WinDbg 内核调试常用命令(2)

接上一章继续介绍内核调试下的常用命令，这一章主要涉及内存相关、对象相关、驱动设备相关以及蓝屏 Dump 相关命令。介绍每个命令的主要作用，以及常用方式，不会涉及详细的命令参数，目的是能快速上手熟悉内核调试下的常用操作，而不是替代帮助文件。

内存相关

内存操作应该是调试最常用的，比如查看内存、修改内存等。本节介绍内核模式下常用的内存操作命令，大部分是内核模式下特有的命令，诸如 db/eb/dt/s 等基本内存命令则不会介绍。

!address

!address 命令显示内存信息，如内存范围、内存权限等。这条命令在用户模式下也能用，而且显示的信息比较丰富。

!address 命令不带参数时，显示所有内存信息。

```
kd> !address
  80800000 - 0026b000
        Usage      KernelSpaceUsageImage
        ImageName   ntoskrnl.exe

  80a6b000 - 0001f000
        Usage      KernelSpaceUsageImage
        ImageName   halacpi.dll

  ......

  f51d9000 - 00005000
        Usage      KernelSpaceUsageKernelStack
        KernelStack 81827020 : 340.7ac

  ......

  f894f000 - 00002000
        Usage      KernelSpaceUsageImage
        ImageName   swenum.sys

  f8951000 - 00256000
        Usage      KernelSpaceUsageNonPagedSystem

  f8ba8000 - 07038000
        Usage      KernelSpaceUsageNonPagedPoolExpansion
```

Usage 表示内存用途，如内核映像、非分页内存、内核栈、会话空间等。通过 Usage 就能大概了解某段内存的

使用情况，也为进一步分析内存指明了方向。

!address xxxxxxxx 显示指定地址的内存信息。

```
kd> x win32k!NtUserCreateWindowEx
bf8a8779 win32k!NtUserCreateWindowEx = <no type information>
kd> !address bf8a8779
  bf800000 - 001cf000
        Usage      KernelSpaceUsageImage
        ImageName   \SystemRoot\System32\win32k.sys
```

