffce80 Blocked    nt!EtwpLogger+0xdd
   4.0000a4  fffffa80019df8b0 ffff5a84 Blocked    nt!WdipSemCheckTimeout+0x128
   4.0000a8  fffffa8001a13040 ffff5cda Blocked    acpi!ACPIWorkerThread+0x74
   4.0000b0  fffffa8002bf5390 fffffea8 Blocked    acpi!PciRootBusBiosMethodDispatcherOnResume+0x51
   4.0000b4  fffffa8002bf81a0 fffffea2 Blocked    pci!ExpressRootComplexPmeEventDispatcher+0x57
   4.0000bc  fffffa8001bd3450 ffff5fff Blocked    ndis!ndisCmWaitThread+0x6e
   4.0000c0  fffffa80025c5bb0 ffffdafb Blocked    ecache!EcCacheIoWorker+0x63
   4.0000c4  fffffa8001bd5450 ffffdb3f Blocked    ecache!EcCacheIoWatchdog+0x36d
   4.0000cc  fffffa8001c1fbb0 ffff5aa7 Blocked    Ntfs!TxfPrivateThreadWorkerRoutine+0x3f
   4.0000d4  fffffa8001caa270 fffffc23 Blocked    dxgkrnl!DpiPdoPollingThread+0x51
   4.0000d8  fffffa8001cae3a0 fffffc23 Blocked    watchdog!SMgrGdiCalloutThread+0x5d
   4.0000dc  fffffa8001ca03a0 fffffc23 Blocked    dxgkrnl!DpiPowerArbiterThread+0x4b
*** ERROR: Module load completed but symbols could not be loaded for vmci.sys
   4.0000f4  fffffa8001d02740 fffffbba Blocked    vmci+0x4d3f
   4.0000f8  fffffa8001cfe3a0 fffffbba Blocked    vmci+0x5087
   4.0001b0  fffffa80024136e0 ffff5ac8 Blocked    dxgkrnl!VidSchiWaitForSchedulerEvents+0x161
   4.0001ec  fffffa800248b9d0 ffff5ae1 Blocked    nt!IoRemoveIoCompletion+0x47
   4.00025c  fffffa8002501bb0 ffffde44 Blocked    nt!EtwpLogger+0xdd
   4.0002fc  fffffa800208cbb0 ffff5a5d Blocked    luafv!SynchronousFsControl+0x102
*** ERROR: Module load completed but symbols could not be loaded for spsys.sys
   4.000378  fffffa800341b950 ffffeb0a Blocked    spsys+0x46dc1
   4.0004c4  fffffa80035de040 ffff713e Blocked    HTTP!UlpScavengerThread+0x81
   4.000510  fffffa80034d06c0 ffffde29 Blocked    mpsdrv!IP6StringToAddress+0x738
   4.000554  fffffa80038c0710 ffffe8d4 Blocked    srv2!SrvProcWorkerThread+0x74
   4.000558  fffffa8002043040 ffffe8d3 Blocked    srv2!SrvProcWorkerThread+0x74
   4.00055c  fffffa80035fd040 ffffe8d2 Blocked    srv2!SrvProcWorkerThread+0x74
   4.000560  fffffa8002043bb0 ffffe8d1 Blocked    srv2!SrvProcWorkerThread+0x74
   4.000564  fffffa8002043720 ffffe8d0 Blocked    srv2!SrvProcWorkerThread+0x74
*** ERROR: Module load completed but symbols could not be loaded for vmmemctl.sys
   4.0005b4  fffffa8004063bb0 ffff5a47 Blocked    vmmemctl+0x13e3
   4.000714  fffffa8003989bb0 ffffe7bc Blocked    nt!EtwpLogger+0xdd
   4.0007f8  fffffa8003b15bb0 ffffe60d Blocked    nt!EtwpLogger+0xdd
   4.000acc  fffffa800412d060 ffffde3e Blocked    nt!EtwpLogger+0xdd
   4.000b48  fffffa8004108060 ffff5a5d Blocked    nt!PfTLoggingWorker+0x81
   4.0007a0  fffffa80031335d0 ffff5a60 Blocked    nt!EtwpLogger+0xdd
   4.000724  fffffa8004122060 ffffdabc Blocked    nt!EtwpLogger+0xdd
   4.0001d4  fffffa8001214040 ffffce07 Blocked    nt!EtwpLogger+0x84

                           [fffffa8002366c10 smss.exe]

                           [fffffa80023cba80 csrss.exe]
 1a4.0001bc  fffffa80024437c0 ffff5a60 Blocked    nt!AlpcpReceiveMessagePort+0x298
 1a4.0001d0  fffffa80023ef650 ffff5a60 Blocked    nt!AlpcpReceiveMessagePort+0x298
 1a4.000240  fffffa80024d1bb0 ffff5a5d Blocked    nt!AlpcpReceiveMessagePort+0x298

[...]

                           [fffffa8001412040 NotMyfault.exe]
 3bc.000b34  fffffa8001220060 ffff5a26 RUNNING    nt!KeBugCheckEx
```

12. Let's now check processes that were waiting for user input:

```
1: kd> !stacks 2 win32k!NtUserGetMessage
Proc.Thread  .Thread  Ticks   ThreadState Blocker
                            [fffff8000194f300 Idle]
                            [fffffa8000c360b0 System]

                            [fffffa8002366c10 smss.exe]

                            [fffffa80023cba80 csrss.exe]

                            [fffffa800244dc10 wininit.exe]

                            [fffffa8002453c10 csrss.exe]

                            [fffffa800245fc10 winlogon.exe]

                            [fffffa80024ad920 services.exe]

                            [fffffa80024d7c10 lsass.exe]

                            [fffffa80024dbc10 lsm.exe]

                            [fffffa800349e9d0 svchost.exe]

                            [fffffa800209f880 svchost.exe]

                            [fffffa8002090c10 svchost.exe]

                            [fffffa8002f10970 svchost.exe]

                            [fffffa80031c54f0 svchost.exe]

                            [fffffa8003325540 svchost.exe]

                            [fffffa800335f040 audiodg.exe]

                            [fffffa80033902b0 SLsvc.exe]

                            [fffffa8003424ae0 svchost.exe]

                            [fffffa8003475840 svchost.exe]

                            [fffffa80035db8e0 spoolsv.exe]

                            [fffffa80035edae0 svchost.exe]

                            [fffffa80038ee9b0 svchost.exe]

                            [fffffa800392c2e0 vmtoolsd.exe]

                            [fffffa80039a2940 dwm.exe]

                            [fffffa80039bac10 explorer.exe]
 734.000840  fffffa8002f52060 ffff5afd Blocked    nt!KiSwapContext+0x84
                                       nt!KiSwapThread+0x125
                                       nt!KeWaitForSingleObject+0x5f5
                                       win32k!xxxRealSleepThread+0x278
                                       win32k!xxxSleepThread+0x56
                                       win32k!xxxRealInternalGetMessage+0x72e
```

```
                                     win32k!xxxInternalGetMessage+0x35
                                     win32k!NtUserGetMessage+0x64
                                     nt!KiSystemServiceCopyEnd+0x13
                                     USER32!ZwUserGetMessage+0xa
734.0005b8  fffffa8002f08bb0 ffff5a29 Blocked    nt!KiSwapContext+0x84
                                     nt!KiSwapThread+0x125
                                     nt!KeWaitForSingleObject+0x5f5
                                     win32k!xxxRealSleepThread+0x278
                                     win32k!xxxSleepThread+0x56
                                     win32k!xxxRealInternalGetMessage+0x72e
                                     win32k!xxxInternalGetMessage+0x35
                                     win32k!NtUserGetMessage+0x64
                                     nt!KiSystemServiceCopyEnd+0x13
                                     USER32!ZwUserGetMessage+0xa
734.0009e0  fffffa8001205bb0 ffff5c77 Blocked    nt!KiSwapContext+0x84
                                     nt!KiSwapThread+0x125
                                     nt!KeWaitForSingleObject+0x5f5
                                     win32k!xxxRealSleepThread+0x278
                                     win32k!xxxSleepThread+0x56
                                     win32k!xxxRealInternalGetMessage+0x72e
                                     win32k!xxxInternalGetMessage+0x35
                                     win32k!NtUserGetMessage+0x64
                                     nt!KiSystemServiceCopyEnd+0x13
                                     USER32!ZwUserGetMessage+0xa
734.0009a8  fffffa80013cc060 ffff5c0f Blocked    nt!KiSwapContext+0x84
                                     nt!KiSwapThread+0x125
                                     nt!KeWaitForSingleObject+0x5f5
                                     win32k!xxxRealSleepThread+0x278
                                     win32k!xxxSleepThread+0x56
                                     win32k!xxxRealInternalGetMessage+0x72e
                                     win32k!xxxInternalGetMessage+0x35
                                     win32k!NtUserGetMessage+0x64
                                     nt!KiSystemServiceCopyEnd+0x13
                                     USER32!ZwUserGetMessage+0xa

                           [fffffa80039f5040 svchost.exe]

                           [fffffa80039fd040 SearchIndexer.e]

                           [fffffa8003bac760 VMUpgradeHelper]

                           [fffffa8003b500a0 taskeng.exe]
61c.000860  fffffa8003b106c0 ffff5ae1 Blocked    nt!KiSwapContext+0x84
                                     nt!KiSwapThread+0x125
                                     nt!KeWaitForSingleObject+0x5f5
                                     win32k!xxxRealSleepThread+0x278
                                     win32k!xxxSleepThread+0x56
                                     win32k!xxxRealInternalGetMessage+0x72e
                                     win32k!xxxInternalGetMessage+0x35
                                     win32k!NtUserGetMessage+0x64
                                     nt!KiSystemServiceCopyEnd+0x13
                                     USER32!ZwUserGetMessage+0xa

                           [fffffa8003b9ac10 TPAutoConnSvc.e]

                           [fffffa80039f1c10 taskeng.exe]

                           [fffffa8004039040 TPAutoConnect.e]

                           [fffffa80040a3410 dllhost.exe]
```

```
                          [fffffa80040fba50 MSASCui.exe]

                          [fffffa80040cc280 VMwareTray.exe]
a00.000a04  fffffa8002042060 ffff5a65 Blocked    nt!KiSwapContext+0x84
                                      nt!KiSwapThread+0x125
                                      nt!KeWaitForSingleObject+0x5f5
                                      win32k!xxxRealSleepThread+0x278
                                      win32k!xxxSleepThread+0x56
                                      win32k!xxxRealInternalGetMessage+0x72e
                                      win32k!xxxInternalGetMessage+0x35
                                      win32k!NtUserGetMessage+0x64
                                      nt!KiSystemServiceCopyEnd+0x13
                                      USER32!ZwUserGetMessage+0xa
                                      USER32!ButtonWndProcWorker+0xb
                                      +0xa03cc
                                      +0x1
                                      +0x188ee0
                                      USER32!SendMessageW+0x5b
                                      USER32!xxxBNReleaseCapture+0x444
                                      USER32!ButtonWndProcWorker+0x1490
                                      USER32!ButtonWndProcA+0x66
                                      USER32!UserCallWinProcCheckWow+0x1ad
                                      USER32!DispatchMessageWorker+0x389
                                      USER32!IsDialogMessageW+0x14f
                                      USER32!IsDialogMessageA+0x7a
                                      NotMyfault+0x1ed1
                                      NotMyfault+0x26b5
                                      kernel32!BaseThreadInitThunk+0xd
                                      ntdll!RtlUserThreadStart+0x1d

                          [fffffa80020a1c10 VMwareUser.exe]

                          [fffffa8003994990 sidebar.exe]

                          [fffffa80040e1ae0 msdtc.exe]

                          [fffffa8001361040 WmiPrvSE.exe]

                          [fffffa8001519040 WmiPrvSE.exe]
8e8.0005f4  fffffa8001418770 ffff68d8 Blocked    nt!KiSwapContext+0x84
                                      nt!KiSwapThread+0x125
                                      nt!KeWaitForSingleObject+0x5f5
                                      win32k!xxxRealSleepThread+0x278
                                      win32k!xxxSleepThread+0x56
                                      win32k!xxxRealInternalGetMessage+0x72e
                                      win32k!xxxInternalGetMessage+0x35
                                      win32k!NtUserGetMessage+0x64
                                      nt!KiSystemServiceCopyEnd+0x13
                                      USER32!ZwUserGetMessage+0xa

                          [fffffa80014f0040 ieuser.exe]

                          [fffffa80040d2040 iexplore.exe]

                          [fffffa8001571040 SearchProtocolH]

                          [fffffa80014ce2f0 SearchFilterHos]
```

```
                          [fffffa8001579040 notepad.exe]
 948.00096c  fffffa8001586bb0 ffff5f90 Blocked    nt!KiSwapContext+0x84
                                      nt!KiSwapThread+0x125
                                      nt!KeWaitForSingleObject+0x5f5
                                      win32k!xxxRealSleepThread+0x278
                                      win32k!xxxSleepThread+0x56
                                      win32k!xxxRealInternalGetMessage+0x72e
                                      win32k!xxxInternalGetMessage+0x35
                                      win32k!NtUserGetMessage+0x64
                                      nt!KiSystemServiceCopyEnd+0x13
                                      USER32!ZwUserGetMessage+0xa

                          [fffffa800159fb40 dllhost.exe]
 7c8.0004cc  fffffa800156d060 ffff5b11 Blocked    nt!KiSwapContext+0x84
                                      nt!KiSwapThread+0x125
                                      nt!KeWaitForSingleObject+0x5f5
                                      win32k!xxxRealSleepThread+0x278
                                      win32k!xxxSleepThread+0x56
                                      win32k!xxxRealInternalGetMessage+0x72e
                                      win32k!xxxInternalGetMessage+0x35
                                      win32k!NtUserGetMessage+0x64
                                      nt!KiSystemServiceCopyEnd+0x13
                                      USER32!ZwUserGetMessage+0xa

                          [fffffa8001579c10 dllhost.exe]
 8c8.00057c  fffffa80015a8060 ffff5ae8 Blocked    nt!KiSwapContext+0x84
                                      nt!KiSwapThread+0x125
                                      nt!KeWaitForSingleObject+0x5f5
                                      win32k!xxxRealSleepThread+0x278
                                      win32k!xxxSleepThread+0x56
                                      win32k!xxxRealInternalGetMessage+0x72e
                                      win32k!xxxInternalGetMessage+0x35
                                      win32k!NtUserGetMessage+0x64
                                      nt!KiSystemServiceCopyEnd+0x13
                                      USER32!ZwUserGetMessage+0xa

                         [fffffa8001412040 NotMyfault.exe]


Threads Processed: 590
```

Note: We also see the effect of wrong symbol mapping for VMwareTray.exe process (stack trace module is from a different process, NotMyFault.exe) so we set it as a current process and repeat the command. This time we don't get surprises:

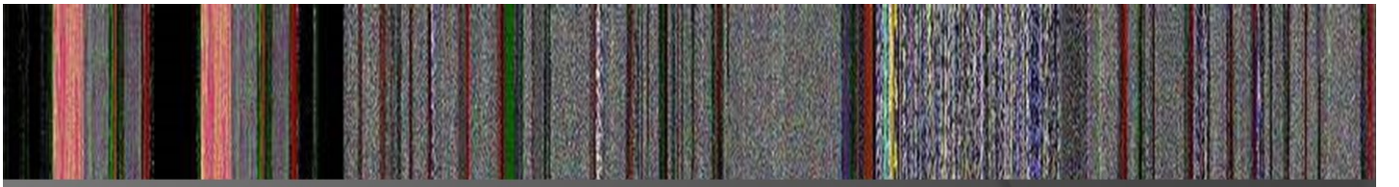
```
1: kd> .process /r /p fffffa80040cc280
Implicit process is now fffffa80`040cc280
Loading User Symbols
.....................................

************* Symbol Loading Error Summary **************
Module name            Error
vmci                   The system cannot find the file specified
intelppm               The system cannot find the file specified
spsys                  The system cannot find the file specified
vmmemctl               The system cannot find the file specified
myfault                The system cannot find the file specified
```

```
You can troubleshoot most symbol related issues by turning on symbol loading diagnostics (!sym
noisy) and repeating the command that caused symbols to be loaded.
You should also verify that your symbol search path (.sympath) is correct.

1: kd> !stacks 2 win32k!NtUserGetMessage
Proc.Thread  .Thread  Ticks   ThreadState Blocker

[...]

                            [fffffa80040cc280 VMwareTray.exe]
 a00.000a04  fffffa8002042060 ffff5a65 Blocked    nt!KiSwapContext+0x84
                                       nt!KiSwapThread+0x125
                                       nt!KeWaitForSingleObject+0x5f5
                                       win32k!xxxRealSleepThread+0x278
                                       win32k!xxxSleepThread+0x56
                                       win32k!xxxRealInternalGetMessage+0x72e
                                       win32k!xxxInternalGetMessage+0x35
                                       win32k!NtUserGetMessage+0x64
                                       nt!KiSystemServiceCopyEnd+0x13
                                       USER32!ZwUserGetMessage+0xa
                                       USER32!GetMessageW+0x34
                                       VMwareTray+0x13be

[...]
```

13. We close logging before exiting WinDbg:

```
1: kd> .logclose
Closing open log file C:\AdvWMDA-Dumps\64-bit\Complete\C1.log
```

Note: To avoid possible confusion and glitches we recommend to exit WinDbg after each exercise.



Windows RT Memory Dump Analysis Advanced

ARM Version

Dmitry Vostokov
Software Diagnostics Services

Published by OpenTask, Republic of Ireland

OpenTask books and magazines are available through booksellers and distributors worldwide. For further information or comments send requests to press@opentask.com.

A CIP catalogue record for this book is available from the British Library.

ISBN-l3: 978-1-908043-73-3 (Paperback)

Version 1, 2014

Contents

Exercise C2: Memory Search (ARM)

Goal: Learn how to recognize byte ordering conventions and search memory for specific values.

Patterns: Value References

1. Launch WinDbg from Windows Kits \ Debugging Tools for Windows (X64).

2. Open \AdvMDA-Dumps\ARM\Complete\MEMORY.DMP

3. We get the dump file loaded (the output should be the same as in the previous exercise).

4. We open a log file, set up symbols and reload them:

```
0: kd> .logopen C:\AdvWMDA-Dumps\ARM\Complete\C2-ARM.log
Opened log file 'C:\AdvWMDA-Dumps\ARM\Complete\C2-ARM.log'

0: kd> .symfix c:\mss

0: kd> .reload
Loading Kernel Symbols
...............................................................
................................................................
.............................
Loading User Symbols

Loading unloaded module list
.......
```

5. We check nonpaged pool consumption:

```
0: kd> !poolused 2
....
 Sorting by NonPaged Pool Consumed

               NonPaged                  Paged
 Tag     Allocs         Used     Allocs         Used

 ConT        28      4915200          0            0      UNKNOWN pooltag 'ConT', please update
pooltag.txt
 NVRM       517      3005728          0            0      UNKNOWN pooltag 'NVRM', please update
pooltag.txt
 EtwB        73      2207760          2       131072      Etw Buffer , Binary: nt!etw
 FVEp         5      2101248          0            0      Write buffers , Binary: fvevol.sys
 MOAL      1312      2087816          0            0      UNKNOWN pooltag 'MOAL', please update
pooltag.txt
 Pool         5      1150080          0            0      Pool tables, etc.
 Thre       627       760144          0            0      Thread objects , Binary: nt!ps
 File      3120       569456          0            0      File objects
 ViSh       428       523648          0            0      Video scheduler , Binary: dxgkrnl.sys
 ViMm      1496       480248       2447      1089224      Video memory manager , Binary: dxgkrnl.sys
 Ntfx      2497       464776          0            0      General Allocation , Binary: ntfs.sys
 FMsl      2321       371360          0            0      STREAM_LIST_CTRL structure , Binary: fltmgr.sys
 HTab        93       318168          0            0      Hash Table pool
 EtwG       883       317880          0            0      Etw Guid , Binary: nt!etw
```

```
 Devi       342       305128          0            0      Device objects
 MmCa      1936       300064          0            0      Mm control areas for mapped files , Binary:
nt!mm
 AfdB        24       275528          0            0      Afd data buffer , Binary: afd.sys
 MmPb         2       270336          0            0      Paging file bitmaps , Binary: nt!mm
 FIvp         4       270192          0            0      FileInfo FS-filter Volume Properties , Binary:
fileinfo.sys
 NvFX         2       268928          0            0      UNKNOWN pooltag 'NvFX', please update
pooltag.txt
 Even      4146       266784          0            0      Event objects
 AmlH         4       262144          0            0      ACPI AMLI Pooltags
 NvLM         1       262144          0            0      UNKNOWN pooltag 'NvLM', please update
pooltag.txt
 Vad       3176       254080          0            0      Mm virtual address descriptors , Binary: nt!mm
 NvLH       385       251288         23        28976      nVidia video driver , Binary: <nvlddmkm.sys>
 ALPC       742       244688          0            0      ALPC port objects , Binary: nt!alpc
 MmCi       752       243104          0            0      Mm control areas for images , Binary: nt!mm
 EtwR      2425       220272          0            0      Etw Registration , Binary: nt!etw
 NtFs         8       212760       3177       272504      StrucSup.c , Binary: ntfs.sys
 CDmp        11       209896          0            0      Crashdump driver , Binary: crashdmp.sys
 Lfsr         3       208896          3          360      UNKNOWN pooltag 'Lfsr', please update
pooltag.txt
 NvLL        15       196784          0            0      UNKNOWN pooltag 'NvLL', please update
pooltag.txt
 Mm           9       187360          4          136      general Mm Allocations , Binary: nt!mm
 NvLa       570       180000          0            0      nVidia video driver , Binary: <nvlddmkm.sys>
[...]
```

Note: Suppose we are interested in drivers that allocated memory with *NVRM* and *MOAL* pool tags.

6. Pool allocations with tags are done using *ExAllocatePoolWithTag* or *ExAllocatePoolWithTagPriority* functions where a 4-byte pool tag is a function parameter. When module is built from code pool tags as strings will be saved in it. Let's now determine pool tag binary equivalents corresponding to their memory layout:

```
0: kd> ? 'NVRM'
Evaluate expression: 1314280013 = 4e56524d

0: kd> ? 'MOAL'
Evaluate expression: 1297039692 = 4d4f414c
```

7. We now search every module for such values. The command to search virtual memory is **s.** So if we want to find 12345678 (78 56 34 12 as a byte stream, ARM has the same byte ordering as x86/x64) in the memory region PC – PC+100 we use this command (PC is a CPU register pointing to the next instruction to execute, in our case 83587dd0). Let's write this value to memory pointed to by PC for this exercise:

```
0: kd> r pc
pc=83587dd0

0: kd> ed @pc 12345678

0: kd> dc 83587dd0 L1
83587dd0  12345678                                xV4.