!vm

!vm 命令显示虚拟内存信息，查看当前虚拟内存状况、检查虚拟内存耗尽时都非常有用。

!vm 4 显示基本内存信息、进程虚拟内存、会话空间内存信息。

```
kd> !vm 4

*** Virtual Memory Usage ***
    Physical Memory:     130938 (    523752 Kb)
    Page File: \??\C:\pagefile.sys
      Current:    786432 Kb  Free Space:    780020 Kb
      Minimum:    786432 Kb  Maximum:     1572864 Kb
    Available Pages:     105581 (    422324 Kb)
    ResAvail Pages:       98280 (    393120 Kb)
    Locked IO Pages:         80 (       320 Kb)
    Free System PTEs:     54462 (    217848 Kb)
    Free NP PTEs:         28726 (    114904 Kb)
    Free Special NP:          0 (         0 Kb)
    Modified Pages:         148 (       592 Kb)
    Modified PF Pages:      148 (       592 Kb)
    NonPagedPool Usage:    1878 (      7512 Kb)
    NonPagedPool Max:     32055 (    128220 Kb)
    PagedPool 0 Usage:     1416 (      5664 Kb)
    PagedPool 1 Usage:      365 (      1460 Kb)
    PagedPool 2 Usage:      376 (      1504 Kb)
    PagedPool Usage:       2157 (      8628 Kb)
    PagedPool Maximum:    64512 (    258048 Kb)
    Shared Commit:          940 (      3760 Kb)
    Special Pool:             0 (         0 Kb)
    Shared Process:        1644 (      6576 Kb)
    PagedPool Commit:      2161 (      8644 Kb)
    Driver Commit:         1295 (      5180 Kb)
    Committed pages:      20930 (     83720 Kb)
    Commit limit:        321906 (   1287624 Kb)

    Total Private:        13733 (     54932 Kb)
        0340 SVCHOST.EXE      3395 (     13580 Kb)
```

```
        019c WINLOGON.EXE     1686 (     6744 Kb)
        0678 Explorer.EXE     1684 (     6736 Kb)
        01d4 LSASS.EXE        1672 (     6688 Kb)
        0308 SVCHOST.EXE       870 (     3480 Kb)
        03b8 SPOOLSV.EXE       814 (     3256 Kb)
        07b4 WMIPRVSE.EXE      463 (     1852 Kb)
        03f8 MSDTC.EXE         396 (     1584 Kb)
        0184 CSRSS.EXE         385 (     1540 Kb)
        0578 SVCHOST.EXE       358 (     1432 Kb)
        01c8 SERVICES.EXE      354 (     1416 Kb)
        06dc vmusrvc.exe       264 (     1056 Kb)
        0320 SVCHOST.EXE       263 (     1052 Kb)
        02d0 SVCHOST.EXE       263 (     1052 Kb)
        0450 VMSRVC.EXE        228 (      912 Kb)
        0290 SVCHOST.EXE       210 (      840 Kb)
        0470 SVCHOST.EXE       135 (      540 Kb)
        06f0 ctfmon.exe        126 (      504 Kb)
        04e0 SVCHOST.EXE        67 (      268 Kb)
        0518 VPCMAP.EXE         58 (      232 Kb)
        014c SMSS.EXE           35 (      140 Kb)
        0004 System              7 (       28 Kb)

    Terminal Server Memory Usage By Session:

    Session Paged Pool Maximum is 4096K
    Session View Space Maximum is 49152K

    Session ID 0 @ f8953000:
    Paged Pool Usage:          0K
    Commit Usage:            344K
```

其中基本内存信息部分，包括物理内存大小、页文件大小、分页内存大小、非分页内存大小等，按页面数和 KB 数显示。如果驱动分配非分页内存失败，则可以检查当前的非分页内存使用情况，看当前值是不是接近最大值，所以导致分配失败。在进程虚拟内存部分，按进程占用内存量从大到小往下排列，很容易看出最大内存占用的进程。最后一部分是会话空间内存信息，显示每个会话的内存情况。创建窗口、设置消息钩子等都会占用这块内存，如果创建窗口失败，则可以检查一下这块内存使用量是不是达到了最大值。

!memusage

!memusage 命令显示物理内存信息。主要是通过统计 PFN 数据库信息，得到物理内存的情况。PFN（Page Frame Number）表示页面帧编号，PFN 数据库用来描述每个物理内存页的状态。

!memusage 8 显示不同类型页面的大小。

```
lkd> !memusage 8
 loading PFN database
loading (100% complete)
Compiling memory usage data (99% Complete).
           Zeroed:     1 (     4 kb)
```

```
            Free:     25 (   100 kb)
         Standby: 167451 (669804 kb)
        Modified:   9709 ( 38836 kb)
   ModifiedNoWrite:     32 (   128 kb)
     Active/Valid: 344756 (1379024 kb)
      Transition:   2088 (  8352 kb)
             Bad:      0 (     0 kb)
         Unknown:      0 (     0 kb)
           TOTAL: 524062 (2096248 kb)
```

!memusage 命令显示更加详细的信息，包括页面对应的磁盘文件等。

```
lkd> !memusage
 loading PFN database
loading (100% complete)
Compiling memory usage data (99% Complete).
           Zeroed:      0 (     0 kb)
             Free:     69 (   276 kb)
          Standby: 134832 (539328 kb)
         Modified:   7863 ( 31452 kb)
   ModifiedNoWrite:     15 (    60 kb)
     Active/Valid: 379194 (1516776 kb)
      Transition:   2089 (  8356 kb)
             Bad:      0 (     0 kb)
         Unknown:      0 (     0 kb)
           TOTAL: 524062 (2096248 kb)
  Building kernel map
  Finished building kernel map
Scanning PFN database - (100% complete)

  Usage Summary (in Kb):
Control Valid Standby Dirty Shared Locked PageTables  name
ffffedcb 567948    724     0     0   724     0    AWE
845ca228 45608  82904     0     0     0     0  mapped_file( 002.part )
869aebd0   788    372     0   388     0     0  mapped_file( GdiPlus.dll )
85160d30 31804  101188     0     0     0     0  mapped_file( 004.part )
84d5a320 42504  96184     0     0     0     0  mapped_file( 003.part )
851fdd48  3260  14528     0     0     0     0  mapped_file( 001.part )
85cd26f0  1132  15380     0     0     0     0  mapped_file( $LogFile )
86e5ca28     0   1080     0     0     0     0  mapped_file( cimwin32.dll )
86466958     0     24     0     0     0     0  mapped_file( vqqsdl.dll )
8475cae8     0    492     0     0     0     0  mapped_file( msvcr90.dll )

......

--------    24      0     0 -----     0 -----  driver ( vmnetuserif.sys )
--------    24      0     0 -----     0 -----  driver ( vstor2.sys )
--------    40      0     0 -----     0 -----  driver ( cdfs.sys )
--------   168      0     0 -----     0 -----  driver ( udfs.sys )
```

```
--------    16      0     0 -----     0 -----  driver ( PROCEXP113.SYS )
--------    16      0     0 -----     0 -----  driver ( VlkdKnl.sys )
--------     8      0     0 -----     0 -----  driver ( kldbgdrv.sys )
--------  194736     0 13120 -----    16  7956        ( Paged Pool )
--------  5972      0  1276 -----     0   204        ( Kernel Stacks )
--------  54568     0     0 -----     0     4        ( NonPaged Pool )
Summary   1632484  544292 37864 105296 25632 24444  Total
```

!pte

!pte 命令显示指定地址对应的页表项（PTE）和页目录项（PDE）。显示地址是否有效，或者转换物理地址时经常用到。

```
kd> !process 0 0 notepad.exe
PROCESS 81ddb3e0  SessionId: 0  Cid: 016c    Peb: 7ffd7000  ParentCid: 0678
    DirBase: 1fc63000  ObjectTable: e1a30860  HandleCount:  43.
    Image: notepad.exe

kd> .process /i 81ddb3e0
You need to continue execution (press 'g' <enter>) for the context
to be switched. When the debugger breaks in again, you will be in
the new process context.
kd> g
Break instruction exception - code 80000003 (first chance)
nt!RtlpBreakWithStatusInstruction:
8081d97e int     3
kd> !pte 1000000
             VA 01000000
PDE at   C0300010       PTE at C0004000
contains 069FC067     contains 15B62025
pfn 69fc ---DA--UWEV    pfn 15b62 ----A--UREV

kd> x win32k!NtUserCreateWindowEx
bf8a8779 win32k!NtUserCreateWindowEx = <no type information>
kd> !pte bf8a8779
             VA bf8a8779
PDE at   C0300BF8       PTE at C02FE2A0
contains 1C59D063     contains 1E211021
pfn 1c59d ---DA--KWEV    pfn 1e211 ----A--KREV
```

分别显示了一个用户态地址对应的 PTE 和一个内核态地址对应的 PTE，从输出来看，两个地址都有效（V），一个是用户态（U），一个是内核态（K）。

!pfn

!pfn 命令显示物理内存页的详细信息，指定页面帧编号为参数。

```
kd> x win32k!NtUserCreateWindowEx
```

```
bf8a8779 win32k!NtUserCreateWindowEx = <no type information>
kd> !pte bf8a8779
            VA bf8a8779
PDE at   C0300BF8       PTE at C02FE2A0
contains 1C59D063     contains 1E211021
pfn 1c59d ---DA--KWEV    pfn 1e211 ----A--KREV

kd> !pfn 1e211
    PFN 0001E211 at address 812D3198
    flink      0000015E  blink / share count 00000001  pteaddress E13CE2E0
    reference count 0001   Cached    color 0
    restore pte 8BC6A460  containing page       01E71F  Active     P
      Shared
```