0: kd> dw 83587dd0 L4
83587dd0  5678 1234 f738 fb12
```

```
0: kd> dc 83587dd0 L1
83587dd0  12345678                                xV4.

0: kd> db 83587dd0 L8
83587dd0  78 56 34 12 38 f7 12 fb                         xV4.8...

0: kd> s-d @pc @pc+100 12345678
83587dd0  12345678 fb12f738 1518f8d4 f44fa820  xV4.8....... .O.
```

Note: If we search for word values we shouldn't forget to respect alignment boundary:

```
0: kd> s-w @pc @pc+100 1234
83587dd2  1234 f738 fb12 f8d4 1518 a820 f44f 72d0  4.8....... .O..r

0: kd> s-w @pc+1 @pc+100 1234
```

8. To find module that uses chosen pool tags above we repeat memory byte search for every module:

```
0: kd> ? 'NVRM'
Evaluate expression: 1314280013 = 4e56524d

0: kd> !for_each_module "s-b @#Base @#End 4e 56 52 4d"
83f7c138  4e 56 52 4d 20 43 4c 4f-43 4b 53 3a 20 50 4c 4c  NVRM CLOCKS: PLL
83f7c15c  4e 56 52 4d 20 43 4c 4f-43 4b 53 3a 20 50 4c 4c  NVRM CLOCKS: PLL
83f7c180  4e 56 52 4d 20 43 4c 4f-43 4b 53 3a 20 50 4c 4c  NVRM CLOCKS: PLL
83f7c1a4  4e 56 52 4d 20 43 4c 4f-43 4b 53 3a 20 50 4c 4c  NVRM CLOCKS: PLL
83f7c1c8  4e 56 52 4d 20 43 4c 4f-43 4b 53 3a 20 50 4c 4c  NVRM CLOCKS: PLL
83f7c1ec  4e 56 52 4d 20 43 4c 4f-43 4b 53 3a 20 50 4c 4c  NVRM CLOCKS: PLL
83f7c210  4e 56 52 4d 20 43 4c 4f-43 4b 53 3a 20 43 50 55  NVRM CLOCKS: CPU
83f7c234  4e 56 52 4d 20 43 4c 4f-43 4b 53 3a 20 41 56 50  NVRM CLOCKS: AVP
83f7c258  4e 56 52 4d 20 43 4c 4f-43 4b 53 3a 20 53 79 73  NVRM CLOCKS: Sys
83f7c27c  4e 56 52 4d 20 43 4c 4f-43 4b 53 3a 20 4d 65 6d  NVRM CLOCKS: Mem
83f7c2a0  4e 56 52 4d 20 43 4c 4f-43 4b 53 3a 20 45 78 74  NVRM CLOCKS: Ext
9155988c  4e 56 52 4d 00 20 5f 91-2d e9 3c 48 0d f2 10 0b  NVRM. _.-.<H....
9156fe14  4e 56 52 4d d4 22 5f 91-00 20 5f 91 88 74 61 91  NVRM."_.. _..ta.
9156ffc4  4e 56 52 4d d4 22 5f 91-30 ea 66 91 2d e9 f0 4b  NVRM."_.0.f.-..K
91570080  4e 56 52 4d d4 22 5f 91-00 38 61 91 30 ea 66 91  NVRM."_..8a.0.f.
9157022c  4e 56 52 4d d4 22 5f 91-10 38 61 91 f0 a3 5f 91  NVRM."_..8a..._.
915d666c  4e 56 52 4d d4 22 5f 91-00 38 61 91 88 74 61 91  NVRM."_..8a..ta.
915d67bc  4e 56 52 4d d4 22 5f 91-2d e9 f0 48 0d f2 10 0b  NVRM."_.-..H....
915d6918  4e 56 52 4d 2d e9 70 48-0d f2 0c 0b 89 b0 00 23  NVRM-.pH.......#
91613cd8  4e 56 52 4d 2d 52 43 3a-20 4e 76 69 64 69 61 20  NVRM-RC: Nvidia
91614c90  4e 56 52 4d 2d 52 43 3a-20 4e 76 69 64 69 61 20  NVRM-RC: Nvidia
91614cb8  4e 56 52 4d 2d 52 43 3a-20 4e 76 69 64 69 61 20  NVRM-RC: Nvidia
91615650  4e 56 52 4d 2d 52 43 3a-20 4e 76 69 64 69 61 20  NVRM-RC: Nvidia
9161ea34  4e 56 52 4d 2d 52 43 3a-20 4e 76 69 64 69 61 20  NVRM-RC: Nvidia
```

Note: When choosing between address to inspect we should start first with values that have a non-ASCII terminator such as 00 because it less likely that a part of ASCII string was used as a pool tag. Here we see that the address 9155988c is in *nvlddmkm* module address range:

```
0: kd> lmk
start    end        module name
[...]
9142a000 916ac000   nvlddmkm   (export symbols)       nvlddmkm.sys
[...]
```

```
0: kd> lmv m nvlddmkm
start    end        module name
9142a000 916ac000   nvlddmkm   (export symbols)       nvlddmkm.sys
    Loaded symbol image file: nvlddmkm.sys
    Image path: \SystemRoot\system32\DRIVERS\nvlddmkm.sys
    Image name: nvlddmkm.sys
    Timestamp:        Wed Jun 12 09:51:11 2013 (51B8367F)
    CheckSum:         0026E2D3
    ImageSize:        00282000
    Translations:     0000.04b0 0000.04e4 0409.04b0 0409.04e4

0: kd> !lmi nvlddmkm
Loaded Module Info: [nvlddmkm]
         Module: nvlddmkm
   Base Address: 9142a000
     Image Name: nvlddmkm.sys
   Machine Type: 452 (ARMNT)
     Time Stamp: 51b8367f Wed Jun 12 09:51:11 2013
           Size: 282000
       CheckSum: 26e2d3
Characteristics: 122
Debug Data Dirs: Type  Size     VA  Pointer
             CODEVIEW    85, 1fd2e8,  1fbce8 RSDS - GUID: {601B4A1A-E5E2-49D8-A8B5-
12E5F9E2552A}
                Age: 1, Pdb:
c:\dvs\p4\build\sw\rel\tegra_drv\w8ot\drivers\display\lddm\nvlddmkm\_out\T30_win8_ARMv7_release
\nvlddmkm.pdb
                   ??    10, 1fd370,  1fbd70 [Data not mapped]
     Image Type: MEMORY   - Image read successfully from loaded memory.
    Symbol Type: EXPORT   - PDB not found
    Load Report: export symbols
```

9. We continue with the second pool tag:

```
0: kd> !for_each_module "s-b @#Base @#End 'M' 'O' 'A' 'L'"
916c7374  4d 4f 41 4c 40 b0 75 91-c8 f9 73 91 b4 f9 73 91  MOAL@.u...s...s.
916ef2dc  4d 4f 41 4c 00 00 00 00-00 00 00 2d e9 1e 48  MOAL........-..H
9171dba1  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
9171dbe1  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
9171dc29  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
9171dc71  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
[...]
9171f161  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
9171f199  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
9171f1bb  4d 4f 41 4c 20 6d 61 69-6e 20 6d 6f 64 75 6c 65  MOAL main module
9171f1e1  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
9171f231  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
[,,,]
91722d14  4d 4f 41 4c 5f 41 53 53-4f 43 5f 53 54 41 54 55  MOAL_ASSOC_STATU
91722d49  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
[...]
91724501  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
91724548  4d 4f 41 4c 5f 43 4f 4e-4e 45 43 54 5f 53 54 41  MOAL_CONNECT_STA
91724581  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
[...]
917294e1  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
91729541  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
9172a564  4d 4f 41 4c 5f 4f 49 44-5f 48 4f 53 54 5f 46 57  MOAL_OID_HOST_FW
9172a57c  4d 4f 41 4c 5f 4f 49 44-5f 44 45 45 50 5f 53 4c  MOAL_OID_DEEP_SL
```

```
9172a590  4d 4f 41 4c 5f 4f 49 44-5f 52 45 47 5f 43 4d 44  MOAL_OID_REG_CMD
9172a5a4  4d 4f 41 4c 5f 4f 49 44-5f 4c 45 44 5f 43 4f 4e  MOAL_OID_LED_CON
9172a5bc  4d 4f 41 4c 5f 4f 49 44-5f 4e 4c 4f 5f 43 4d 44  MOAL_OID_NLO_CMD
9172a5d0  4d 4f 41 4c 5f 4f 49 44-5f 51 55 45 52 59 5f 53  MOAL_OID_QUERY_S
9172a5e8  4d 4f 41 4c 5f 4f 49 44-5f 47 45 54 5f 46 57 5f  MOAL_OID_GET_FW_
9172a5fc  4d 4f 41 4c 5f 4f 49 44-5f 54 58 5f 50 4f 57 45  MOAL_OID_TX_POWE
9172b008  4d 4f 41 4c 5f 4f 49 44-5f 57 4f 4c 5f 50 41 54  MOAL_OID_WOL_PAT
9172b540  4d 4f 41 4c 5f 4f 49 44-5f 45 41 50 4f 4c 5f 53  MOAL_OID_EAPOL_S
9172b558  4d 4f 41 4c 5f 4f 49 44-5f 4c 49 4e 4b 5f 53 54  MOAL_OID_LINK_ST
9172b56c  4d 4f 41 4c 5f 4f 49 44-5f 4d 4f 45 4d 4f 52 59  MOAL_OID_MOEMORY
9172b584  4d 4f 41 4c 5f 50 4e 50-5f 45 56 45 4e 54 5f 51  MOAL_PNP_EVENT_Q
9172b5a4  4d 4f 41 4c 5f 50 4e 50-5f 45 56 45 4e 54 5f 52  MOAL_PNP_EVENT_R
9172b5bc  4d 4f 41 4c 5f 50 4e 50-5f 45 56 45 4e 54 5f 53  MOAL_PNP_EVENT_S
9172b5e0  4d 4f 41 4c 5f 50 4e 50-5f 45 56 45 4e 54 5f 51  MOAL_PNP_EVENT_Q
9172b600  4d 4f 41 4c 5f 50 4e 50-5f 45 56 45 4e 54 5f 53  MOAL_PNP_EVENT_S
9172b618  4d 4f 41 4c 5f 50 4e 50-5f 45 56 45 4e 54 5f 50  MOAL_PNP_EVENT_P
9172bddc  4d 4f 41 4c 5f 43 4f 4e-4e 45 43 54 5f 53 54 41  MOAL_CONNECT_STA
9172bdfc  4d 4f 41 4c 5f 43 4f 4e-4e 45 43 54 5f 53 54 41  MOAL_CONNECT_STA
9172be1c  4d 4f 41 4c 5f 43 4f 4e-4e 45 43 54 5f 53 54 41  MOAL_CONNECT_STA
9172be48  4d 4f 41 4c 5f 43 4f 4e-4e 45 43 54 5f 53 54 41  MOAL_CONNECT_STA
9172be68  4d 4f 41 4c 5f 43 4f 4e-4e 45 43 54 5f 53 54 41  MOAL_CONNECT_STA
9172be8c  4d 4f 41 4c 5f 43 4f 4e-4e 45 43 54 5f 53 54 41  MOAL_CONNECT_STA
9172bea8  4d 4f 41 4c 5f 43 4f 4e-4e 45 43 54 5f 53 54 41  MOAL_CONNECT_STA
9172c0e1  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
9172c13c  4d 4f 41 4c 5f 46 53 4d-5f 4d 41 58 5f 4e 55 4d  MOAL_FSM_MAX_NUM
9172c160  4d 4f 41 4c 5f 46 53 4d-5f 4d 41 58 5f 4e 55 4d  MOAL_FSM_MAX_NUM
9172c199  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
9172c229  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
[...]
91735a31  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
91735a81  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
91735ad1  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
91735b0d  4d 4f 41 4c 0a 00 00 00-00 00 00 5b 4d 4f 41 4c  MOAL.......[MOAL
91735b19  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
91735b89  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
[...]
9173c33d  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
9173c38d  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
9173c3c2  4d 4f 41 4c 0a 00 5b 4d-4f 41 4c 20 25 73 20 25  MOAL..[MOAL %s %
9173c3c9  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
9173c409  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
[...]
9173c5b9  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
9173c609  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
9173c64e  4d 4f 41 4c 0a 00 6d 6f-61 6c 5f 73 68 69 6d 3a  MOAL..moal_shim:
9173c679  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
[...]
9174dbd1  4d 4f 41 4c 20 25 73 20-25 35 64 20 25 2d 31 38  MOAL %s %5d %-18
9174f04c  4d 4f 41 4c 20 43 61 6e-63 65 6c 20 49 4f 43 54  MOAL Cancel IOCT
917cb705  4d 4f 41 4c 20 49 20 20-32 36 34 31 20 6d 6f 61  MOAL I  2641 moa
917cb7c5  4d 4f 41 4c 20 49 20 20-20 31 32 31 20 6d 6f 61  MOAL I   121 moa
917cb885  4d 4f 41 4c 20 57 20 20-20 31 36 39 20 6d 6f 61  MOAL W   169 moa
917cb945  4d 4f 41 4c 20 49 20 20-33 34 39 36 20 6d 6f 61  MOAL I  3496 moa
[...]
917d2905  4d 4f 41 4c 20 49 20 20-20 37 37 38 20 6d 6f 61  MOAL I   778 moa
917d3985  4d 4f 41 4c 20 49 20 20-20 31 30 39 20 6d 6f 61  MOAL I   109 moa
917d39e7  4d 4f 41 4c 0a 00 00 00-00 00 00 00 00 00 00 00  MOAL............
917d3a45  4d 4f 41 4c 20 49 20 20-32 39 38 32 20 6d 6f 61  MOAL I  2982 moa
917d3b05  4d 4f 41 4c 20 49 20 20-20 37 39 30 20 6d 6f 61  MOAL I   790 moa
[...]
```

```
917d5c05  4d 4f 41 4c 20 4f 20 20-20 36 34 37 20 6d 6f 61  MOAL O   647 moa
917d5f05  4d 4f 41 4c 20 49 20 20-20 31 30 39 20 6d 6f 61  MOAL I   109 moa
917d5f69  4d 4f 41 4c 0a 00 00 00-00 00 00 00 00 00 00 00  MOAL............
917d5fc5  4d 4f 41 4c 20 49 20 20-32 39 38 32 20 6d 6f 61  MOAL I  2982 moa
917d6085  4d 4f 41 4c 20 49 20 20-20 37 39 30 20 6d 6f 61  MOAL I   790 moa
917d6145  4d 4f 41 4c 20 49 20 20-20 39 35 37 20 6d 6f 61  MOAL I   957 moa
917d6205  4d 4f 41 4c 20 49 20 20-20 37 39 36 20 6d 6f 61  MOAL I   796 moa
[...]
917d8305  4d 4f 41 4c 20 49 20 20-20 37 37 38 20 6d 6f 61  MOAL I   778 moa
917d8c05  4d 4f 41 4c 20 49 20 20-20 31 30 39 20 6d 6f 61  MOAL I   109 moa
917d8c66  4d 4f 41 4c 0a 00 00 00-00 00 00 00 00 00 00 00  MOAL............
917d8cc5  4d 4f 41 4c 20 49 20 20-20 32 35 39 20 6d 6f 61  MOAL I   259 moa
917d8d85  4d 4f 41 4c 20 49 20 20-32 39 38 32 20 6d 6f 61  MOAL I  2982 moa
[...]
917e2505  4d 4f 41 4c 20 45 20 20-33 39 30 38 20 6d 6f 61  MOAL E  3908 moa
917e25c5  4d 4f 41 4c 20 4f 20 20-20 36 34 37 20 6d 6f 61  MOAL O   647 moa

0: kd> lmk
start    end        module name
[...]
916ac000 917f1000   mwls97w8arm   (deferred)
[...]