!db 和!eb

!db/!eb 命令用来显示、修改物理内存，类似的还有!dd/!dc/!dq/!ed 等。

```
kd> !process 0 0 notepad.exe
PROCESS 81eb91d8  SessionId: 0  Cid: 0174    Peb: 7ffdf000  ParentCid: 0674
    DirBase: 00075000  ObjectTable: e1650888  HandleCount:  41.
    Image: notepad.exe

kd> .process /i 81eb91d8
You need to continue execution (press 'g' <enter>) for the context
to be switched. When the debugger breaks in again, you will be in
the new process context.
kd> g
Break instruction exception - code 80000003 (first chance)
nt!RtlpBreakWithStatusInstruction:
8081d97e int     3
kd> db 1000000 l100
01000000  4d 5a 90 00 03 00 00 00-04 00 00 00 ff ff 00 00  MZ..............
01000010  b8 00 00 00 00 00 00 00-40 00 00 00 00 00 00 00  ........@.......
01000020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
01000030  00 00 00 00 00 00 00 00-00 00 00 00 e8 00 00 00  ................
01000040  0e 1f ba 0e 00 b4 09 cd-21 b8 01 4c cd 21 54 68  ........!..L.!Th
01000050  69 73 20 70 72 6f 67 72-61 6d 20 63 61 6e 6e 6f  is program canno
01000060  74 20 62 65 20 72 75 6e-20 69 6e 20 44 4f 53 20  t be run in DOS
01000070  6d 6f 64 65 2e 0d 0d 0a-24 00 00 00 00 00 00 00  mode....$.......
01000080  ec 85 5b bd a8 e4 35 ee-a8 e4 35 ee a8 e4 35 ee  ..[...5...5...5.
01000090  6b eb 3a ee a9 e4 35 ee-6b eb 55 ee a9 e4 35 ee  k.:...5.k.U...5.
010000a0  6b eb 68 ee bb e4 35 ee-a8 e4 34 ee 62 e4 35 ee  k.h...5...4.b.5.
010000b0  6b eb 6a ee bf e4 35 ee-6b eb 6b ee a9 e4 35 ee  k.j...5.k.k...5.
010000c0  6b eb 6f ee a9 e4 35 ee-52 69 63 68 a8 e4 35 ee  k.o...5.Rich..5.
010000d0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
010000e0  00 00 00 00 00 00 00 00-50 45 00 00 4c 01 03 00  ........PE..L...
010000f0  9a 5b 43 42 00 00 00 00-00 00 00 00 e0 00 0f 01  .[CB............
```

```
kd> !pte 1000000
            VA 01000000
PDE at   C0300010       PTE at C0004000
contains 00DFC067     contains 19D34025
pfn dfc ---DA--UWEV    pfn 19d34 ----A--UREV

kd> !db 19d34000
#19d34000 4d 5a 90 00 03 00 00 00-04 00 00 00 ff ff 00 00 MZ..............
#19d34010 b8 00 00 00 00 00 00 00-40 00 00 00 00 00 00 00 ........@.......
#19d34020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
#19d34030 00 00 00 00 00 00 00 00-00 00 00 00 e8 00 00 00 ................
#19d34040 0e 1f ba 0e 00 b4 09 cd-21 b8 01 4c cd 21 54 68 ........!..L.!Th
#19d34050 69 73 20 70 72 6f 67 72-61 6d 20 63 61 6e 6e 6f is program canno
#19d34060 74 20 62 65 20 72 75 6e-20 69 6e 20 44 4f 53 20 t be run in DOS
#19d34070 6d 6f 64 65 2e 0d 0d 0a-24 00 00 00 00 00 00 00 mode....$.......
kd> !db 19d34080
#19d34080 ec 85 5b bd a8 e4 35 ee-a8 e4 35 ee a8 e4 35 ee ..[...5...5...5.
#19d34090 6b eb 3a ee a9 e4 35 ee-6b eb 55 ee a9 e4 35 ee k.:...5.k.U...5.
#19d340a0 6b eb 68 ee bb e4 35 ee-a8 e4 34 ee 62 e4 35 ee k.h...5...4.b.5.
#19d340b0 6b eb 6a ee bf e4 35 ee-6b eb 6b ee a9 e4 35 ee k.j...5.k.k...5.
#19d340c0 6b eb 6f ee a9 e4 35 ee-52 69 63 68 a8 e4 35 ee k.o...5.Rich..5.
#19d340d0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
#19d340e0 00 00 00 00 00 00 00 00-50 45 00 00 4c 01 03 00 ........PE..L...
#19d340f0 9a 5b 43 42 00 00 00 00-00 00 00 00 e0 00 0f 01 .[CB............
```

!pool

!pool 命令显示内核内存池信息。

!pool xxxxxxxx 显示指定内存的信息。

```
kd> !process 0 0 notepad.exe
PROCESS 81eb91d8  SessionId: 0  Cid: 0174    Peb: 7ffdf000  ParentCid: 0674
    DirBase: 00075000  ObjectTable: e1650888  HandleCount:  41.
    Image: notepad.exe

kd> !pool 81eb91d8
Pool page 81eb91d8 region is Nonpaged pool
 81eb9000 size:   a8 previous size:    0  (Allocated)  Mdl
 81eb90a8 size:   18 previous size:   a8  (Free)       ....
 81eb90c0 size:   40 previous size:   18  (Allocated)  FatE
 81eb9100 size:   a8 previous size:   40  (Allocated)  File (Protected)
 81eb91a8 size:   10 previous size:   a8  (Free)       Irp
*81eb91b8 size:  298 previous size:   10  (Allocated) *Proc (Protected)
        Pooltag Proc : Process objects, Binary : nt!ps
 81eb9450 size:   40 previous size:  298  (Allocated)  FatE
 ......
kd> ?? sizeof(nt!_EPROCESS)
```

```
unsigned int 0x278
kd> ?? sizeof(nt!_OBJECT_HEADER)
unsigned int 0x20
kd> ?? sizeof(nt!_POOL_HEADER)
unsigned int 8
kd> dt nt!_POOL_HEADER 81eb91b8
   +0x000 PreviousSize     : 0y000000010 (0x2)
   +0x000 PoolIndex        : 0y0000000 (0)
   +0x002 BlockSize        : 0y001010011 (0x53)
   +0x002 PoolType         : 0y0000101 (0x5)
   +0x000 Ulong1           : 0xa530002
   +0x004 PoolTag          : 0xe36f7250
   +0x004 AllocatorBackTraceIndex : 0x7250
   +0x006 PoolTagHash      : 0xe36f
kd> dt nt!_OBJECT_HEADER 81eb91c0
   +0x000 PointerCount     : 14
   +0x004 HandleCount      : 1
   +0x004 NextToFree       : 0x00000001
   +0x008 Type             : 0x81f9b040 _OBJECT_TYPE
   +0x00c NameInfoOffset   : 0 ''
   +0x00d HandleInfoOffset : 0 ''
   +0x00e QuotaInfoOffset  : 0 ''
   +0x00f Flags            : 0x20 ' '
   +0x010 ObjectCreateInfo : 0x81e12438 _OBJECT_CREATE_INFORMATION
   +0x010 QuotaBlockCharged : 0x81e12438
   +0x014 SecurityDescriptor : 0xe1868463
   +0x018 Body             : _QUAD
kd> ? 53 * 8
Evaluate expression: 664 = 00000298
```