0: kd> lmv m mwls97w8arm
start    end        module name
916ac000 917f1000   mwls97w8arm   (deferred)
    Image path: \SystemRoot\system32\DRIVERS\mwls97w8arm.sys
    Image name: mwls97w8arm.sys
    Timestamp:        Sat Mar 02 09:21:08 2013 (5131C484)
    CheckSum:         0012D61B
    ImageSize:        00145000
    Translations:     0000.04b0 0000.04e4 0409.04b0 0409.04e4

0: kd> !lmi mwls97w8arm
Loaded Module Info: [mwls97w8arm]
         Module: mwls97w8arm
   Base Address: 916ac000
     Image Name: mwls97w8arm.sys
   Machine Type: 452 (ARMNT)
     Time Stamp: 5131c484 Sat Mar 02 09:21:08 2013
           Size: 145000
       CheckSum: 12d61b
Characteristics: 122
Debug Data Dirs: Type  Size     VA  Pointer
             CODEVIEW    86, af9e8,   ae7e8 RSDS - GUID: {9FEEF0F0-D618-4F2A-87B0-6C2398002C94}
               Age: 1, Pdb: e:\MSI-PA-
922\MRVL\BR_WSCM_DEV\EBDC\W8797\p4_DR201303020114\build\w8797\obj\wlan\win8\fre\arm\mwls97w8arm
.pdb
                   ??    10, afa70,   ae870 [Data not mapped]
    Symbol Type: DEFERRED - No error - symbol load deferred
    Load Report: no symbols loaded
```

10. We can also look for ASCII and UNICODE strings in the found driver(s) and on the current thread raw stack (we restrict the output to at least 10 characters long:

```
0: kd> s-[l 10]sa 916ac000 917f1000
916ac04d  "!This program cannot be run in D"
916ac06d  "OS mode."
[...]
9171db6d  "FpGmoal_adapter.cpp"
9171dba0  "[MOAL %s %5d %-18.18s] No priv p"
9171dbc0  "ointer for BSS_INDEX %d"
9171dbe0  "[MOAL %s %5d %-18.18s] Cannot re"
9171dc00  "treive permanent address from ML"
9171dc20  "AN"
9171dc28  "[MOAL %s %5d %-18.18s] Permanent"
9171dc48  " MAC: %02x:%02x:%02x:%02x:%02x:%"
9171dc68  "02x"
9171dc70  "[MOAL %s %5d %-18.18s] open_conf"
9171dc90  "iguration failed"
9171dca8  "[MOAL %s %5d %-18.18s] Registry "
9171dcc8  "MAC: %02x:%02x:%02x:%02x:%02x:%0"
9171dce8  "2x"
9171dcf0  "[MOAL %s %5d %-18.18s] Configure"
9171dd10  "d MAC address is found in the Re"
9171dd30  "gistry. Use it as current addres"
9171dd50  "s"
9171dd58  "[MOAL %s %5d %-18.18s] Cannot se"
9171dd78  "t configured MAC address with ML"
9171dd98  "AN.  Returning mstatus= %s"
[...]
917e2604  "LIST returning with nstatus= NDI"
917e2624  "S_STATUS_PENDING"
917e2684  "Wakeup device..."
917e2744  "cmd_sent=0, data_sent=0"
917e2804  "RX: port=0 rx_len=10"
917e2984  "card_2_host_mp_aggr: No aggr for"
917e29a4  " control port"
917e2a44  "RX: f_do_rx_cur: port: 0 rx_len:"
917e2a64  " 256"
917e2b04  "--- Rx: Event [10] ---"

0: kd> s-[l 10]su 916ac000 917f1000
[...]
917e6388  "WmiQueryTraceInformation"
917e63bc  "EtwRegisterClassicProvider"

0: kd> !thread
THREAD 83637600  Cid 0000.0000  Teb: 00000000 Win32Thread: 00000000 RUNNING on processor 0
Not impersonating
DeviceMap                 84009120
Owning Process            8362f200       Image:         Idle
Attached Process          8685f100       Image:         System
Wait Start TickCount      74044          Ticks: 1 (0:00:00:00.015)
Context Switch Count      90718          IdealProcessor: 0
UserTime                  00:00:00.000
KernelTime                00:17:09.906
Win32 Start Address nt!KiIdleLoop (0x83426161)
Stack Init 84774ed8 Current 84774ec0 Base 84775000 Limit 84772000 Call 0
Priority 0 BasePriority 0 UnusualBoost 0 ForegroundBoost 0 IoPriority 0 PagePriority 5
[...]
```

```
0: kd> s-[l 5]sa 84772000 84775000
847741b8  "ErToErHw"
847746d9  "3=i/B"
84774779  "3=i/B"
84774ad4  "usbpusbp"
84774c18  "HFDOH"
84774e99  "3=i/B"
84774f31  "3=i/B"
84774fd1  "3=i/B"
```

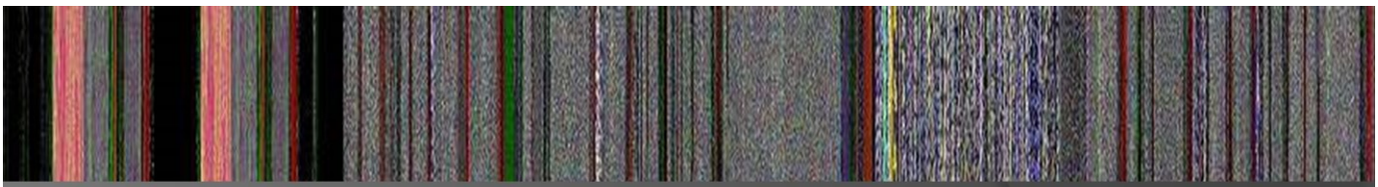
Note: We also check for any pointers on raw stack that might point to ASCII or UNICODE values:

```
0: kd> dpa 84772000 84775000
84772000  00000000
84772004  00000000
[...]
84774198  00000200
8477419c  8fef0a94 "wp  %p DataCommandWorker COMPLETE, Info=0x%x"
847741a0  8dc10d8a
[...]
84774588  00000000
8477458c  8836d000 "FDMPDUMP"
84774590  00000000
[...]
84774ff8  8b242bcc
84774ffc  181359b4
84775000  ????????
```

11. We close logging before exiting WinDbg:

```
0: kd> .logclose
Closing open log file C:\AdvWMDA-Dumps\ARM\Complete\C2-ARM.log
```

Note: To avoid possible confusion and glitches we recommend exiting WinDbg after each exercise.



Windows Debugging³ Accelerated

Dmitry Vostokov
Software Diagnostics Services

Published by OpenTask, Republic of Ireland

OpenTask books and magazines are available through booksellers and distributors worldwide. For further information or comments send requests to press@opentask.com.

A CIP catalogue record for this book is available from the British Library.

ISBN-l3: 978-1-908043-56-6 (Paperback)

Contents

Exercise D1

Goal: Learn how code generation parameters can influence process execution behavior.

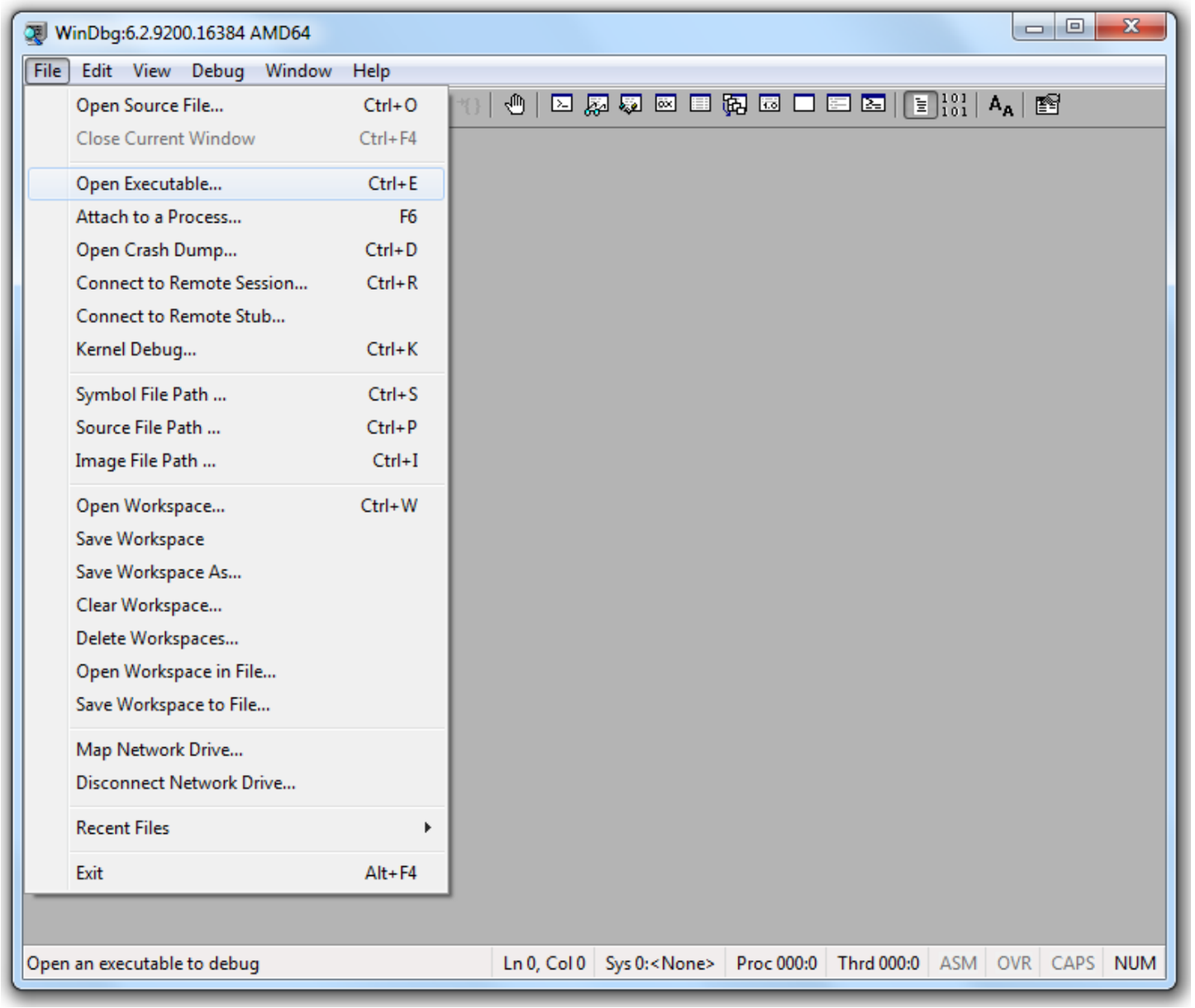
Elementary Diagnostics Patterns: Crash

Memory Analysis Patterns: Exception Stack Trace

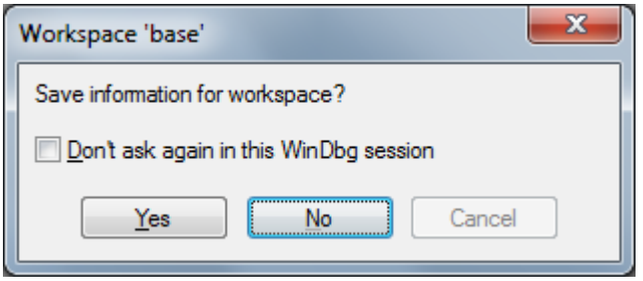
Debugging Implementation Patterns: Scope, Variable Value, Type Structure, Code Breakpoint

1. Launch WinDbg from *Windows Kits \ Debugging Tools for Windows (X64)*

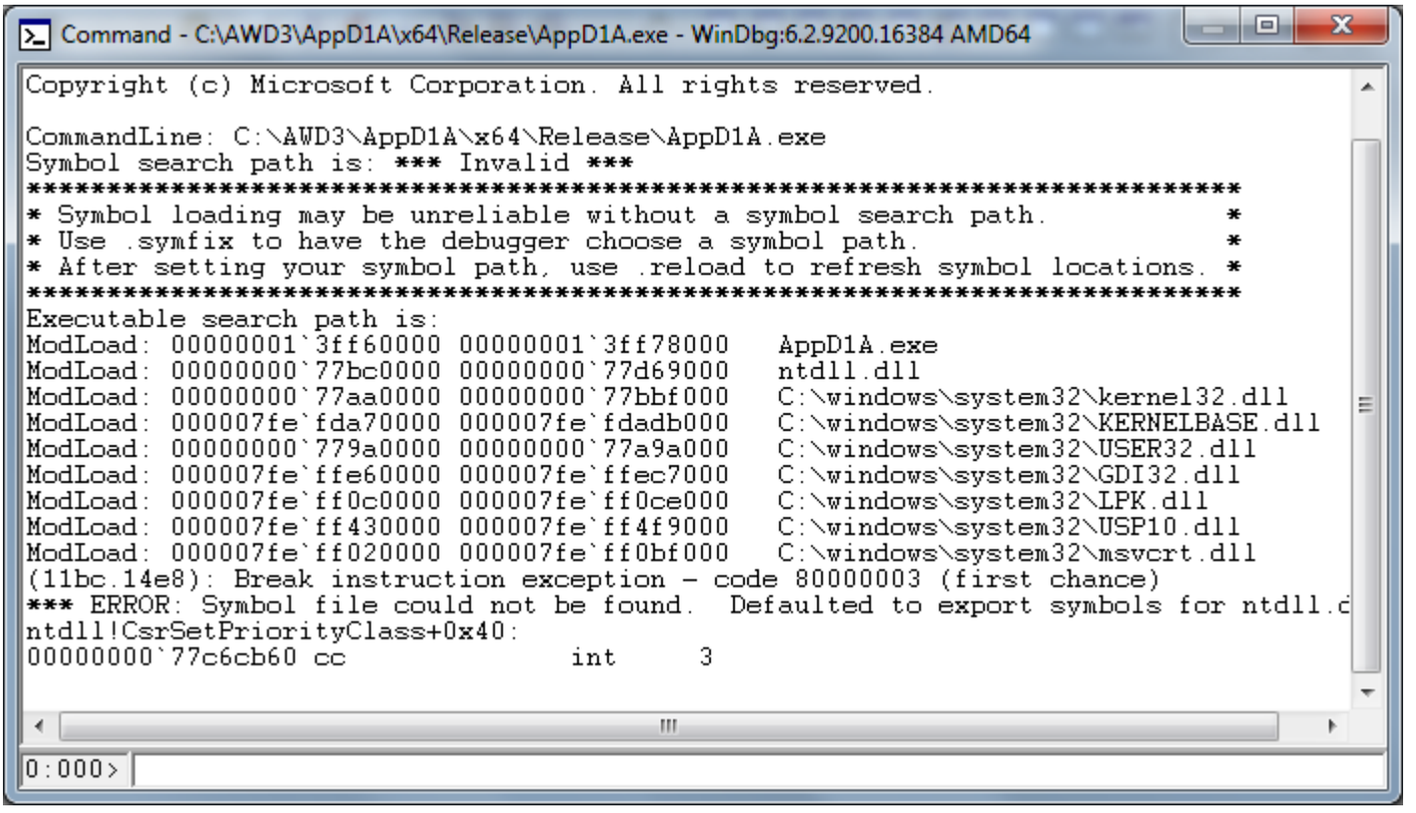
2. Open *\AWD3\AppD1A\x64\Release\AppD1A.exe* executable:



3. If you are presented with this dialog choose *No*:



4. You get the executable file loaded and ready for a debugging session:



5. Open a log file:

```
0:000> .logopen C:\AWD3\D1A.log
Opened log file 'C:\AWD3\D1A.log'
```

6. Set up a link to Microsoft symbol server and reload symbols:

```
0:000> .symfix c:\mss

0:000> .reload
.....................
```

7. **lm** command lists module information:

```
0:000> lm
start             end                 module name
00000000`779a0000 00000000`77a9a000   USER32     (deferred)
00000000`77aa0000 00000000`77bbf000   kernel32   (deferred)
00000000`77bc0000 00000000`77d69000   ntdll      (export symbols)
C:\windows\SYSTEM32\ntdll.dll
00000001`3ff60000 00000001`3ff78000   AppD1A     (deferred)
000007fe`fda70000 000007fe`fdadb000   KERNELBASE   (deferred)
000007fe`ff020000 000007fe`ff0bf000   msvcrt     (deferred)
000007fe`ff0c0000 000007fe`ff0ce000   LPK        (deferred)
000007fe`ff430000 000007fe`ff4f9000   USP10      (deferred)
000007fe`ffe60000 000007fe`ffec7000   GDI32      (deferred)
```

8. We continue process execution using **g** command and ignore any first chance exceptions until we come to a second chance exception:

```
0:000> g
ModLoad: 000007fe`ff340000 000007fe`ff36e000   C:\windows\system32\IMM32.DLL
ModLoad: 000007fe`ff230000 000007fe`ff339000   C:\windows\system32\MSCTF.dll
(3008.3338): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
ntdll!RtlImageNtHeaderEx+0x58:
00000000`77c17f18 66390a          cmp     word ptr [rdx],cx ds:27910f7f`00000000=????

0:000> g
(3008.3338): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
USER32!InitClsMenuNameW+0x3e:
00000000`779c3c69 66f2af          repne scas word ptr [rdi]

0:000> g
(3008.3338): Access violation - code c0000005 (!!! second chance !!!)
USER32!InitClsMenuNameW+0x3e:
00000000`779c3c69 66f2af          repne scas word ptr [rdi]
```

9. We see that a crash happened in USER32 module with the following CPU state:

```
0:000> r
rax=0000000000000000 rbx=000000000031f748 rcx=ffffffffffffffff
rdx=3f62260000000000 rsi=3f62260000000000 rdi=3f62260000000000
rip=00000000779c3c69 rsp=000000000031f660 rbp=000000000031f730
 r8=000000000031f748  r9=000000000031f688 r10=00000000c000007b
r11=0000000000000000 r12=3f62260000000000 r13=0000000000000000
r14=0000000000000000 r15=0000000000000000
iopl=0         nv up ei pl zr na po nc
cs=0033  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00010244
USER32!InitClsMenuNameW+0x3e:
00000000`779c3c69 66f2af          repne scas word ptr [rdi]
```

10. The default analysis command also gives us a source code:

```
0:000> !analyze -v
*******************************************************************************
*                                                                             *
*                        Exception Analysis                                   *
*                                                                             *
*******************************************************************************


FAULTING_IP:
USER32!InitClsMenuNameW+3e
00000000`779c3c69 66f2af          repne scas word ptr [rdi]

EXCEPTION_RECORD:  ffffffffffffffff -- (.exr 0xffffffffffffffff)
ExceptionAddress: 00000000779c3c69 (USER32!InitClsMenuNameW+0x000000000000003e)
   ExceptionCode: c0000005 (Access violation)
  ExceptionFlags: 00000000
NumberParameters: 2
   Parameter[0]: 0000000000000000
   Parameter[1]: ffffffffffffffff
Attempt to read from address ffffffffffffffff

FAULTING_THREAD:  0000000000003338

DEFAULT_BUCKET_ID:  INVALID_POINTER_READ

PROCESS_NAME:  AppD1A.exe

ERROR_CODE: (NTSTATUS) 0xc0000005 - The instruction at 0x%08lx referenced memory at 0x%08lx.
The memory could not be %s.

EXCEPTION_CODE: (NTSTATUS) 0xc0000005 - The instruction at 0x%08lx referenced memory at
0x%08lx. The memory could not be %s.

EXCEPTION_PARAMETER1:  0000000000000000

EXCEPTION_PARAMETER2:  ffffffffffffffff

READ_ADDRESS:  ffffffffffffffff

FOLLOWUP_IP:
AppD1A!MyRegisterClass+8d [c:\awd3\appd1a\appd1a\appd1a.cpp @ 84]
00000001`3f61116d 4883c478        add     rsp,78h

NTGLOBALFLAG:  470

APPLICATION_VERIFIER_FLAGS:  0

APP:  appd1a.exe

PRIMARY_PROBLEM_CLASS:  INVALID_POINTER_READ

BUGCHECK_STR:  APPLICATION_FAULT_INVALID_POINTER_READ

LAST_CONTROL_TRANSFER:  from 00000000779aff55 to 00000000779c3c69
```

```
STACK_TEXT:
00000000`0031f660 00000000`779aff55 : 00000000`0031fa30 00000000`00000080 00000000`00000000
00000000`0000030a : USER32!InitClsMenuNameW+0x3e
00000000`0031f6b0 00000000`779b0913 : 0000765c`619b382b 00000000`0031fab0 00000000`00000000
00000000`00000000 : USER32!RegisterClassExWOWW+0x151
00000000`0031fa00 00000001`3f61116d : 00000000`00000000 00000000`00000006 00000001`3f626bf8
00000000`00000000 : USER32!RegisterClassW+0x53
00000000`0031fa90 00000001`3f61105c : 00000001`3f610000 00000000`0000000a 00000001`3f610000
00000001`3f612a36 : AppD1A!MyRegisterClass+0x8d
00000000`0031fb10 00000001`3f6115c8 : 00000001`3f610000 00000000`00000000 00000000`000835d6
00000001`0000000a : AppD1A!wWinMain+0x5c
00000000`0031fb70 00000000`77ab652d : 00000000`00000000 00000000`00000000 00000000`00000000
00000000`00000000 : AppD1A!__tmainCRTStartup+0x148
00000000`0031fbb0 00000000`77bec521 : 00000000`00000000 00000000`00000000 00000000`00000000
00000000`00000000 : kernel32!BaseThreadInitThunk+0xd
00000000`0031fbe0 00000000`00000000 : 00000000`00000000 00000000`00000000 00000000`00000000
00000000`00000000 : ntdll!RtlUserThreadStart+0x1d


FAULTING_SOURCE_LINE:  c:\awd3\appd1a\appd1a\appd1a.cpp

FAULTING_SOURCE_FILE:  c:\awd3\appd1a\appd1a\appd1a.cpp

FAULTING_SOURCE_LINE_NUMBER:  84

FAULTING_SOURCE_CODE:
    80:      wc.lpszMenuName     = MAKEINTRESOURCE(IDC_APPD1A);
    81:      wc.lpszClassName    = szWindowClass;
    82:
    83:      return RegisterClass(&wc);
>   84: }
    85:
    86: //
    87: //   FUNCTION: InitInstance(HINSTANCE, int)
    88: //
    89: //   PURPOSE: Saves instance handle and creates main window


SYMBOL_STACK_INDEX:  3

SYMBOL_NAME:  appd1a!MyRegisterClass+8d

FOLLOWUP_NAME:  MachineOwner

MODULE_NAME: AppD1A

IMAGE_NAME:  AppD1A.exe

DEBUG_FLR_IMAGE_TIMESTAMP:  51d188c3

STACK_COMMAND:  dt ntdll!LdrpLastDllInitializer BaseDllName ; dt ntdll!LdrpFailureData ; ~0s ;
kb