!poolused

!poolused 命令按照内存池 Tag 显示内存池信息。

!poolused 命令显示所有 Tag 的内存池信息，按 Tag 名排列。

```
lkd> !poolused
   Sorting by  Tag

  Pool Used:
           NonPaged            Paged
 Tag    Allocs     Used    Allocs     Used
  OPM        0        0         2       64   UNKNOWN pooltag ' OPM', please update pooltag.txt
 010X        1       24         0        0   UNKNOWN pooltag '010X', please update pooltag.txt
 100X        4       88         0        0   UNKNOWN pooltag '100X', please update pooltag.txt
 110X        1       48         0        0   UNKNOWN pooltag '110X', please update pooltag.txt
 1394      602   149376       128     9760 UNKNOWN pooltag '1394', please update pooltag.txt
 ......
```

```
 Proc      71    46576        0       0  Process objects , Binary: nt!ps
 ......
 vdbg       1      448        0       0  UNKNOWN pooltag 'vdbg', please update pooltag.txt
 virt     230    11800        5    1280 Virtual Machine Manager (VPC/VS) , Binary: vmm.sys
 whea      12    17096        0       0  UNKNOWN pooltag 'whea', please update pooltag.txt
 TOTAL      97080 37977336     94811 92370752
```

!poolused 2 命令显示非分页内存信息，按内存大小排列。

```
lkd> !poolused 2
   Sorting by  NonPaged Pool Consumed

  Pool Used:
           NonPaged           Paged
 Tag    Allocs     Used    Allocs     Used
 EtwB      146  6672384        16   802816Etw Buffer , Binary: nt!etw
 Cont      309  2648608         0        0 Contiguous  physical  memory  allocations  for  device
drivers
 Pool        5  2253224         0        0  Pool tables, etc.
 HTab      505  1955584         0        0  UNKNOWN pooltag 'HTab', please update pooltag.txt
 Wdm       718  1542936         0        0  WDM
 Devi      598  1234176         0        0  Device objects
 NV       3389  1039168         0        0  nVidia video driver
 File     5281   897688         0        0  File objects
 ......
 Txls        0        0         1       32   TXF_CANCEL_LSN , Binary: ntfs.sys
 PcSt        0        0        36     4608  WDM audio stuff
 TOTAL      98637 38128336     95265 91889040
```

!poolused 4 命令显示分页内存信息，按内存大小排列。

```
lkd> !poolused 4
   Sorting by  Paged Pool Consumed

  Pool Used:
           NonPaged           Paged
 Tag    Allocs     Used    Allocs     Used
 MmSt        0        0      3929 28541680 Mm section object prototype ptes , Binary: nt!mm
 CM25        0        0      3942 18165760 Internal  Configuration  manager  allocations  ,  Binary:
nt!cm
 MmRe        0        0       651 10185528 UNKNOWN pooltag 'MmRe', please update pooltag.txt
 CM35        0        0        22  3301376  Internal  Configuration  manager  allocations  ,  Binary:
nt!cm
 NtFB        0        0        85  2839560  BitmpSup.c , Binary: ntfs.sys
 ClfI        0        0        21  2648560  CLFS_MARSHALBUFFER_TAG , Binary: clfs.sys
 ......
 FlmC        3      288         0        0  UNKNOWN pooltag 'FlmC', please update pooltag.txt
 AlIn      246    26016         0        0  ALPC Internal allocation , Binary: nt!alpc
 TOTAL      98213 4333055120     94529 91533248
```

!poolused 2 和!poolused 4 两条命令检查内存池当前使用情况非常方便，很容易定位内存使用量大的模块。所

有的 Tag 都会自动在 triage\pooltag.txt 文件中查找对应的模块，如果没有显示模块，也可以在所有驱动文件中查找字符串，从而找到消耗内存的驱动。

!poolfind

!poolfind 命令显示所有指定 Tag 的内存信息。

!poolfind xxxx 0 命令显示在非分页池中所有 Tag 为 xxxx 的内存信息。

```
lkd> !poolfind Proc 0

Scanning large pool allocation table for Tag: Proc (84a6c000 : 84c6c000)


Searching NonPaged pool (80000000 : ffc00000) for Tag: Proc

8408ebb8 size:  290 previous size:  2b8  (Allocated) Proc (Protected)
845e5000 size:  290 previous size:    0  (Allocated) Proc (Protected)
84697758 size:  290 previous size:   30  (Allocated) Proc (Protected)
846d4480 size:  290 previous size:   70  (Allocated) Proc (Protected)
......
8735cc50 size:  290 previous size:   58  (Allocated) Proc (Protected)

lkd> !process (8408ebb8 + 20) 0
PROCESS 8408ebd8  SessionId: none  Cid: 0004    Peb: 00000000  ParentCid: 0000
    DirBase: 00122000  ObjectTable: 88a000c0  HandleCount: 932.
    Image: System
```

!poolfind xxxx 1 命令显示在分页池中所有 Tag 为 xxxx 的内存信息。

```
lkd> !poolfind Sect 1

Scanning large pool allocation table for Tag: Sect (84a6c000 : 84c6c000)


Searching Paged pool (80000000 : ffc00000) for Tag: Sect

816098d0 size:   58 previous size:   38  (Allocated) Sect (Protected)
81610570 size:   58 previous size:    8  (Allocated) Sect (Protected)
8161f130 size:   68 previous size:   40  (Allocated) Sect (Protected)
......
c16fd378 size:   58 previous size:  378  (Allocated) Sect (Protected)
c17030e8 size:   58 previous size:   80  (Allocated) Sect (Protected)
```

对象相关

在 Windows 内核中，所有资源都是以对象形式存在的，如进程对象、文件对象等。大部分内核函数都会和对象管理器打交道，系统大部分时间也在操作不同的对象。句柄是对象的上层抽象，不同的句柄可以表示同一个对象，不同的进程可以有值相同的句柄。由句柄可以得到对象，由对象可以生成句柄。