FAILURE_BUCKET_ID:  INVALID_POINTER_READ_c0000005_AppD1A.exe!MyRegisterClass

BUCKET_ID:  X64_APPLICATION_FAULT_INVALID_POINTER_READ_appd1a!MyRegisterClass+8d

Followup: MachineOwner
---------
```

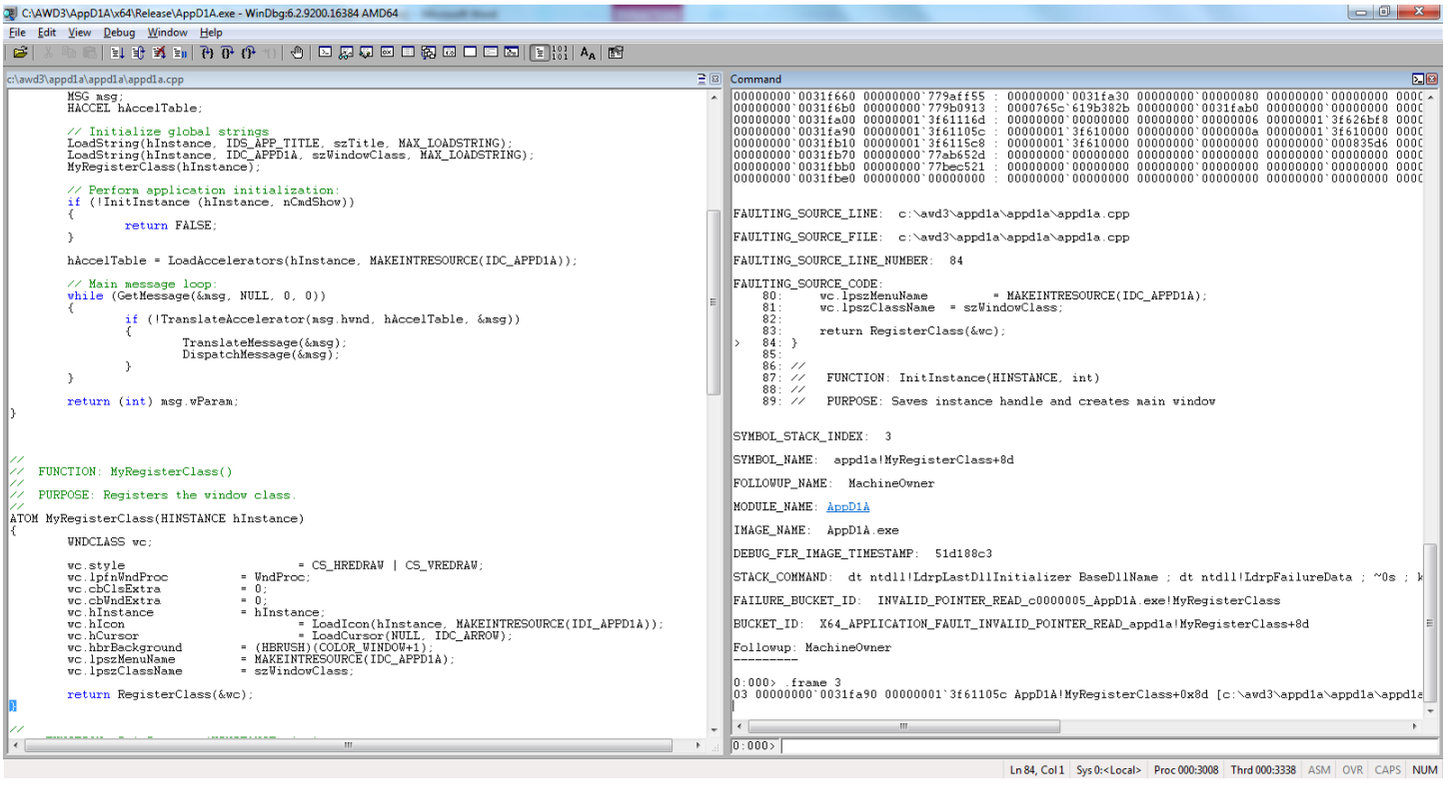
11. We get a stack trace with frame numbers using **kn** command:

```
0:000> kn
 # Child-SP          RetAddr           Call Site
00 00000000`0031f660 00000000`779aff55 USER32!InitClsMenuNameW+0x3e
01 00000000`0031f6b0 00000000`779b0913 USER32!RegisterClassExWOWW+0x151
02 00000000`0031fa00 00000001`3f61116d USER32!RegisterClassW+0x53
03 00000000`0031fa90 00000001`3f61105c AppD1A!MyRegisterClass+0x8d
[c:\awd3\appd1a\appd1a\appd1a.cpp @ 84]
04 00000000`0031fb10 00000001`3f6115c8 AppD1A!wWinMain+0x5c [c:\awd3\appd1a\appd1a\appd1a.cpp @
41]
05 00000000`0031fb70 00000000`77ab652d AppD1A!__tmainCRTStartup+0x148
[f:\dd\vctools\crt_bld\self_64_amd64\crt\src\crt0.c @ 238]
06 00000000`0031fbb0 00000000`77bec521 kernel32!BaseThreadInitThunk+0xd
07 00000000`0031fbe0 00000000`00000000 ntdll!RtlUserThreadStart+0x1d
```

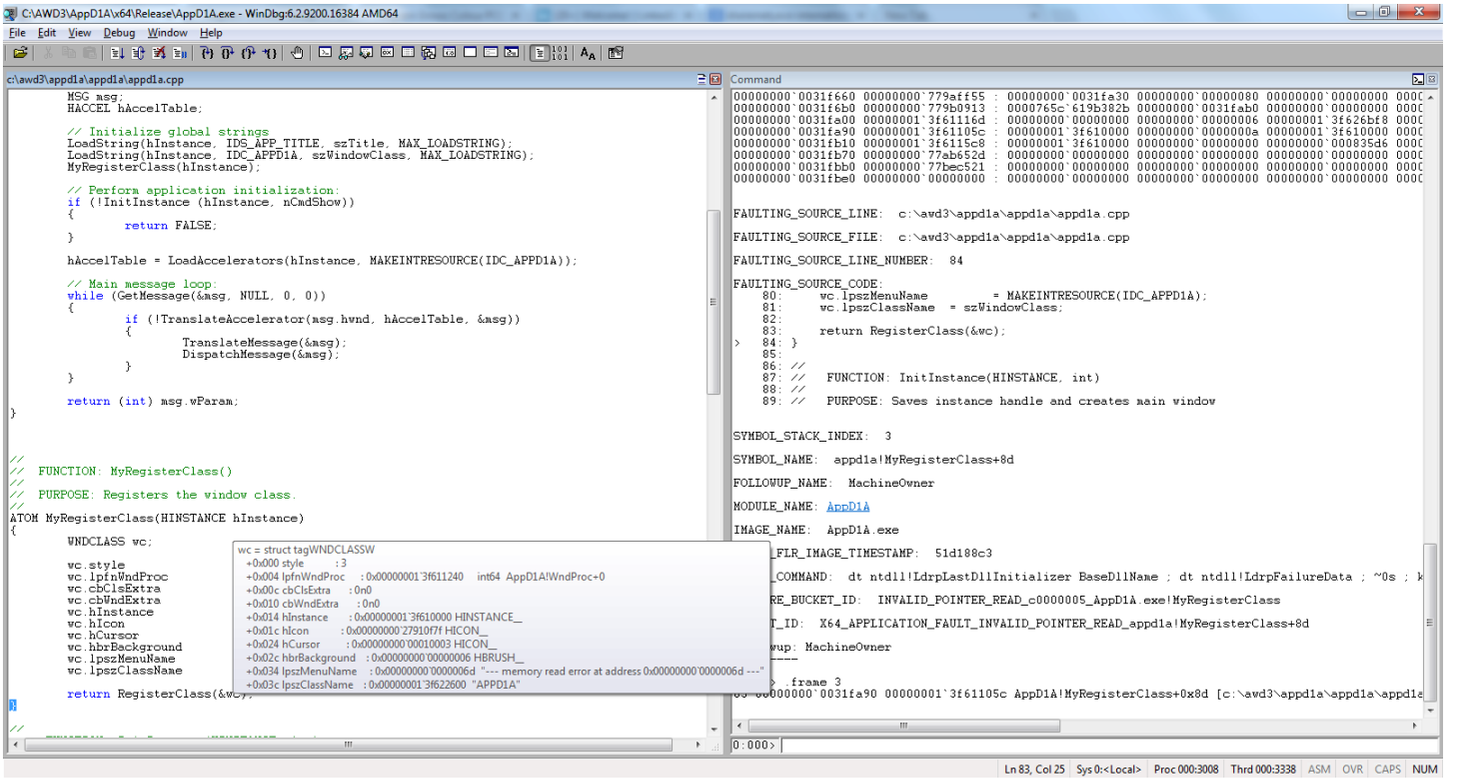
12. Now we can set the frame we want to investigate (from where *RegisterClassW* was called):

```
0:000> .frame 3
03 00000000`0031fa90 00000001`3f61105c AppD1A!MyRegisterClass+0x8d
[c:\awd3\appd1a\appd1a\appd1a.cpp @ 84]
```

Note: you see immediately a source code window to the left of the command window:



13. If we select the source code window and hover a mouse pointer over **wc** variable we get that structure variables:



We can also dump this variable using type information:

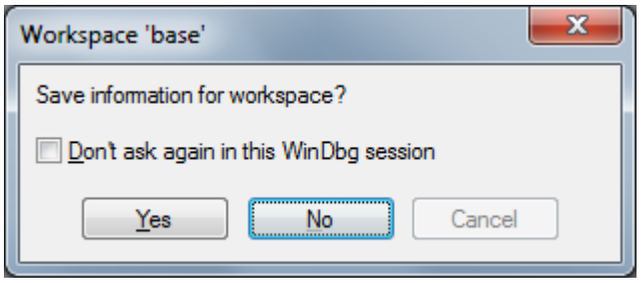
```
0:000> dt wc
Local var @ 0x31fab0 Type tagWNDCLASSW
   +0x000 style            : 3
   +0x004 lpfnWndProc      : 0x00000001`3f611240     int64  AppD1A!WndProc+0
   +0x00c cbClsExtra       : 0n0
   +0x010 cbWndExtra       : 0n0
   +0x014 hInstance        : 0x00000001`3f610000 HINSTANCE__
   +0x01c hIcon            : 0x00000000`27910f7f HICON__
   +0x024 hCursor          : 0x00000000`00010003 HICON__
   +0x02c hbrBackground    : 0x00000000`00000006 HBRUSH__
   +0x034 lpszMenuName     : 0x00000000`0000006d  "--- memory read error at address
0x00000000`0000006d ---"
   +0x03c lpszClassName    : 0x00000001`3f622600  "APPD1A"
```

14. We can also list all other local variables and parameters for the current frame:

```
0:000> dv /i /V
prv param  00000000`0031fb10 @rsp+0x0080            hInstance = 0x00000001`3f610000
prv local  00000000`0031fab0 @rsp+0x0020                  wc = struct tagWNDCLASSW
```

Note: Since all structure members seems to be valid let's compare it with another application that doesn't crash.

15. Launch another instance of WinDbg from *Windows Kits* \ *Debugging Tools for Windows (X64)* and open *\AWD3\AppD1B\x64\Release\AppD1B.exe* executable. If you are presented with this dialog choose *No*:



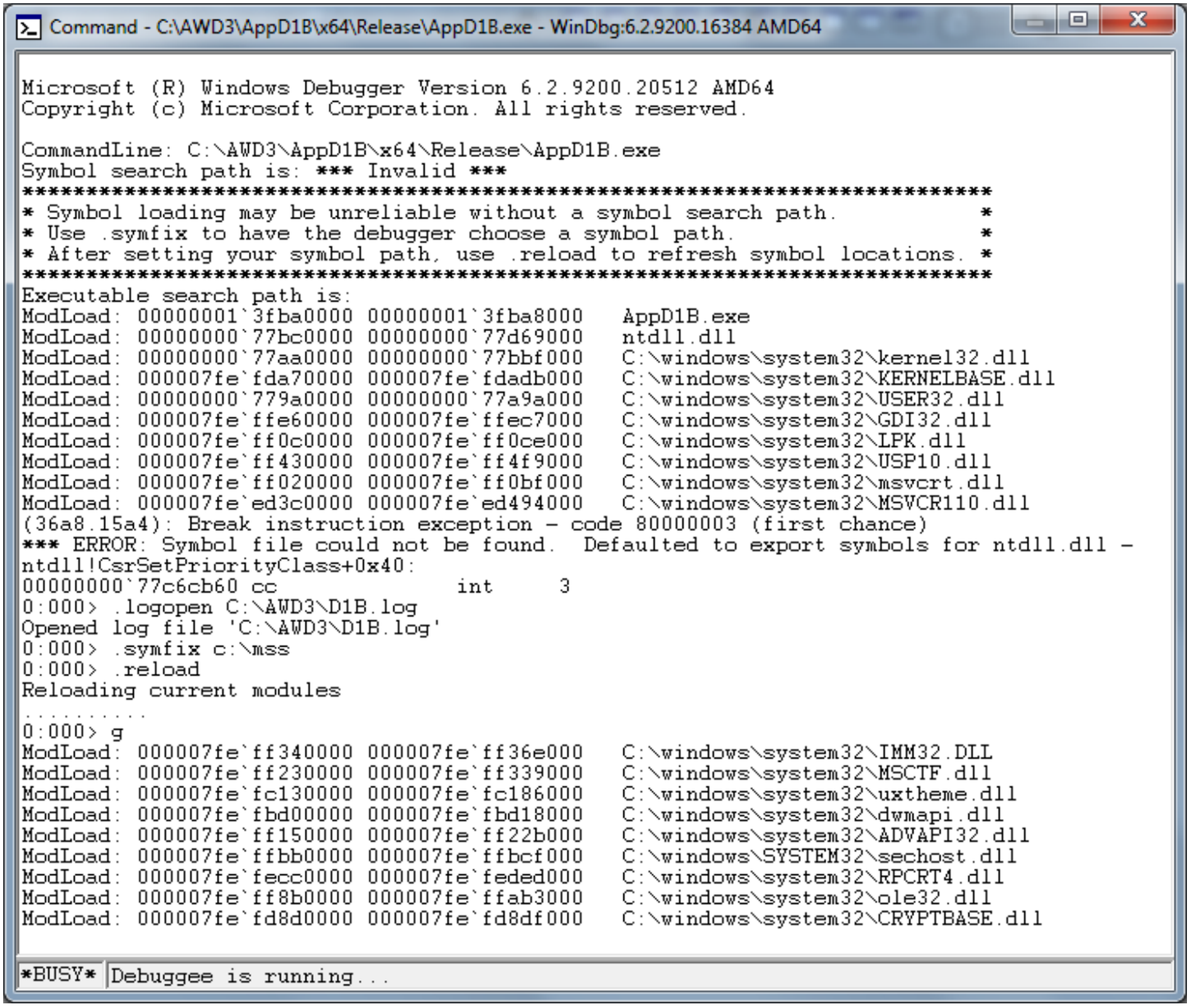
We get the following output:

```
Microsoft (R) Windows Debugger Version 6.2.9200.20512 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.

CommandLine: C:\AWD3\AppD1B\x64\Release\AppD1B.exe
Symbol search path is: *** Invalid ***
************************************************************************
* Symbol loading may be unreliable without a symbol search path.       *
* Use .symfix to have the debugger choose a symbol path.               *
* After setting your symbol path, use .reload to refresh symbol locations. *
************************************************************************
Executable search path is:
ModLoad: 00000001`3fba0000 00000001`3fba8000   AppD1B.exe
ModLoad: 00000000`77bc0000 00000000`77d69000   ntdll.dll
ModLoad: 00000000`77aa0000 00000000`77bbf000   C:\windows\system32\kernel32.dll
ModLoad: 000007fe`fda70000 000007fe`fdadb000   C:\windows\system32\KERNELBASE.dll
ModLoad: 00000000`779a0000 00000000`77a9a000   C:\windows\system32\USER32.dll
ModLoad: 000007fe`ffe60000 000007fe`ffec7000   C:\windows\system32\GDI32.dll
ModLoad: 000007fe`ff0c0000 000007fe`ff0ce000   C:\windows\system32\LPK.dll
ModLoad: 000007fe`ff430000 000007fe`ff4f9000   C:\windows\system32\USP10.dll
ModLoad: 000007fe`ff020000 000007fe`ff0bf000   C:\windows\system32\msvcrt.dll
ModLoad: 000007fe`ed3c0000 000007fe`ed494000   C:\windows\system32\MSVCR110.dll
(36a8.15a4): Break instruction exception - code 80000003 (first chance)
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for ntdll.dll -
ntdll!CsrSetPriorityClass+0x40:
00000000`77c6cb60 cc              int     3
```

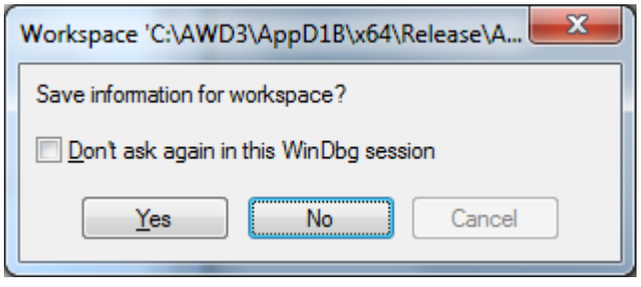
16. We open a new log file, fix and reload symbols:

```
0:000> .logopen C:\AWD3\D1B.log
Opened log file 'C:\AWD3\D1B.log'

0:000> .symfix c:\mss

0:000> .reload
Reloading current modules
..........
```

17. If we run it via **g** command we don't get any exceptions:



18. So we choose *Debug* \ *Break* menu option and then *Debug* \ *Restart*. If presented with this dialog choose *No*:



We get the following output:

```
0:000> g
ModLoad: 000007fe`ff340000 000007fe`ff36e000   C:\windows\system32\IMM32.DLL
ModLoad: 000007fe`ff230000 000007fe`ff339000   C:\windows\system32\MSCTF.dll
ModLoad: 000007fe`fc130000 000007fe`fc186000   C:\windows\system32\uxtheme.dll
ModLoad: 000007fe`fbd00000 000007fe`fbd18000   C:\windows\system32\dwmapi.dll
ModLoad: 000007fe`ff150000 000007fe`ff22b000   C:\windows\system32\ADVAPI32.dll
ModLoad: 000007fe`ffbb0000 000007fe`ffbcf000   C:\windows\SYSTEM32\sechost.dll
ModLoad: 000007fe`fecc0000 000007fe`feded000   C:\windows\system32\RPCRT4.dll
ModLoad: 000007fe`ff8b0000 000007fe`ffab3000   C:\windows\system32\ole32.dll
ModLoad: 000007fe`fd8d0000 000007fe`fd8df000   C:\windows\system32\CRYPTBASE.dll
(36a8.3314): Break instruction exception - code 80000003 (first chance)
ntdll!DbgBreakPoint:
00000000`77c10530 cc              int     3

0:001> .restart /f
CommandLine: C:\AWD3\AppD1B\x64\Release\AppD1B.exe
Symbol search path is: srv*
Executable search path is: srv*
ModLoad: 00000001`3f990000 00000001`3f998000   AppD1B.exe
ModLoad: 00000000`77bc0000 00000000`77d69000   ntdll.dll
ModLoad: 00000000`77aa0000 00000000`77bbf000   C:\windows\system32\kernel32.dll
ModLoad: 000007fe`fda70000 000007fe`fdadb000   C:\windows\system32\KERNELBASE.dll
ModLoad: 00000000`779a0000 00000000`77a9a000   C:\windows\system32\USER32.dll
ModLoad: 000007fe`ffe60000 000007fe`ffec7000   C:\windows\system32\GDI32.dll
ModLoad: 000007fe`ff0c0000 000007fe`ff0ce000   C:\windows\system32\LPK.dll
ModLoad: 000007fe`ff430000 000007fe`ff4f9000   C:\windows\system32\USP10.dll
ModLoad: 000007fe`ff020000 000007fe`ff0bf000   C:\windows\system32\msvcrt.dll
ModLoad: 000007fe`ed3c0000 000007fe`ed494000   C:\windows\system32\MSVCR110.dll
(2cc8.3048): Break instruction exception - code 80000003 (first chance)
ntdll!LdrpDoDebuggerBreak+0x30:
00000000`77c6cb60 cc              int     3
```

19. Since we want to compare the same behavior of *RegisterClassW* function we need to put a breakpoint to break in when this function is about to be executed. Then we would see *WNDCLASS* structure passed to it. We set a pattern matching breakpoint using **bm** command:

```
0:000> bm *!RegisterClassW
  1: 00000000`779b08c0 @!"USER32!RegisterClassW"
*** WARNING: Unable to verify checksum for AppD1B.exe
```

20. Indeed we a hit immediately:

```
0:000> g
ModLoad: 000007fe`ff340000 000007fe`ff36e000   C:\windows\system32\IMM32.DLL
ModLoad: 000007fe`ff230000 000007fe`ff339000   C:\windows\system32\MSCTF.dll
Breakpoint 1 hit
USER32!RegisterClassW:
00000000`779b08c0 fff3            push    rbx
```

We get an almost identical stack trace prior to *RegisterClassW* when we compare with the previously running instance of AppD1A.exe:

```
0:000> k ; AppD1B
Child-SP          RetAddr           Call Site
00000000`0023f788 00000001`3f99116d USER32!RegisterClassW
00000000`0023f790 00000001`3f99105c AppD1B!MyRegisterClass+0x8d
[c:\awd3\appd1b\appd1b\appd1b.cpp @ 84]
00000000`0023f810 00000001`3f991712 AppD1B!wWinMain+0x5c [c:\awd3\appd1b\appd1b\appd1b.cpp @
41]
00000000`0023f880 00000000`77ab652d AppD1B!__tmainCRTStartup+0x15e
[f:\dd\vctools\crt_bld\self_64_amd64\crt\src\crtexe.c @ 528]
00000000`0023f8c0 00000000`77bec521 kernel32!BaseThreadInitThunk+0xd
00000000`0023f8f0 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

0:000> k ; AppD1A
Child-SP          RetAddr           Call Site
00000000`0031f660 00000000`779aff55 USER32!InitClsMenuNameW+0x3e
00000000`0031f6b0 00000000`779b0913 USER32!RegisterClassExWOWW+0x151
00000000`0031fa00 00000001`3f61116d USER32!RegisterClassW+0x53
00000000`0031fa90 00000001`3f61105c AppD1A!MyRegisterClass+0x8d
[c:\awd3\appd1a\appd1a\appd1a.cpp @ 84]
00000000`0031fb10 00000001`3f6115c8 AppD1A!wWinMain+0x5c [c:\awd3\appd1a\appd1a\appd1a.cpp @
41]
00000000`0031fb70 00000000`77ab652d AppD1A!__tmainCRTStartup+0x148
[f:\dd\vctools\crt_bld\self_64_amd64\crt\src\crt0.c @ 238]
00000000`0031fbb0 00000000`77bec521 kernel32!BaseThreadInitThunk+0xd
00000000`0031fbe0 00000000`00000000 ntdll!RtlUserThreadStart+0x1d
```

21. We choose frame 1 which called *RegisterClassW* and immediately get access to wc variable (we also note that function *MyRegisterClass* source code is identical to AppD1A):

```
0:000> kn
 # Child-SP          RetAddr           Call Site
00 00000000`0023f700 00000001`3f99116d USER32!RegisterClassW+0x1b
01 00000000`0023f790 00000001`3f99105c AppD1B!MyRegisterClass+0x8d
[c:\awd3\appd1b\appd1b\appd1b.cpp @ 84]
02 00000000`0023f810 00000001`3f991712 AppD1B!wWinMain+0x5c [c:\awd3\appd1b\appd1b\appd1b.cpp @
41]
03 00000000`0023f880 00000000`77ab652d AppD1B!__tmainCRTStartup+0x15e
[f:\dd\vctools\crt_bld\self_64_amd64\crt\src\crtexe.c @ 528]
04 00000000`0023f8c0 00000000`77bec521 kernel32!BaseThreadInitThunk+0xd
05 00000000`0023f8f0 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

0:000> .frame 1
01 00000000`0023f790 00000001`3f99105c AppD1B!MyRegisterClass+0x8d
[c:\awd3\appd1b\appd1b\appd1b.cpp @ 84]

0:000> dt wc ; AppD1B
Local var @ 0x23f7b0 Type tagWNDCLASSW
   +0x000 style            : 3
   +0x008 lpfnWndProc      : 0x00000001`3f991240     int64  AppD1B!WndProc+0
   +0x010 cbClsExtra       : 0n0
   +0x014 cbWndExtra       : 0n0
   +0x018 hInstance        : 0x00000001`3f990000 HINSTANCE__
   +0x020 hIcon            : 0xffffffff`b90303e9 HICON__
   +0x028 hCursor          : 0x00000000`00010003 HICON__
   +0x030 hbrBackground    : 0x00000000`00000006 HBRUSH__
   +0x038 lpszMenuName     : 0x00000000`0000006d  "--- memory read error at address
0x00000000`0000006d ---"
   +0x040 lpszClassName    : 0x00000001`3f9935e0  "APPD1B"
```

22. But if we look at AppD1A structure variant we see its members have different offsets:

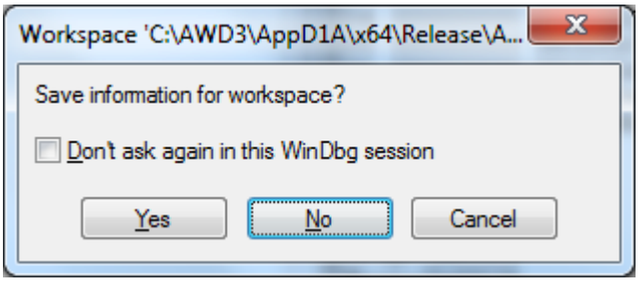
```
0:000> dt wc ; AppD1A
Local var @ 0x31fab0 Type tagWNDCLASSW
   +0x000 style            : 3
   +0x004 lpfnWndProc      : 0x00000001`3f611240     int64  AppD1A!WndProc+0
   +0x00c cbClsExtra       : 0n0
   +0x010 cbWndExtra       : 0n0
   +0x014 hInstance        : 0x00000001`3f610000 HINSTANCE__
   +0x01c hIcon            : 0x00000000`27910f7f HICON__
   +0x024 hCursor          : 0x00000000`00010003 HICON__
   +0x02c hbrBackground    : 0x00000000`00000006 HBRUSH__
   +0x034 lpszMenuName     : 0x00000000`0000006d  "--- memory read error at address
0x00000000`0000006d ---"
   +0x03c lpszClassName    : 0x00000001`3f622600  "APPD1A"
```

23. We close logs in both WinDbg instances:

```
0:000> .logclose
Closing open log file C:\AWD3\D1A.log
```

24. To avoid possible confusion and glitches we recommend exiting WinDbg after each exercise.

If you are presented with this dialog choose No:



25. The problem was partially fixed without changing alignment by using a different bigger structure *WNDCLASSEX* and *RegisterClassExW* Win32 API function. We can open AppD1C.exe in another WinDbg instance:

```
Microsoft (R) Windows Debugger Version 6.2.9200.20512 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.

CommandLine: C:\AWD3\AppD1C\x64\Release\AppD1C.exe
Symbol search path is: *** Invalid ***
************************************************************************
* Symbol loading may be unreliable without a symbol search path.       *
* Use .symfix to have the debugger choose a symbol path.               *
* After setting your symbol path, use .reload to refresh symbol locations. *
************************************************************************
Executable search path is:
ModLoad: 00000001`3ff60000 00000001`3ff68000   AppD1C.exe
ModLoad: 00000000`77bc0000 00000000`77d69000   ntdll.dll
ModLoad: 00000000`77aa0000 00000000`77bbf000   C:\windows\system32\kernel32.dll
ModLoad: 000007fe`fda70000 000007fe`fdadb000   C:\windows\system32\KERNELBASE.dll
ModLoad: 00000000`779a0000 00000000`77a9a000   C:\windows\system32\USER32.dll
ModLoad: 000007fe`ffe60000 000007fe`ffec7000   C:\windows\system32\GDI32.dll
ModLoad: 000007fe`ff0c0000 000007fe`ff0ce000   C:\windows\system32\LPK.dll
ModLoad: 000007fe`ff430000 000007fe`ff4f9000   C:\windows\system32\USP10.dll
ModLoad: 000007fe`ff020000 000007fe`ff0bf000   C:\windows\system32\msvcrt.dll
```

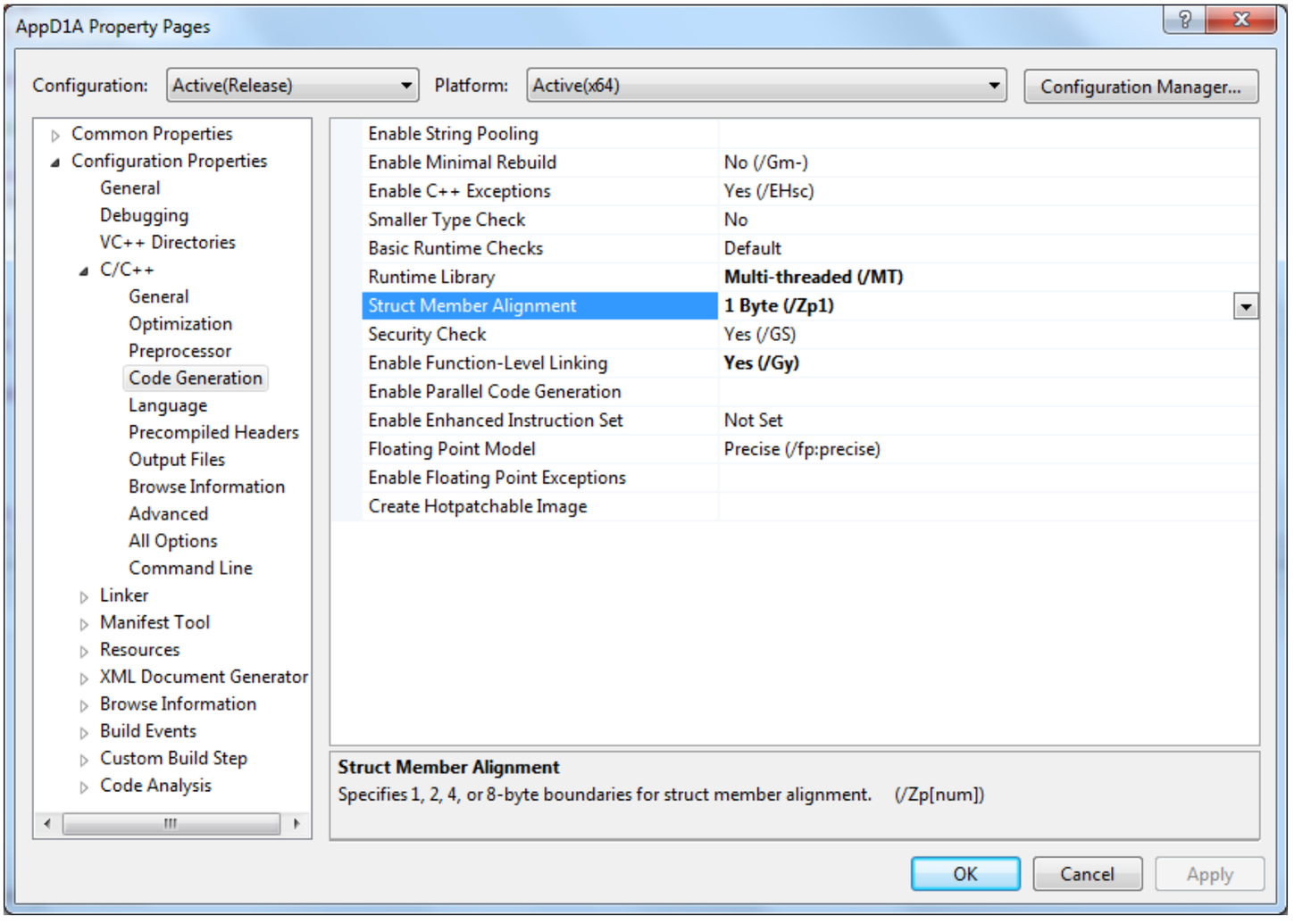
```
ModLoad: 000007fe`ed3c0000 000007fe`ed494000   C:\windows\system32\MSVCR110.dll
(1ad4.269c): Break instruction exception - code 80000003 (first chance)
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for ntdll.dll -
ntdll!CsrSetPriorityClass+0x40:
00000000`77c6cb60 cc              int     3

0:000> .symfix c:\mss

0:000> .reload
Reloading current modules
..........

0:000> bm *!RegisterClassExW
  1: 00000000`779b0e9c @!"USER32!RegisterClassExW"
*** WARNING: Unable to verify checksum for AppD1C.exe

0:000> g
ModLoad: 000007fe`ff340000 000007fe`ff36e000   C:\windows\system32\IMM32.DLL
ModLoad: 000007fe`ff230000 000007fe`ff339000   C:\windows\system32\MSCTF.dll
Breakpoint 1 hit
USER32!RegisterClassExW:
00000000`779b0e9c 4883ec38        sub     rsp,38h

0:000> kn
 # Child-SP          RetAddr           Call Site
00 00000000`002afb48 00000001`3ff6118a USER32!RegisterClassExW
01 00000000`002afb50 00000001`3ff6105c AppD1C!MyRegisterClass+0xaa
[c:\awd3\appd1c\appd1c\appd1c.cpp @ 84]
02 00000000`002afbd0 00000001`3ff61722 AppD1C!wWinMain+0x5c [c:\awd3\appd1c\appd1c\appd1c.cpp @
38]
03 00000000`002afc30 00000000`77ab652d AppD1C!__tmainCRTStartup+0x15e
[f:\dd\vctools\crt_bld\self_64_amd64\crt\src\crtexe.c @ 528]
04 00000000`002afc70 00000000`77bec521 kernel32!BaseThreadInitThunk+0xd
05 00000000`002afca0 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

0:000> .frame 1
01 00000000`002afb50 00000001`3ff6105c AppD1C!MyRegisterClass+0xaa
[c:\awd3\appd1c\appd1c\appd1c.cpp @ 84]

0:000> dv /i /V
prv param  00000000`002afbd0 @rsp+0x0080             hInstance = 0x00000001`3ff60000
prv local  00000000`002afb70 @rsp+0x0020                  wcex = struct tagWNDCLASSEXW
```

Note: adding a new extra member in the new structure shifts the remaining members and set the same layout as in AppD1B:

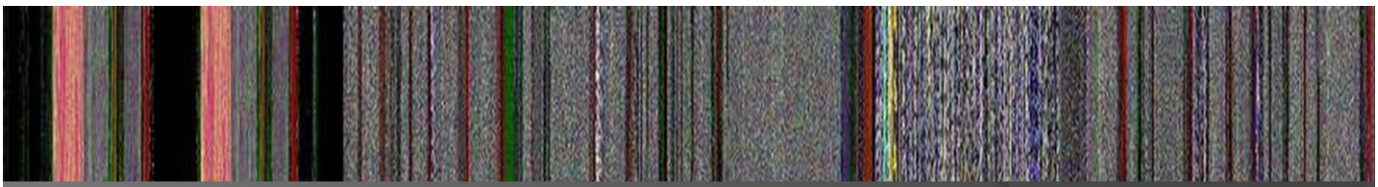
```
0:000> dt wcex ; AppD1C
Local var @ 0x2afb70 Type tagWNDCLASSEXW
   +0x000 cbSize           : 0x50
   +0x004 style            : 3
   +0x008 lpfnWndProc      : 0x00000001`3ff61250     int64  AppD1C!WndProc+0
   +0x010 cbClsExtra       : 0n0
   +0x014 cbWndExtra       : 0n0
   +0x018 hInstance        : 0x00000001`3ff60000 HINSTANCE__
   +0x020 hIcon            : 0x00000000`09eb137b HICON__
   +0x028 hCursor          : 0x00000000`00010003 HICON__
   +0x030 hbrBackground    : 0x00000000`00000006 HBRUSH__
   +0x038 lpszMenuName     : 0x00000000`0000006d  "--- memory read error at address
0x00000000`0000006d ---"
   +0x040 lpszClassName    : 0x00000001`3ff635e0  "APPD1C"
```

```
   +0x048 hIconSm          : 0x00000000`04b4132f HICON__

0:000> dt wc ; AppD1B
Local var @ 0x23f7b0 Type tagWNDCLASSW
   +0x000 style            : 3
   +0x008 lpfnWndProc      : 0x00000001`3f991240     int64  AppD1B!WndProc+0
   +0x010 cbClsExtra       : 0n0
   +0x014 cbWndExtra       : 0n0
   +0x018 hInstance        : 0x00000001`3f990000 HINSTANCE__
   +0x020 hIcon            : 0xffffffff`b90303e9 HICON__
   +0x028 hCursor          : 0x00000000`00010003 HICON__
   +0x030 hbrBackground    : 0x00000000`00000006 HBRUSH__
   +0x038 lpszMenuName     : 0x00000000`0000006d  "--- memory read error at address
0x00000000`0000006d ---"
   +0x040 lpszClassName    : 0x00000001`3f9935e0  "APPD1B"
```

Note: AppD1A wasn't working because of structure member alignment. This models an old Windows 3.x project that was ported to x64. It had the minimum alignment in the past to reduce memory consumption:



AppD1B was working because the alignment was changed to default. AppD1C still used the same 1 byte alignment but because the bigger structure shifted members of the substructure it didn't crash.



Windows Malware Analysis Accelerated

with Memory Dumps

Dmitry Vostokov
Software Diagnostics Services

Published by OpenTask, Republic of Ireland

OpenTask books and magazines are available through booksellers and distributors worldwide. For further information or comments send requests to press@opentask.com.

A CIP catalogue record for this book is available from the British Library.

ISBN-l3: 978-1-908043-44-3 (Paperback)

First printing, 2013

Contents

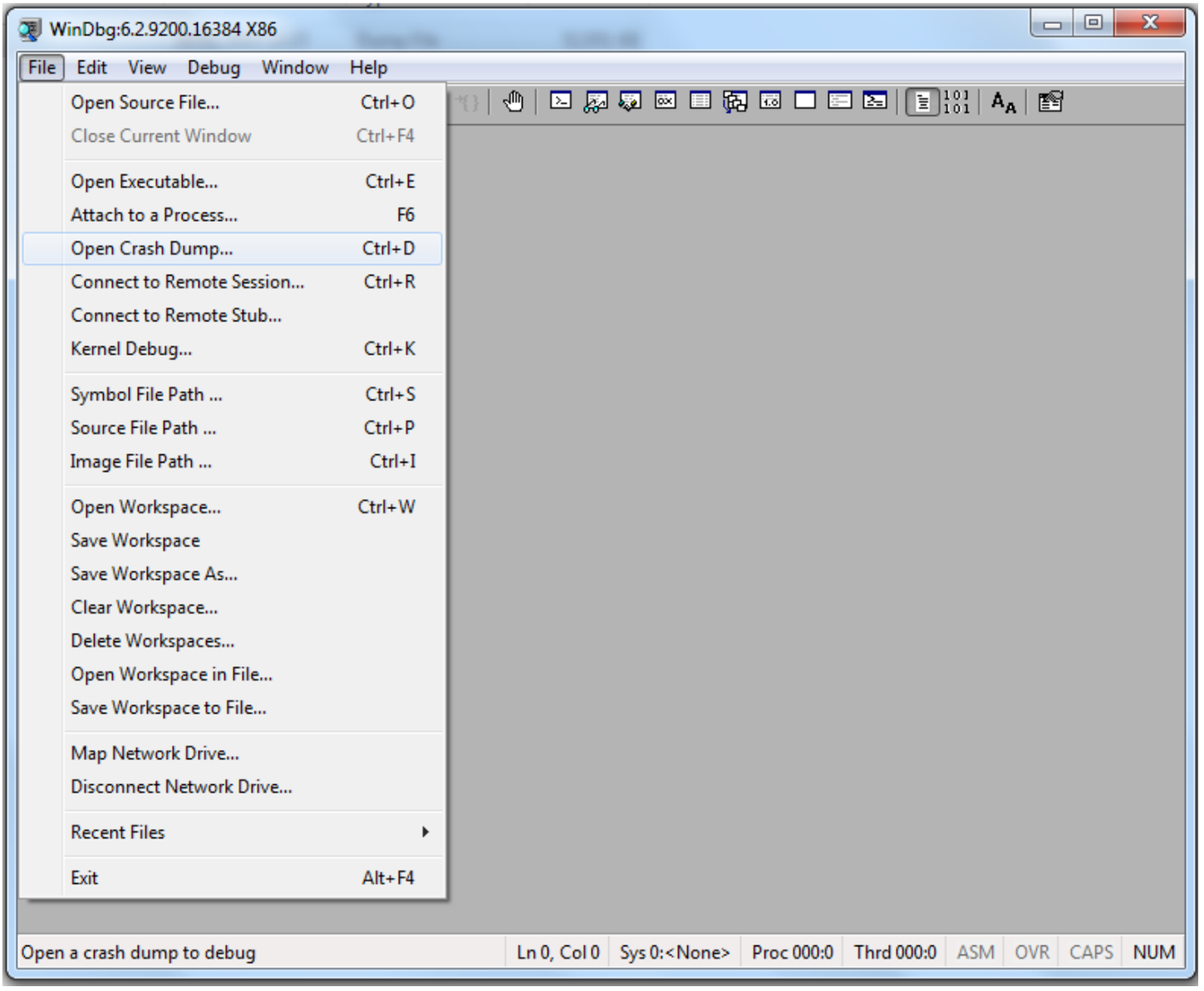
Exercise M1A

Goal: Look at module headers and version information before load.

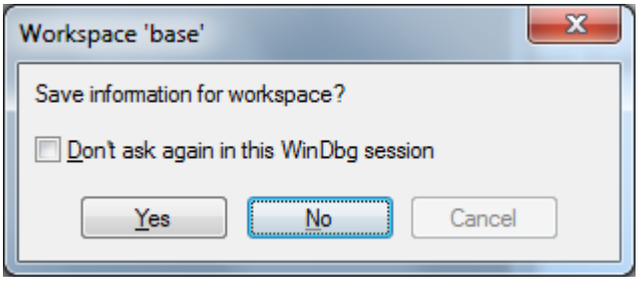
Patterns: Unknown Module

1. Launch WinDbg from Windows Kits \ Debugging Tools for Windows (X64) or Debugging Tools for Windows (X86)

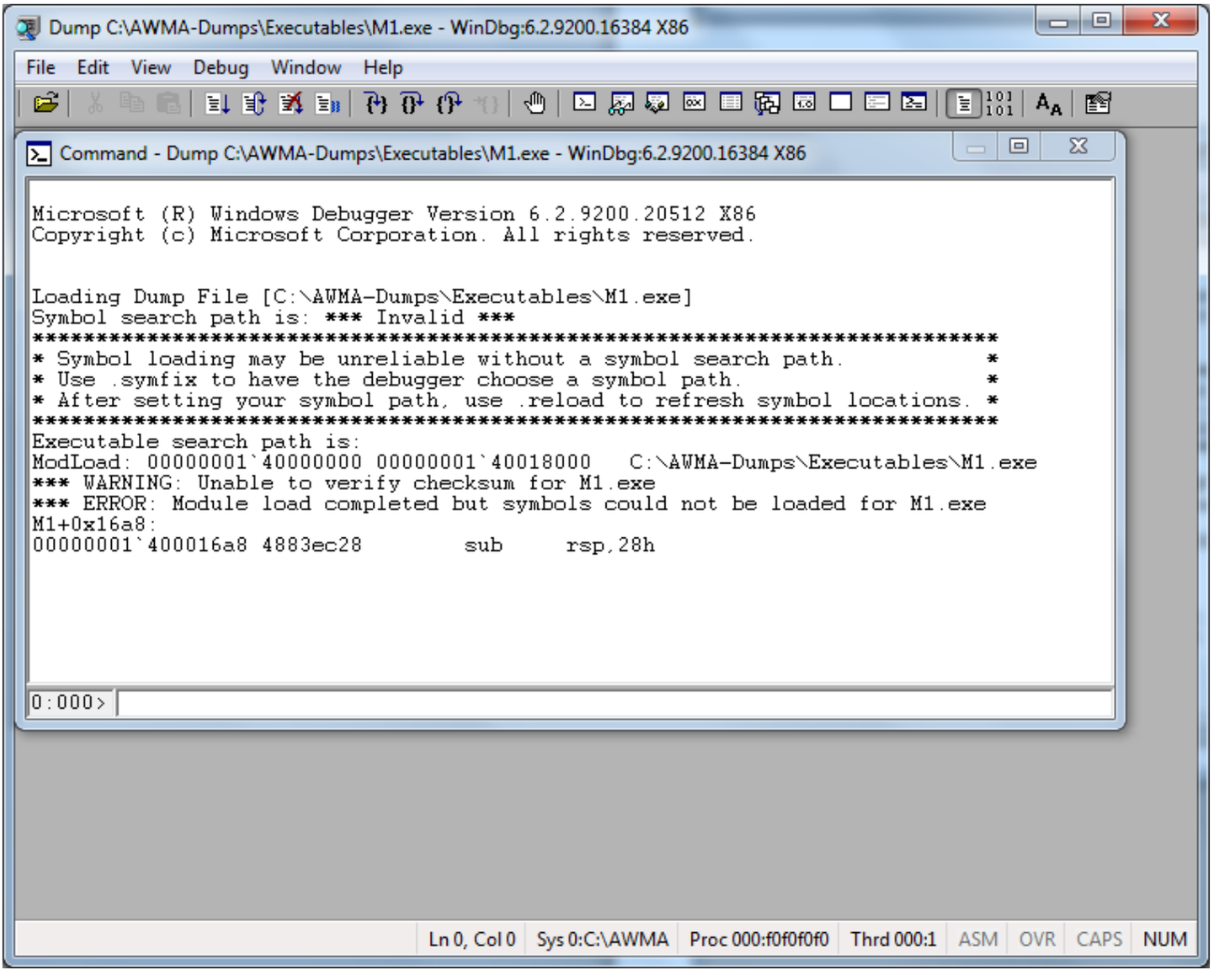
2. Open \AWMA-Dumps\Executables\M1.exe



3. If you are presented with this dialog say No:



4. You get the EXE file loaded:



5. Symbols are not necessary here.

6. Open a log file:

```
0:000> .logopen C:\AWMA-Dumps\M1A.log
Opened log file 'C:\AWMA-Dumps\M1A.log'
```

7. **lmv** command lists module information:

```
0:000> lmv
start             end                 module name
00000001`40000000 00000001`40018000   M1       C (no symbols)
    Loaded symbol image file: M1.exe
    Mapped memory image file: C:\AWMA-Dumps\Executables\M1.exe
    Image path: C:\AWMA-Dumps\Executables\M1.exe
    Image name: M1.exe
    Timestamp:        Mon Jan 28 15:24:45 2013 (5106983D)
    CheckSum:         00000000
    ImageSize:        00018000
    Translations:     0000.04b0 0000.04e4 0409.04b0 0409.04e4
```

Note module default load address.

8. **!lmi** command gives a bit more information:

```
0:000> !lmi 00000001`40000000
Loaded Module Info: [00000001`40000000]
         Module: M1
   Base Address: 0000000140000000
     Image Name: M1.exe
   Machine Type: 34404 (X64)
     Time Stamp: 5106983d Mon Jan 28 15:24:45 2013
           Size: 18000
       CheckSum: 0
Characteristics: 22
Debug Data Dirs: Type  Size     VA  Pointer
             CODEVIEW    3b,  e370,    cb70 RSDS - GUID: {3F1487A5-A6DC-4351-AD23-76FC12BB9482}
               Age: 1, Pdb: C:\Work\AWMA\M1\x64\Release\M1.pdb
                   ??    10,  e3ac,    cbac [Data not mapped]
     Image Type: FILE     - Image read successfully from debugger.
                 M1.exe
    Symbol Type: NONE     - PDB not found from image path.
    Load Report: no symbols loaded
```

Note a reference to a PDB file. If left by a developer it might give some clues as we in other exercises.

9. We dump the first kilobyte:

```
0:000> dc 00000001`40000000 L100
00000001`40000000  00905a4d 00000003 00000004 0000ffff  MZ..............
00000001`40000010  000000b8 00000000 00000040 00000000  ........@.......
00000001`40000020  00000000 00000000 00000000 00000000  ................
00000001`40000030  00000000 00000000 00000000 000000e8  ................
00000001`40000040  0eba1f0e cd09b400 4c01b821 685421cd  ........!..L.!Th
00000001`40000050  70207369 72676f72 63206d61 6f6e6e61  is program canno
00000001`40000060  65622074 6e757220 206e6920 20534f44  t be run in DOS
00000001`40000070  65646f6d 0a0d0d2e 00000024 00000000  mode....$.......
```

```
00000001`40000080  cb8e1818 98e0795c 98e0795c 98e0795c  ....\y..\y..\y..
00000001`40000090  982fbfad 98e0794e 982ebfad 98e07908  ../.Ny.......y..
00000001`400000a0  982dbfad 98e0795b 98e1795c 98e07903  ..-.[y..\y...y..
00000001`400000b0  98590ea0 98e07959 9833befe 98e0795e  ..Y.Yy....3.^y..
00000001`400000c0  9829befe 98e0795d 9877795c 98e0795d  ..).]y..\yw.]y..
00000001`400000d0  982cbefe 98e0795d 68636952 98e0795c  ..,.]y..Rich\y..
00000001`400000e0  00000000 00000000 00004550 00068664  ........PE..d...
00000001`400000f0  5106983d 00000000 00000000 002200f0  =..Q.........".
00000001`40000100  000b020b 00007400 0000d200 00000000  .....t.........
00000001`40000110  000016a8 00001000 40000000 00000001  ...........@....
00000001`40000120  00001000 00000200 00000006 00000000  ................
00000001`40000130  00000006 00000000 00018000 00000400  ................
00000001`40000140  00000000 81600002 00100000 00000000  ......`.........
00000001`40000150  00001000 00000000 00100000 00000000  ................
00000001`40000160  00001000 00000000 00000000 00000010  ................
00000001`40000170  00000000 00000000 0000eaa4 0000003c  ............<...
00000001`40000180  00015000 00001d68 00014000 0000078c  .P..h....@......
00000001`40000190  00000000 00000000 00017000 00000530  .........p..0...
00000001`400001a0  00009320 00000038 00000000 00000000   ...8...........
00000001`400001b0  00000000 00000000 00000000 00000000  ................
00000001`400001c0  0000e300 00000070 00000000 00000000  ....p...........
00000001`400001d0  00009000 000002a0 00000000 00000000  ................
00000001`400001e0  00000000 00000000 00000000 00000000  ................
00000001`400001f0  7865742e 00000074 0000731b 00001000  .text....s......
00000001`40000200  00007400 00000400 00000000 00000000  .t..............
00000001`40000210  00000000 60000020 6164722e 00006174  .... ..`.rdata..
00000001`40000220  00006366 00009000 00006400 00007800  fc.......d...x..
00000001`40000230  00000000 00000000 00000000 40000040  ............@..@
00000001`40000240  7461642e 00000061 00003900 00010000  .data....9......
00000001`40000250  00001400 0000dc00 00000000 00000000  ................
00000001`40000260  00000000 c0000040 6164702e 00006174  ....@....pdata..
00000001`40000270  0000078c 00014000 00000800 0000f000  .....@..........
00000001`40000280  00000000 00000000 00000000 40000040  ............@..@
00000001`40000290  7273722e 00000063 00001d68 00015000  .rsrc...h....P..
00000001`400002a0  00001e00 0000f800 00000000 00000000  ................
00000001`400002b0  00000000 40000040 6c65722e 0000636f  ....@..@.reloc..
00000001`400002c0  00000c52 00017000 00000e00 00011600  R....p..........
00000001`400002d0  00000000 00000000 00000000 42000040  ............@..B
00000001`400002e0  00000000 00000000 00000000 00000000  ................
00000001`400002f0  00000000 00000000 00000000 00000000  ................
00000001`40000300  00000000 00000000 00000000 00000000  ................
00000001`40000310  00000000 00000000 00000000 00000000  ................
00000001`40000320  00000000 00000000 00000000 00000000  ................
00000001`40000330  00000000 00000000 00000000 00000000  ................
00000001`40000340  00000000 00000000 00000000 00000000  ................
00000001`40000350  00000000 00000000 00000000 00000000  ................
00000001`40000360  00000000 00000000 00000000 00000000  ................
00000001`40000370  00000000 00000000 00000000 00000000  ................
00000001`40000380  00000000 00000000 00000000 00000000  ................
00000001`40000390  00000000 00000000 00000000 00000000  ................
00000001`400003a0  00000000 00000000 00000000 00000000  ................
00000001`400003b0  00000000 00000000 00000000 00000000  ................
00000001`400003c0  00000000 00000000 00000000 00000000  ................
00000001`400003d0  00000000 00000000 00000000 00000000  ................
00000001`400003e0  00000000 00000000 00000000 00000000  ................
00000001`400003f0  00000000 00000000 00000000 00000000  ................
```

10. **!dh** command dumps PE header:

```
0:000> !dh 00000001`40000000

File Type: EXECUTABLE IMAGE
FILE HEADER VALUES
    8664 machine (X64)
       6 number of sections
5106983D time date stamp Mon Jan 28 15:24:45 2013

       0 file pointer to symbol table
       0 number of symbols
      F0 size of optional header
      22 characteristics
            Executable
            App can handle >2gb addresses

OPTIONAL HEADER VALUES
     20B magic #
   11.00 linker version
    7400 size of code
    D200 size of initialized data
       0 size of uninitialized data
    16A8 address of entry point
    1000 base of code
         ----- new -----
0000000140000000 image base
    1000 section alignment
     200 file alignment
       2 subsystem (Windows GUI)
    6.00 operating system version
    0.00 image version
    6.00 subsystem version
   18000 size of image
     400 size of headers
       0 checksum
0000000000100000 size of stack reserve
0000000000001000 size of stack commit
0000000000100000 size of heap reserve
0000000000001000 size of heap commit
    8160  DLL characteristics
            High entropy VA supported
            Dynamic base
            NX compatible
            Terminal server aware
       0 [       0] address [size] of Export Directory
    EAA4 [      3C] address [size] of Import Directory
   15000 [    1D68] address [size] of Resource Directory
   14000 [     78C] address [size] of Exception Directory
       0 [       0] address [size] of Security Directory
   17000 [     530] address [size] of Base Relocation Directory
    9320 [      38] address [size] of Debug Directory
       0 [       0] address [size] of Description Directory
       0 [       0] address [size] of Special Directory
       0 [       0] address [size] of Thread Storage Directory
    E300 [      70] address [size] of Load Configuration Directory
       0 [       0] address [size] of Bound Import Directory
    9000 [     2A0] address [size] of Import Address Table Directory
       0 [       0] address [size] of Delay Import Directory
       0 [       0] address [size] of COR20 Header Directory
```

```
       0 [       0] address [size] of Reserved Directory


SECTION HEADER #1
   .text name
    731B virtual size
    1000 virtual address
    7400 size of raw data
     400 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
60000020 flags
         Code
         (no align specified)
         Execute Read

SECTION HEADER #2
  .rdata name
    6366 virtual size
    9000 virtual address
    6400 size of raw data
    7800 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
         Initialized Data
         (no align specified)
         Read Only


Debug Directories(2)
       Type       Size     Address  Pointer
       cv           3b        e370     cb70    Format: RSDS, guid, 1,
C:\Work\AWMA\M1\x64\Release\M1.pdb
       (    12)      10        e3ac     cbac

SECTION HEADER #3
   .data name
    3900 virtual size
   10000 virtual address
    1400 size of raw data
    DC00 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
C0000040 flags
         Initialized Data
         (no align specified)
         Read Write
```

```
SECTION HEADER #4
  .pdata name
     78C virtual size
   14000 virtual address
     800 size of raw data
    F000 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
         Initialized Data
         (no align specified)
         Read Only

SECTION HEADER #5
   .rsrc name
    1D68 virtual size
   15000 virtual address
    1E00 size of raw data
    F800 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
         Initialized Data
         (no align specified)
         Read Only

SECTION HEADER #6
  .reloc name
     C52 virtual size
   17000 virtual address
     E00 size of raw data
   11600 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
42000040 flags
         Initialized Data
         Discardable
         (no align specified)
         Read Only
```

Note Import Address Table and code .text section.

11. Let's look at Import Address Table before dynamic linking takes place:

```
0:000> dps 00000001`40000000+9000 L2A0/8
00000001`40009000  00000000`0000ed80
00000001`40009008  00000000`0000f34a
00000001`40009010  00000000`0000f33a
00000001`40009018  00000000`0000f326
00000001`40009020  00000000`0000f316
00000001`40009028  00000000`0000f304
00000001`40009030  00000000`0000f2f4
00000001`40009038  00000000`0000f2e0
```

```
00000001`40009040  00000000`0000f2d0
00000001`40009048  00000000`0000f2c4
00000001`40009050  00000000`0000f2b2
00000001`40009058  00000000`0000f29c
00000001`40009060  00000000`0000f28e
00000001`40009068  00000000`0000f282
00000001`40009070  00000000`0000eee4
00000001`40009078  00000000`0000eef6
00000001`40009080  00000000`0000ef0a
00000001`40009088  00000000`0000ef26
00000001`40009090  00000000`0000ef36
00000001`40009098  00000000`0000ef46
00000001`400090a0  00000000`0000ef5c
00000001`400090a8  00000000`0000ef6c
00000001`400090b0  00000000`0000ef7c
00000001`400090b8  00000000`0000ef8a
00000001`400090c0  00000000`0000efa0
00000001`400090c8  00000000`0000efb2
00000001`400090d0  00000000`0000efc8
00000001`400090d8  00000000`0000efd8
00000001`400090e0  00000000`0000efe4
00000001`400090e8  00000000`0000effa
00000001`400090f0  00000000`0000f00c
00000001`400090f8  00000000`0000f01a
00000001`40009100  00000000`0000f042
00000001`40009108  00000000`0000f05a
00000001`40009110  00000000`0000f06c
00000001`40009118  00000000`0000f086
00000001`40009120  00000000`0000f09c
00000001`40009128  00000000`0000f0b6
00000001`40009130  00000000`0000f0d0
00000001`40009138  00000000`0000f0ea
00000001`40009140  00000000`0000f0fe
00000001`40009148  00000000`0000f118
00000001`40009150  00000000`0000f12c
00000001`40009158  00000000`0000f148
00000001`40009160  00000000`0000f166
00000001`40009168  00000000`0000f17a
00000001`40009170  00000000`0000f18e
00000001`40009178  00000000`0000f19a
00000001`40009180  00000000`0000f1a8
00000001`40009188  00000000`0000f1b6
00000001`40009190  00000000`0000f1c0
00000001`40009198  00000000`0000f1d4
00000001`400091a0  00000000`0000f1e2
00000001`400091a8  00000000`0000f1fa
00000001`400091b0  00000000`0000f212
00000001`400091b8  00000000`0000f21e
00000001`400091c0  00000000`0000f226
00000001`400091c8  00000000`0000f238
00000001`400091d0  00000000`0000f242
00000001`400091d8  00000000`0000f24e
00000001`400091e0  00000000`0000f25a
00000001`400091e8  00000000`0000f26c
00000001`400091f0  00000000`0000f358
00000001`400091f8  00000000`00000000
00000001`40009200  00000000`0000eecc
00000001`40009208  00000000`0000eeba
00000001`40009210  00000000`0000eeae
00000001`40009218  00000000`0000eea0
```

```
00000001`40009220  00000000`0000ee8e
00000001`40009228  00000000`0000ee7e
00000001`40009230  00000000`0000ee6c
00000001`40009238  00000000`0000ee5c
00000001`40009240  00000000`0000ee4e
00000001`40009248  00000000`0000ee3c
00000001`40009250  00000000`0000ee28
00000001`40009258  00000000`0000ee1a
00000001`40009260  00000000`0000ee0e
00000001`40009268  00000000`0000edfa
00000001`40009270  00000000`0000ede6
00000001`40009278  00000000`0000edce
00000001`40009280  00000000`0000edc0
00000001`40009288  00000000`0000edac
00000001`40009290  00000000`0000ed9e
00000001`40009298  00000000`00000000

0:000> dc 00000001`40000000+00000000`0000ed80 L100
00000001`4000ed80  6f4c03c6 694c6461 72617262 00005779  ..LoadLibraryW..
00000001`4000ed90  4e52454b 32334c45 6c6c642e 02330000  KERNEL32.dll..3.
00000001`4000eda0  64616f4c 69727453 0057676e 6f4c021e  LoadStringW...Lo
00000001`4000edb0  63416461 656c6563 6f746172 00577372  adAcceleratorsW.
00000001`4000edc0  65470175 73654d74 65676173 03410057  u.GetMessageW.A.
00000001`4000edd0  6e617254 74616c73 63634165 72656c65  TranslateAcceler
00000001`4000ede0  726f7461 03430057 6e617254 74616c73  atorW.C.Translat
00000001`4000edf0  73654d65 65676173 00b60000 70736944  eMessage....Disp
00000001`4000ee00  68637461 7373654d 57656761 02260000  atchMessageW..&.
00000001`4000ee10  64616f4c 6e6f6349 02240057 64616f4c  LoadIconW.$.Load
00000001`4000ee20  73727543 0057726f 6552028a 74736967  CursorW...Regist
00000001`4000ee30  6c437265 45737361 00005778 72430071  erClassExW..q.Cr
00000001`4000ee40  65746165 646e6957 7845776f 03240057  eateWindowExW.$.
00000001`4000ee50  776f6853 646e6957 0000776f 7055035b  ShowWindow..[.Up
00000001`4000ee60  65746164 646e6957 0000776f 694400b3  dateWindow....Di
00000001`4000ee70  676f6c61 50786f42 6d617261 00ad0057  alogBoxParamW...
00000001`4000ee80  74736544 57796f72 6f646e69 00a10077  DestroyWindow...
00000001`4000ee90  57666544 6f646e69 6f725077 00005763  DefWindowProcW..
00000001`4000eea0  6542000e 506e6967 746e6961 00ea0000  ..BeginPaint....
00000001`4000eeb0  50646e45 746e6961 02720000 74736f50  EndPaint..r.Post
00000001`4000eec0  74697551 7373654d 00656761 6e4500e8  QuitMessage...En
00000001`4000eed0  61694464 00676f6c 52455355 642e3233  dDialog.USER32.d
00000001`4000eee0  00006c6c 654701e9 6d6f4374 646e616d  ll....GetCommand
00000001`4000eef0  656e694c 03860057 65447349 67677562  LineW...IsDebugg
00000001`4000ef00  72507265 6e657365 038b0074 72507349  erPresent...IsPr
00000001`4000ef10  7365636f 46726f73 75746165 72506572  ocessorFeaturePr
00000001`4000ef20  6e657365 02700074 4c746547 45747361  esent.p.GetLastE
00000001`4000ef30  726f7272 05250000 4c746553 45747361  rror..%.SetLastE
00000001`4000ef40  726f7272 022e0000 43746547 65727275  rror....GetCurre
00000001`4000ef50  6854746e 64616572 00006449 6e450140  ntThreadId..@.En
00000001`4000ef60  65646f63 6e696f50 00726574 65440118  codePointer...De
00000001`4000ef70  65646f63 6e696f50 00726574 78450173  codePointer.s.Ex
00000001`4000ef80  72507469 7365636f 02860073 4d746547  itProcess...GetM
00000001`4000ef90  6c75646f 6e614865 45656c64 00005778  oduleHandleExW..
00000001`4000efa0  654702bc 6f725074 64644163 73736572  ..GetProcAddress
00000001`4000efb0  03ef0000 746c754d 74794269 576f5465  ....MultiByteToW
00000001`4000efc0  43656469 00726168 654702e4 64745374  ideChar...GetStd
00000001`4000efd0  646e6148 0000656c 72570601 46657469  Handle....WriteF
00000001`4000efe0  00656c69 65470283 646f4d74 46656c75  ile...GetModuleF
00000001`4000eff0  4e656c69 57656d61 02c10000 50746547  ileNameW....GetP
00000001`4000f000  65636f72 65487373 00007061 6547025e  rocessHeap..^.Ge
00000001`4000f010  6c694674 70795465 036f0065 74696e49  tFileType.o.Init
```

```
00000001`4000f020  696c6169 7243657a 63697469 65536c61  ializeCriticalSe
00000001`4000f030  6f697463 646e416e 6e697053 6e756f43  ctionAndSpinCoun
00000001`4000f040  011f0074 656c6544 72436574 63697469  t...DeleteCritic
00000001`4000f050  65536c61 6f697463 02de006e 53746547  alSection...GetS
00000001`4000f060  74726174 6e497075 00576f66 7551043f  tartupInfoW.?.Qu
00000001`4000f070  50797265 6f667265 6e616d72 6f436563  eryPerformanceCo
00000001`4000f080  65746e75 022a0072 43746547 65727275  unter.*.GetCurre
00000001`4000f090  7250746e 7365636f 00644973 654702fb  ntProcessId...Ge
00000001`4000f0a0  73795374 546d6574 41656d69 6c694673  tSystemTimeAsFil
00000001`4000f0b0  6d695465 02470065 45746547 7269766e  eTime.G.GetEnvir
00000001`4000f0c0  656d6e6f 7453746e 676e6972 00005773  onmentStringsW..
00000001`4000f0d0  724601bd 6e456565 6f726976 6e656d6e  ..FreeEnvironmen
00000001`4000f0e0  72745374 73676e69 04bb0057 436c7452  tStringsW...RtlC
00000001`4000f0f0  75747061 6f436572 7865746e 04c20074  aptureContext...
00000001`4000f100  4c6c7452 756b6f6f 6e754670 6f697463  RtlLookupFunctio
00000001`4000f110  746e456e 00007972 745204c9 7269566c  nEntry....RtlVir
00000001`4000f120  6c617574 69776e55 0000646e 6e5505a0  tualUnwind....Un
00000001`4000f130  646e6168 4564656c 70656378 6e6f6974  handledException
00000001`4000f140  746c6946 00007265 6553055f 686e5574  Filter..._.SetUnh
00000001`4000f150  6c646e61 78456465 74706563 466e6f69  andledExceptionF
00000001`4000f160  65746c69 02290072 43746547 65727275  ilter.).GetCurre
00000001`4000f170  7250746e 7365636f 057e0073 6d726554  ntProcess.~.Term
```

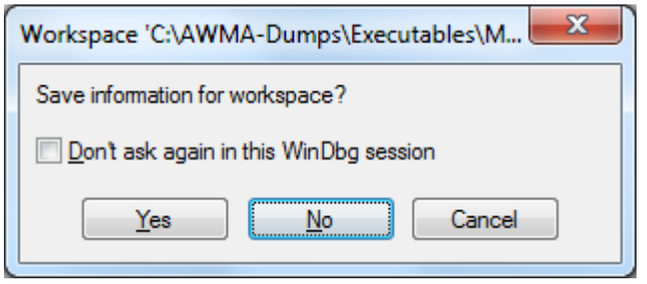
We see it contains function names that need to be imported from DLLs such as kernel32.dll and user32.dll.

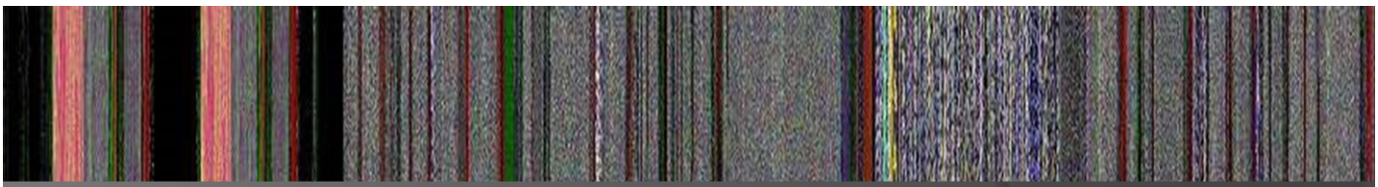
12. Close the log file:

```
0:000> .logclose
Closing open log file C:\AWMA-Dumps\M1A.log
```

13. To avoid possible confusion and glitches we recommend exiting WinDbg after each exercise.

If you are presented with this dialog choose No:





Reversing Disassembly Reconstruction

Accelerated

Dmitry Vostokov
Software Diagnostics Services

Published by OpenTask, Republic of Ireland

OpenTask books and magazines are available through booksellers and distributors worldwide. For further information or comments send requests to press@opentask.com.

A CIP catalogue record for this book is available from the British Library.

ISBN-l3: 978-1-908043-67-2 (Paperback)

Contents

Exercise R1

Goal: Review x64 assembly fundamentals; learn how to reconstruct stack trace manually.

ADDR Patterns: Universal Pointer, Symbolic Pointer S^2, Interpreted Pointer S^3, Context Pyramid

Memory Cell Diagrams: Register, Pointer, Stack Frame

1. Launch WinDbg from *Windows Kits* \ *Debugging Tools for Windows (X64)*

2. Choose *File* \ *Open Crash Dump…* menu option and load \ADDR\MemoryDumps\notepad.dmp.

3. You get the following output:

```
Microsoft (R) Windows Debugger Version 6.3.9600.16384 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.


Loading Dump File [C:\ADDR\MemoryDumps\notepad.dmp]
User Mini Dump File with Full Memory: Only application data is available

Symbol search path is: *** Invalid ***
************************************************************************
* Symbol loading may be unreliable without a symbol search path.       *
* Use .symfix to have the debugger choose a symbol path.               *
* After setting your symbol path, use .reload to refresh symbol locations. *
************************************************************************
Executable search path is:
Windows 7 Version 7601 (Service Pack 1) MP (4 procs) Free x64
Product: WinNt, suite: SingleUserTS Personal
Machine Name:
Debug session time: Wed Oct  9 20:25:46.000 2013 (UTC + 0:00)
System Uptime: 2 days 23:35:31.218
Process Uptime: 0 days 0:00:53.000
.............................
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for ntdll.dll -

************* Symbol Loading Error Summary **************
Module name            Error
ntdll                  The system cannot find the file specified

You can troubleshoot most symbol related issues by turning on symbol loading diagnostics (!sym
noisy) and repeating the command that caused symbols to be loaded.
You should also verify that your symbol search path (.sympath) is correct.
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for user32.dll -
user32!SfmDxSetSwapChainStats+0x1a:
00000000`77619e6a c3              ret
```

4. Set up a link to Microsoft symbol server and reload symbol files:

```
0:000> .symfix c:\mss

0:000> .reload
.............................
```

5. We get this stack trace:

```
0:000> k
Child-SP          RetAddr           Call Site
00000000`000efdc8 00000000`77619e9e user32!ZwUserGetMessage+0xa
00000000`000efdd0 00000000`ff131064 user32!GetMessageW+0x34
00000000`000efe00 00000000`ff13133c notepad!WinMain+0x182
00000000`000efe80 00000000`7771652d notepad!DisplayNonGenuineDlgWorker+0x2da
00000000`000eff40 00000000`7784c541 kernel32!BaseThreadInitThunk+0xd
00000000`000eff70 00000000`00000000 ntdll!RtlUserThreadStart+0x1d
```

6. Let's check the main CPU registers:

```
0:000> r
rax=00000000000206c0 rbx=00000000000efe40 rcx=000000000d0111c6
rdx=000000000000003b rsi=0000000000000001 rdi=0000000000000000
rip=0000000077619e6a rsp=00000000000efdc8 rbp=00000000ff130000
 r8=0000000000000000  r9=ffffffffffffffff r10=0000000000000000
r11=0000000004f424c8 r12=0000000000000000 r13=0000000000000000
r14=0000000000000000 r15=0000000000000000
iopl=0         nv up ei pl zr na po nc
cs=0033  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
user32!ZwUserGetMessage+0xa:
00000000`77619e6a c3              ret
```

Note: The register parts and naming are illustrated in MCD-R1.xlsx A section.

7. The current instruction registers (registers that are used and affected by the current instruction or semantically tied to it) can be checked by **r.** command:

```
0:000> r.
At return instr, rax = 206c0
```

8. Any register value or its named parts can be checked with **?** command:

```
0:000> ? r11
Evaluate expression: 83109064 = 00000000`04f424c8

0:000> ? r11d
Evaluate expression: 83109064 = 00000000`04f424c8

0:000> ? r11w
Evaluate expression: 9416 = 00000000`000024c8

0:000> ? r11b
Evaluate expression: 200 = 00000000`000000c8
```

9. Individual parts can also be interpreted using typed **r** command (here we format them as signed values, see WinDbg help for all other format types):

```
0:000> r r9
r9=ffffffffffffffff

0:000> r r9:iq
r9=-1
```

```
0:000> r r9:id
r9=-1 -1

0:000> r r9:iw
r9=65535 65535 65535 65535

0:000> r r9:ib
r9=255 255 255 255 255 255 255 255
```

10. Any registry value can be interpreted as a pointer to memory cells, a memory address (Universal Pointer pattern vs. a pointer that was originally designed to be such). However, memory contents at that address may be inaccessible or unknown as in the case of RCX and RDI below.

```
0:000> dp rcx
00000000`0d0111c6  ????????`???????? ????????`????????
00000000`0d0111d6  ????????`???????? ????????`????????
00000000`0d0111e6  ????????`???????? ????????`????????
00000000`0d0111f6  ????????`???????? ????????`????????
00000000`0d011206  ????????`???????? ????????`????????
00000000`0d011216  ????????`???????? ????????`????????
00000000`0d011226  ????????`???????? ????????`????????
00000000`0d011236  ????????`???????? ????????`????????
```

Note: The following output for R11 is illustrated in MCD-R1.xlsx B section.

```
0:000> dp r11
00000000`04f424c8  80000710`00020002 50200104`00000a00
00000000`04f424d8  00000000`ff130000 00000000`00000000
00000000`04f424e8  fffff900`c06f2760 00000000`00000000
00000000`04f424f8  fffff900`c06b3ef0 00000000`00000000
00000000`04f42508  00000000`00000000 000000a3`000000ea
00000000`04f42518  000002b9`0000054a 000000a5`000000ec
00000000`04f42528  000002b7`00000537 000007fe`fc00975c
00000000`04f42538  fffff900`c06f23d0 00000000`00000000

0:000> dp rax
00000000`000206c0  00260002`00000000 006e0065`0070004f
00000000`000206d0  0009002e`002e002e 006c0072`00740043
00000000`000206e0  00000000`004f002b 00610053`00260003
00000000`000206f0  00430009`00650076 002b006c`00720074
00000000`00020700  00040000`00000053 00650076`00610053
00000000`00020710  00730041`00260020 0000002e`002e002e
00000000`00020720  00000000`00000000 00670061`00500005
00000000`00020730  00650053`00200065 00700075`00260074

0:000> dp rbx
00000000`000efe40  00000000`0005096e 00000000`00000113
00000000`000efe50  00000000`00000001 00000000`00000000
00000000`000efe60  000002f8`0f5c7a0f 00000000`00000375
00000000`000efe70  00000000`ff13cab0 00000000`ff13133c
00000000`000efe80  00000000`00000000 00000000`00000000
00000000`000efe90  00000000`00000000 00000000`01985022
00000000`000efea0  00000000`00000000 00000000`01985022
00000000`000efeb0  00000000`00000000 00000000`ff13cab0
```

```
0:000> dp rdi
00000000`00000000  ????????`???????? ????????`????????
00000000`00000010  ????????`???????? ????????`????????
00000000`00000020  ????????`???????? ????????`????????
00000000`00000030  ????????`???????? ????????`????????
00000000`00000040  ????????`???????? ????????`????????
00000000`00000050  ????????`???????? ????????`????????
00000000`00000060  ????????`???????? ????????`????????
00000000`00000070  ????????`???????? ????????`????????
```

11. We can also specify a range or limit to just one value and use finer granularity for memory dumping:

```
0:000> dp rax L1
00000000`000206c0  00260002`00000000
```

Note: The similar output for R11 as below is illustrated in MCD-R1.xlsx C section.

```
0:000> dd rax
00000000`000206c0  00000000 00260002 0070004f 006e0065
00000000`000206d0  002e002e 0009002e 00740043 006c0072
00000000`000206e0  004f002b 00000000 00260003 00610053
00000000`000206f0  00650076 00430009 00720074 002b006c
00000000`00020700  00000053 00040000 00610053 00650076
00000000`00020710  00260020 00730041 002e002e 0000002e
00000000`00020720  00000000 00000000 00500005 00670061
00000000`00020730  00200065 00650053 00260074 00700075
```

Note: Visible 00xx00yy pattern in the output of **dp** command: UNICODE string fragments, an example of Regular Data memory analysis pattern.

```
0:000> dw rax
00000000`000206c0  0000 0000 0002 0026 004f 0070 0065 006e
00000000`000206d0  002e 002e 002e 0009 0043 0074 0072 006c
00000000`000206e0  002b 004f 0000 0000 0003 0026 0053 0061
00000000`000206f0  0076 0065 0009 0043 0074 0072 006c 002b
00000000`00020700  0053 0000 0000 0004 0053 0061 0076 0065
00000000`00020710  0020 0026 0041 0073 002e 002e 002e 0000
00000000`00020720  0000 0000 0000 0000 0005 0050 0061 0067
00000000`00020730  0065 0020 0053 0065 0074 0026 0075 0070

0:000> db rax
00000000`000206c0  00 00 00 00 02 00 26 00-4f 00 70 00 65 00 6e 00  ......&.O.p.e.n.
00000000`000206d0  2e 00 2e 00 2e 00 09 00-43 00 74 00 72 00 6c 00  ........C.t.r.l.
00000000`000206e0  2b 00 4f 00 00 00 00 00-03 00 26 00 53 00 61 00  +.O.......&.S.a.
00000000`000206f0  76 00 65 00 09 00 43 00-74 00 72 00 6c 00 2b 00  v.e...C.t.r.l.+.
00000000`00020700  53 00 00 00 00 00 04 00-53 00 61 00 76 00 65 00  S.......S.a.v.e.
00000000`00020710  20 00 26 00 41 00 73 00-2e 00 2e 00 2e 00 00 00   .&.A.s.........
00000000`00020720  00 00 00 00 00 00 00 00-05 00 50 00 61 00 67 00  ..........P.a.g.
00000000`00020730  65 00 20 00 53 00 65 00-74 00 26 00 75 00 70 00  e. .S.e.t.&.u.p.
```

Note: You may have noticed a slight delay when dumping memory pointed by registers. The faster equivalent approach is to use **@** prefix, for example: **@rax**:

```
0:000> dp @rax
00000000`000206c0  00260002`00000000 006e0065`0070004f
00000000`000206d0  0009002e`002e002e 006c0072`00740043
00000000`000206e0  00000000`004f002b 00610053`00260003
00000000`000206f0  00430009`00650076 002b006c`00720074
00000000`00020700  00040000`00000053 00650076`00610053
00000000`00020710  00730041`00260020 0000002e`002e002e
00000000`00020720  00000000`00000000 00670061`00500005
00000000`00020730  00650053`00200065 00700075`00260074
```

12. Notice a difference between a value and its organization in memory stemmed from the little-endian organization of Intel x86-x64 platform (least significant parts are located at lower addresses):

```
0:000> dp @rbp L1
00000000`ff130000  00000003`00905a4d

0:000> dd @rbp L2
00000000`ff130000  00905a4d 00000003
```

Note: The similar double word output for R11 is illustrated in MCD-R1.xlsx C section.

```
0:000> dp @rbp L1
00000000`ff130000  00000003`00905a4d

0:000> dw @rbp L4
00000000`ff130000  5a4d 0090 0003 0000

0:000> dp @rbp L1
00000000`ff130000  00000003`00905a4d

0:000> db @rbp L8
00000000`ff130000  4d 5a 90 00 03 00 00 00                         MZ......
```

13. Every value can be associated with a symbolic value from PDB symbols files or from the binary (exported symbols) if available. We call this Symbolic Pointer or S^2:

```
0:000> dps r11
00000000`04f424c8  80000710`00020002
00000000`04f424d0  50200104`00000a00
00000000`04f424d8  00000000`ff130000 notepad!CFileDialogEvents_QueryInterface <PERF>
(notepad+0x0)
00000000`04f424e0  00000000`00000000
00000000`04f424e8  fffff900`c06f2760
00000000`04f424f0  00000000`00000000
00000000`04f424f8  fffff900`c06b3ef0
00000000`04f42500  00000000`00000000
00000000`04f42508  00000000`00000000
00000000`04f42510  000000a3`000000ea
00000000`04f42518  000002b9`0000054a
00000000`04f42520  000000a5`000000ec
00000000`04f42528  000002b7`00000537
00000000`04f42530  000007fe`fc00975c comctl32!Edit_WndProc
00000000`04f42538  fffff900`c06f23d0
00000000`04f42540  00000000`00000000

0:000> ln 000007fe`fc00975c
(000007fe`fc00975c)   comctl32!Edit_WndProc   |  (000007fe`fc00a650)
comctl32!Edit_CalcChangeBlocks
Exact matches:
```

```
    comctl32!Edit_WndProc (<no parameter info>)

0:000> dt 000007fe`fc00975c
Edit_WndProc
Symbol  not found.
```

Note: The address **00000000`04f42530** that points to **000007fe`fc00975c** doesn't have an associated symbol:

```
0:000> dt 00000000`04f42530
Symbol not found at address 0000000004f42530.
```

Note: The next instruction pointer address contained in RIP should have an associated symbol of the current function in our example, because we have symbols for user32.dll:

```
0:000> ? @rip
Evaluate expression: 2002886250 = 00000000`77619e6a

0:000> dt @rip
ZwUserGetMessage
Symbol  not found.

0:000> r
rax=00000000000206c0 rbx=00000000000efe40 rcx=000000000d0111c6
rdx=000000000000003b rsi=0000000000000001 rdi=0000000000000000
rip=0000000077619e6a rsp=00000000000efdc8 rbp=00000000ff130000
 r8=0000000000000000  r9=ffffffffffffffff r10=0000000000000000
r11=0000000004f424c8 r12=0000000000000000 r13=0000000000000000
r14=0000000000000000 r15=0000000000000000
iopl=0         nv up ei pl zr na po nc
cs=0033  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
user32!ZwUserGetMessage+0xa:
00000000`77619e6a c3              ret
```

14. Now we come to the next pointer level after its value and its symbol: its interpretation. We call it an Interpreted Pointer, S^3. Such interpretation is implemented either via typed structures (**dt** command) or via various WinDbg extension commands (**!** Commands) that format information for us. In our example we would like to check memory pointed to by the value of RBX register. We suspect it might be MSG structure related to get message loop:

```
typedef struct tagMSG {
  HWND   hwnd;
  UINT   message;
  WPARAM wParam;
  LPARAM lParam;
  DWORD  time;
  POINT  pt;
} MSG;
```

```
0:000> dp @rbx
00000000`000efe40  00000000`0005096e 00000000`00000113
00000000`000efe50  00000000`00000001 00000000`00000000
00000000`000efe60  000002f8`0f5c7a0f 00000000`00000375
00000000`000efe70  00000000`ff13cab0 00000000`ff13133c
00000000`000efe80  00000000`00000000 00000000`00000000
00000000`000efe90  00000000`00000000 00000000`01985022
00000000`000efea0  00000000`00000000 00000000`01985022
00000000`000efeb0  00000000`00000000 00000000`ff13cab0
```

Note: The raw structure makes sense for WM_TIMER message (0x113) where wParam is a time ID (1) and usually a callback function (lParam) is NULL (0x0). Also mouse pointer data makes sense. Unfortunately, MSG structure is not available in symbol files available for notepad memory dump. However, we can load a different unrelated module with better symbol files, for example, CPUx64.exe from C:\ADDR\MemoryDumps\ExtraSymbols which was compiled as Windows application with full symbols and so should have structures necessary for thread message loop processing.

15. We add an additional symbol file path:

```
0:000> .sympath+ C:\ADDR\MemoryDumps\ExtraSymbols
Symbol search path is: srv*;C:\ADDR\MemoryDumps\ExtraSymbols
Expanded Symbol search path is:
SRV*c:\mss*http://msdl.microsoft.com/download/symbols;c:\addr\memorydumps\extrasymbols
```

We need to find an address to “load” CPUx64 module with its symbols. We choose a committed address 02000000 from the output of **!address** command:

```
0:000> !address


Mapping file section regions...
Mapping module regions...
Mapping PEB regions...
Mapping TEB and stack regions...
Mapping heap regions...
Mapping page heap regions...
Mapping other regions...
Mapping stack trace database regions...
Mapping activation context regions...

        BaseAddress      EndAddress+1        RegionSize     Type       State                 Protect             Usage
----------------------------------------------------------------------------------------------------------------------
--
[…]
         0`01ffe000        0`01fff000        0`00001000 MEM_PRIVATE MEM_RESERVE
PageHeap   [PageHeap: 18f1000; NormalHeap: 2920000]
         0`01fff000        0`02000000        0`00001000 MEM_PRIVATE MEM_COMMIT  PAGE_NOACCESS
PageHeap   [PageHeap: 18f1000; NormalHeap: 2920000]
         0`02000000        0`02001000        0`00001000 MEM_PRIVATE MEM_RESERVE
PageHeap   [PageHeap: 18f1000; NormalHeap: 2920000]
         0`02001000        0`02002000        0`00001000 MEM_PRIVATE MEM_COMMIT  PAGE_NOACCESS
PageHeap   [PageHeap: 18f1000; NormalHeap: 2920000]
         0`02002000        0`02003000        0`00001000 MEM_PRIVATE MEM_RESERVE
PageHeap   [PageHeap: 18f1000; NormalHeap: 2920000]
         0`02003000        0`02004000        0`00001000 MEM_PRIVATE MEM_COMMIT  PAGE_NOACCESS
PageHeap   [PageHeap: 18f1000; NormalHeap: 2920000]
[…]


0:000> .reload /f C:\ADDR\MemoryDumps\ExtraSymbols\CPUx64=02000000

0:000> lm m CPU*
start             end                 module name
00000000`02000000 00000000`02000000   CPUx64     (private pdb symbols)  c:\addr\memorydumps\extrasymbols\CPUx64.pdb
```

16. Now we are able to use MSG structure:

```
0:000> dt MSG
CPUx64!MSG
   +0x000 hwnd             : Ptr64 HWND__
   +0x008 message          : Uint4B
   +0x010 wParam           : Uint8B
   +0x018 lParam           : Int8B
   +0x020 time             : Uint4B
   +0x024 pt               : tagPOINT

0:000> dt -r MSG
CPUx64!MSG
   +0x000 hwnd             : Ptr64 HWND__
      +0x000 unused           : Int4B
   +0x008 message          : Uint4B
   +0x010 wParam           : Uint8B
   +0x018 lParam           : Int8B
   +0x020 time             : Uint4B
   +0x024 pt               : tagPOINT
      +0x000 x                : Int4B
      +0x004 y                : Int4B

0:000> dt -r MSG @rbx
CPUx64!MSG
   +0x000 hwnd             : 0x00000000`0005096e HWND__
      +0x000 unused           : 0n0
   +0x008 message          : 0x113
   +0x010 wParam           : 1
   +0x018 lParam           : 0n0
   +0x020 time             : 0xf5c7a0f
   +0x024 pt               : tagPOINT
      +0x000 x                : 0n760
      +0x004 y                : 0n885
```

17. When we have an exception such as a breakpoint or access violation the values of the thread CPU registers are saved in the so called exception context structure and valid for the currently executing function and its next instruction pointed to by RIP register (the topmost frame). In other situations such as a manual memory dump we can only be sure about some registers such as RIP and RSP:

```
0:000> k
Child-SP          RetAddr           Call Site
00000000`000efdc8 00000000`77619e9e user32!ZwUserGetMessage+0xa
00000000`000efdd0 00000000`ff131064 user32!GetMessageW+0x34
00000000`000efe00 00000000`ff13133c notepad!WinMain+0x182
00000000`000efe80 00000000`7771652d notepad!DisplayNonGenuineDlgWorker+0x2da
00000000`000eff40 00000000`7784c541 kernel32!BaseThreadInitThunk+0xd
00000000`000eff70 00000000`00000000 ntdll!RtlUserThreadStart+0x1d
```

```
0:000> r
rax=00000000000206c0 rbx=00000000000efe40 rcx=000000000d0111c6
rdx=000000000000003b rsi=0000000000000001 rdi=0000000000000000
rip=0000000077619e6a rsp=00000000000efdc8 rbp=00000000ff130000
 r8=0000000000000000  r9=ffffffffffffffff r10=0000000000000000
r11=0000000004f424c8 r12=0000000000000000 r13=0000000000000000
r14=0000000000000000 r15=0000000000000000
iopl=0         nv up ei pl zr na po nc
cs=0033  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
user32!ZwUserGetMessage+0xa:
00000000`77619e6a c3              ret
```

18. In any situation when we move down to the next frame, for example, to *GetMessageW+0x34* (which points to the next instruction after *ZwUserGetMessage* was called), we don't have its CPU registers values saved previously (**r** command gives values only for the topmost frame 0):

```
0:000> k
Child-SP          RetAddr           Call Site
00000000`000efdc8 00000000`77619e9e user32!ZwUserGetMessage+0xa
00000000`000efdd0 00000000`ff131064 user32!GetMessageW+0x34
00000000`000efe00 00000000`ff13133c notepad!WinMain+0x182
00000000`000efe80 00000000`7771652d notepad!DisplayNonGenuineDlgWorker+0x2da
00000000`000eff40 00000000`7784c541 kernel32!BaseThreadInitThunk+0xd
00000000`000eff70 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

0:000> ub 00000000`77619e9e
user32!GetMessageW+0xc:
00000000`77619e80 b90000feff      mov     ecx,0FFFE0000h
00000000`77619e85 410bc1          or      eax,r9d
00000000`77619e88 458bd1          mov     r10d,r9d
00000000`77619e8b 85c1            test    ecx,eax
00000000`77619e8d 0f85968d0100    jne     user32!GetMessageW+0x1b (00000000`77632c29)
00000000`77619e93 458bca          mov     r9d,r10d
00000000`77619e96 488bcb          mov     rcx,rbx
00000000`77619e99 e8c2ffffff      call    user32!ZwUserGetMessage (00000000`77619e60)

0:000> u 00000000`77619e9e
user32!GetMessageW+0x34:
00000000`77619e9e 817b0802010000  cmp     dword ptr [rbx+8],102h
00000000`77619ea5 448bd0          mov     r10d,eax
00000000`77619ea8 0f844e480000    je      user32!GetMessageW+0x49 (00000000`7761e6fc)
00000000`77619eae 817b08cc000000  cmp     dword ptr [rbx+8],0CCh
00000000`77619eb5 0f8441480000    je      user32!GetMessageW+0x49 (00000000`7761e6fc)
00000000`77619ebb 418bc2          mov     eax,r10d
00000000`77619ebe 4883c420        add     rsp,20h
00000000`77619ec2 5b              pop     rbx

0:000> kn
 # Child-SP          RetAddr           Call Site
00 00000000`000efdc8 00000000`77619e9e user32!ZwUserGetMessage+0xa
01 00000000`000efdd0 00000000`ff131064 user32!GetMessageW+0x34
02 00000000`000efe00 00000000`ff13133c notepad!WinMain+0x182
03 00000000`000efe80 00000000`7771652d notepad!DisplayNonGenuineDlgWorker+0x2da
04 00000000`000eff40 00000000`7784c541 kernel32!BaseThreadInitThunk+0xd
05 00000000`000eff70 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

0:000> .frame 1
01 00000000`000efdd0 00000000`ff131064 user32!GetMessageW+0x34
```

```
0:000> r
rax=00000000000206c0 rbx=00000000000efe40 rcx=000000000d0111c6
rdx=000000000000003b rsi=0000000000000001 rdi=0000000000000000
rip=0000000077619e6a rsp=00000000000efdc8 rbp=00000000ff130000
 r8=0000000000000000  r9=ffffffffffffffff r10=0000000000000000
r11=0000000004f424c8 r12=0000000000000000 r13=0000000000000000
r14=0000000000000000 r15=0000000000000000
iopl=0         nv up ei pl zr na po nc
cs=0033  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
user32!ZwUserGetMessage+0xa:
00000000`77619e6a c3              ret
```

19. But some CPU registers can be recovered such as RIP (saved address when using *call* instruction) and RSP (the stack pointer value that was before saving that RIP address). Other register values can be recovered too if they were not used in called frames or were saved in temporary memory cells (such as on stack). Let's recover some registers for the first few frames.

```
0:000> r
rax=00000000000206c0 rbx=00000000000efe40 rcx=000000000d0111c6
rdx=000000000000003b rsi=0000000000000001 rdi=0000000000000000
rip=0000000077619e6a rsp=00000000000efdc8 rbp=00000000ff130000
 r8=0000000000000000  r9=ffffffffffffffff r10=0000000000000000
r11=0000000004f424c8 r12=0000000000000000 r13=0000000000000000
r14=0000000000000000 r15=0000000000000000
iopl=0         nv up ei pl zr na po nc
cs=0033  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
user32!ZwUserGetMessage+0xa:
00000000`77619e6a c3              ret
```

Let's disassemble the current function:

```
0:000> uf user32!ZwUserGetMessage
user32!ZwUserGetMessage:
00000000`77619e60 4c8bd1          mov     r10,rcx
00000000`77619e63 b806100000      mov     eax,1006h
00000000`77619e68 0f05            syscall
00000000`77619e6a c3              ret
```

It is a very short function we see it overwrites R10 and EAX. Note that EAX value also don't correspond to what we see in the output of **r** command:

```
0:000> r @eax
eax=206c0
```

We see that RSP is not used inside *ZwUserGetMessage* function and its value should point to the return address of the caller, *GetMessageW* function during execution of **call** instruction:

```
0:000> dp @rsp
00000000`000efdc8  00000000`77619e9e 00000000`00000000
00000000`000efdd8  00000000`00000000 00000000`00000000
00000000`000efde8  00000000`00000000 00000000`01b20455
00000000`000efdf8  00000000`ff131064 00000000`01950048
00000000`000efe08  00000000`01b20455 000007fe`ff552164
00000000`000efe18  00000000`00000001 00000000`0000193c
00000000`000efe28  000007fe`00000000 00000000`00000000
00000000`000efe38  00000000`00000000 00000000`0005096e
```

```
0:000> ub 00000000`77619e9e
user32!GetMessageW+0xc:
00000000`77619e80 b90000feff      mov     ecx,0FFFE0000h
00000000`77619e85 410bc1          or      eax,r9d
00000000`77619e88 458bd1          mov     r10d,r9d
00000000`77619e8b 85c1            test    ecx,eax
00000000`77619e8d 0f85968d0100    jne     user32!GetMessageW+0x1b (00000000`77632c29)
00000000`77619e93 458bca          mov     r9d,r10d
00000000`77619e96 488bcb          mov     rcx,rbx
00000000`77619e99 e8c2ffffff      call    user32!ZwUserGetMessage (00000000`77619e60)

0:000> u 00000000`77619e9e
user32!GetMessageW+0x34:
00000000`77619e9e 817b0802010000  cmp     dword ptr [rbx+8],102h
00000000`77619ea5 448bd0          mov     r10d,eax
00000000`77619ea8 0f844e480000    je      user32!GetMessageW+0x49 (00000000`7761e6fc)
00000000`77619eae 817b08cc000000  cmp     dword ptr [rbx+8],0CCh
00000000`77619eb5 0f8441480000    je      user32!GetMessageW+0x49 (00000000`7761e6fc)
00000000`77619ebb 418bc2          mov     eax,r10d
00000000`77619ebe 4883c420        add     rsp,20h
00000000`77619ec2 5b              pop     rbx
```

This is RIP value but RSP should be the value before **call** instruction was executed. When a return value is saved RSP is decremented by 8 so the value of RSP before call should be the value of RSP pointing to the saved return address + 8:

```
0:000> ? @rsp + 8
Evaluate expression: 982480 = 00000000`000efdd0

0:000> k
Child-SP          RetAddr           Call Site
00000000`000efdc8 00000000`77619e9e user32!ZwUserGetMessage+0xa
00000000`000efdd0 00000000`ff131064 user32!GetMessageW+0x34
00000000`000efe00 00000000`ff13133c notepad!WinMain+0x182
00000000`000efe80 00000000`7771652d notepad!DisplayNonGenuineDlgWorker+0x2da
00000000`000eff40 00000000`7784c541 kernel32!BaseThreadInitThunk+0xd
00000000`000eff70 00000000`00000000 ntdll!RtlUserThreadStart+0x1d
```

Let's now find out RIP and RSP for the next frame (the caller of *GetMessageW* function). To find out RSP we need see how it was used in the callee, *GetMessageW* function before the callee called *ZwUserGetMessage*. We disassemble *GetMessageW* function:

```
0:000> uf user32!GetMessageW
user32!GetMessageW:
00000000`77619e74 fff3            push    rbx
00000000`77619e76 4883ec20        sub     rsp,20h
00000000`77619e7a 418bc0          mov     eax,r8d
00000000`77619e7d 488bd9          mov     rbx,rcx
00000000`77619e80 b90000feff      mov     ecx,0FFFE0000h
00000000`77619e85 410bc1          or      eax,r9d
00000000`77619e88 458bd1          mov     r10d,r9d
00000000`77619e8b 85c1            test    ecx,eax
00000000`77619e8d 0f85968d0100    jne     user32!GetMessageW+0x1b (00000000`77632c29)

user32!GetMessageW+0x29:
00000000`77619e93 458bca          mov     r9d,r10d
00000000`77619e96 488bcb          mov     rcx,rbx
00000000`77619e99 e8c2ffffff      call    user32!ZwUserGetMessage (00000000`77619e60)
00000000`77619e9e 817b0802010000  cmp     dword ptr [rbx+8],102h
```

```
00000000`77619ea5 448bd0          mov     r10d,eax
00000000`77619ea8 0f844e480000    je      user32!GetMessageW+0x49 (00000000`7761e6fc)

user32!GetMessageW+0x40:
00000000`77619eae 817b08cc000000  cmp     dword ptr [rbx+8],0CCh
00000000`77619eb5 0f8441480000    je      user32!GetMessageW+0x49 (00000000`7761e6fc)

user32!GetMessageW+0x51:
00000000`77619ebb 418bc2          mov     eax,r10d
00000000`77619ebe 4883c420        add     rsp,20h
00000000`77619ec2 5b              pop     rbx
00000000`77619ec3 c3              ret

user32!GetMessageW+0x49:
00000000`7761e6fc 48816310ffff0000 and     qword ptr [rbx+10h],0FFFFh
00000000`7761e704 e9b2b7ffff      jmp     user32!GetMessageW+0x51 (00000000`77619ebb)

user32!GetMessageW+0x1b:
00000000`77632c29 4183f9ff        cmp     r9d,0FFFFFFFFh
00000000`77632c2d 750d            jne     user32!GetMessageW+0x5a (00000000`77632c3c)

user32!GetMessageW+0x21:
00000000`77632c2f 4485c1          test    ecx,r8d
00000000`77632c32 7508            jne     user32!GetMessageW+0x5a (00000000`77632c3c)

user32!GetMessageW+0x26:
00000000`77632c34 4533d2          xor     r10d,r10d
00000000`77632c37 e95772feff      jmp     user32!GetMessageW+0x29 (00000000`77619e93)

user32!GetMessageW+0x5a:
00000000`77632c3c b957000000      mov     ecx,57h
00000000`77632c41 ff1561f60400    call    qword ptr [user32!_imp_RtlSetLastWin32Error
(00000000`776822a8)]
00000000`77632c47 4533d2          xor     r10d,r10d
00000000`77632c4a e96c72feff      jmp     user32!GetMessageW+0x51 (00000000`77619ebb)
```

We see that stack pointer was decremented by 0x20 (*sub* instruction) and also by 8 (*push* instruction) and so we add these values to RSP we found out previously for *ZwUserGetMessage* call, 00000000`000efdd0:

```
0:000> dps 00000000`000efdd0 + 20 + 8
00000000`000efdf8  00000000`ff131064 notepad!WinMain+0x182
00000000`000efe00  00000000`01950048
00000000`000efe08  00000000`01b20455
00000000`000efe10  000007fe`ff552164 msctf!UIWndProc
00000000`000efe18  00000000`00000001
00000000`000efe20  00000000`0000193c
00000000`000efe28  000007fe`00000000
00000000`000efe30  00000000`00000000
00000000`000efe38  00000000`00000000
00000000`000efe40  00000000`0005096e
00000000`000efe48  00000000`00000113
00000000`000efe50  00000000`00000001
00000000`000efe58  00000000`00000000
00000000`000efe60  000002f8`0f5c7a0f
00000000`000efe68  00000000`00000375
00000000`000efe70  00000000`ff13cab0 notepad!_xi_z
```

We see that *GetMessageW* was called from *WinMain* function:

```
0:000> ub 00000000`ff131064
notepad!WinMain+0xf5:
00000000`ff131046 ff1544b40000    call    qword ptr [notepad!_imp_SetWinEventHook
(00000000`ff13c490)]
00000000`ff13104c 488bd8          mov     rbx,rax
00000000`ff13104f eb00            jmp     notepad!WinMain+0x16f (00000000`ff131051)
00000000`ff131051 488d4c2440      lea     rcx,[rsp+40h]
00000000`ff131056 4533c9          xor     r9d,r9d
00000000`ff131059 4533c0          xor     r8d,r8d
00000000`ff13105c 33d2            xor     edx,edx
00000000`ff13105e ff1524b40000    call    qword ptr [notepad!_imp_GetMessageW
(00000000`ff13c488)]
```

The value of RSP before call should be adjusted by 8 due to saved return address:

```
0:000> ? 00000000`000efdf8 + 8
Evaluate expression: 982528 = 00000000`000efe00

0:000> k
Child-SP          RetAddr           Call Site
00000000`000efdc8 00000000`77619e9e user32!ZwUserGetMessage+0xa
00000000`000efdd0 00000000`ff131064 user32!GetMessageW+0x34
00000000`000efe00 00000000`ff13133c notepad!WinMain+0x182
00000000`000efe80 00000000`7771652d notepad!DisplayNonGenuineDlgWorker+0x2da
00000000`000eff40 00000000`7784c541 kernel32!BaseThreadInitThunk+0xd
00000000`000eff70 00000000`00000000 ntdll!RtlUserThreadStart+0x1d
```

And so on we are able to reconstruct the stack trace like a debugger. Note that we are able to correctly disassemble functions using **uf** command because function boundaries are saved in PDB symbol files or the start of the function is available from image file as an exported function. If such information is not available we would most likely have a truncated stack trace.

20. Other registers and memory values are reused and overwritten when we move down the frames so less and less information can be recovered. We call this ADDR pattern (Inverse) **Context Pyramid**.

21. We also introduce special Stack Frame memory cell diagrams. The case of stack frame for *GetMessageW* function before calling *ZwUserGetMessage* is illustrated in MCD-R1.xlsx section D.

22. To avoid possible confusion and glitches we recommend exiting WinDbg after each exercise.

A. Main Registers

RAX			
RAX		EAX	
RAX		EAX	AX
RAX		EAX	AH | AL
RSI			
RSI		ESI	
RSI		ESI	SI
RSI		ESI	| SIL
R8			
R8		R8D	
R8		R8D	R8W
R8		R8D	| R8B

B. Universal Pointer

We use a similar color for the value it points to

R11							

C. Pointing to a double word

R11							

D. Stack Frame

RSP				
8				
10				
18				
20				
28				
30				
38				
40				
48				
50				