!handle

!handle 命令显示句柄信息，包括句柄类型、句柄引用计数、句柄名等。

!handle xxxx 命令显示当前进程中指定句柄的信息，如果不指定 xxxx，或者指定 0 则显示当前进程所有句柄的信息。

```
kd> !handle
processor number 0, process 808a0000
PROCESS 808a0000  SessionId: none  Cid: 0000    Peb: 00000000  ParentCid: 0000
    DirBase: 00039000  ObjectTable: e1000d50  HandleCount: 203.
    Image: Idle

Handle table at e1002000 with 203 Entries in use
0004: Object: 81f9b9e8  GrantedAccess: 001f0fff Entry: e1002008
Object: 81f9b9e8  Type: (81f9b040) Process
    ObjectHeader: 81f9b9d0 (old version)
        HandleCount: 2  PointerCount: 65

0008: Object: 81f9b338  GrantedAccess: 00000000 Entry: e1002010
Object: 81f9b338  Type: (81f9be70) Thread
    ObjectHeader: 81f9b320 (old version)
        HandleCount: 1  PointerCount: 1

......

01a0: Object: 81f130d0  GrantedAccess: 00140003 Entry: e1002340
Object: 81f130d0  Type: (81fce040) File
    ObjectHeader: 81f130b8 (old version)
        HandleCount: 1  PointerCount: 2
        Directory Object: 00000000  Name: \pagefile.sys {HarddiskVolume1}

......

0474: Object: e13d7598  GrantedAccess: 00020019 Entry: e10028e8
Object: e13d7598  Type: (81f917e0) Key
    ObjectHeader: e13d7580 (old version)
        HandleCount: 1  PointerCount: 1
        Directory Object: 00000000  Name: \REGISTRY\MACHINE\HARDWARE\DESCRIPTION\SYSTEM\BIOS

kd> !handle 0
processor number 0, process 808a0000
PROCESS 808a0000  SessionId: none  Cid: 0000    Peb: 00000000  ParentCid: 0000
    DirBase: 00039000  ObjectTable: e1000d50  HandleCount: 203.
    Image: Idle

Handle table at e1002000 with 203 Entries in use
0004: Object: 81f9b9e8  GrantedAccess: 001f0fff Entry: e1002008
```

```
Object: 81f9b9e8  Type: (81f9b040) Process
    ObjectHeader: 81f9b9d0 (old version)
        HandleCount: 2  PointerCount: 65

0008: Object: 81f9b338  GrantedAccess: 00000000 Entry: e1002010
Object: 81f9b338  Type: (81f9be70) Thread
    ObjectHeader: 81f9b320 (old version)
        HandleCount: 1  PointerCount: 1

......

01a0: Object: 81f130d0  GrantedAccess: 00140003 Entry: e1002340
Object: 81f130d0  Type: (81fce040) File
    ObjectHeader: 81f130b8 (old version)
        HandleCount: 1  PointerCount: 2
        Directory Object: 00000000  Name: \pagefile.sys {HarddiskVolume1}

......

0474: Object: e13d7598  GrantedAccess: 00020019 Entry: e10028e8
Object: e13d7598  Type: (81f917e0) Key
    ObjectHeader: e13d7580 (old version)
        HandleCount: 1  PointerCount: 1
        Directory Object: 00000000  Name: \REGISTRY\MACHINE\HARDWARE\DESCRIPTION\SYSTEM\BIOS

kd> !handle 1a0
processor number 0, process 808a0000
PROCESS 808a0000  SessionId: none  Cid: 0000    Peb: 00000000  ParentCid: 0000
    DirBase: 00039000  ObjectTable: e1000d50  HandleCount: 203.
    Image: Idle

Handle table at e1002000 with 203 Entries in use
01a0: Object: 81f130d0  GrantedAccess: 00140003 Entry: e1002340
Object: 81f130d0  Type: (81fce040) File
    ObjectHeader: 81f130b8 (old version)
        HandleCount: 1  PointerCount: 2
        Directory Object: 00000000  Name: \pagefile.sys {HarddiskVolume1}
```

!handle xxxx yy zzzzzzzz 命令显示指定进程中指定句柄的信息，xxxx 表示指定句柄，yy 表示显示掩码（缺省为 3），zzzzzzzz 表示指定进程。

```
kd> !process -1 0
PROCESS 808a0000  SessionId: none  Cid: 0000    Peb: 00000000  ParentCid: 0000
    DirBase: 00039000  ObjectTable: e1000d50  HandleCount: 203.
    Image: Idle

kd> !process 0 0 notepad.exe
PROCESS 81eb91d8  SessionId: 0  Cid: 0174    Peb: 7ffdf000  ParentCid: 0674
    DirBase: 00075000  ObjectTable: e1650888  HandleCount:  88.
```

```
    Image: notepad.exe

kd> !handle 0 3 81eb91d8
processor number 0, process 81eb91d8
PROCESS 81eb91d8  SessionId: 0  Cid: 0174    Peb: 7ffdf000  ParentCid: 0674
    DirBase: 00075000  ObjectTable: e1650888  HandleCount:  88.
    Image: notepad.exe

Handle table at e1033000 with 88 Entries in use
0004: Object: e1007ef0  GrantedAccess: 00000003 Entry: e1033008
Object: e1007ef0  Type: (81f96ad0) KeyedEvent
    ObjectHeader: e1007ed8 (old version)
        HandleCount: 23  PointerCount: 24
        Directory Object: e1001508  Name: CritSecOutOfMemoryEvent

0008: Object: 81dcf128  GrantedAccess: 001f0003 Entry: e1033010
Object: 81dcf128  Type: (81f9a720) Event
    ObjectHeader: 81dcf110 (old version)
        HandleCount: 1  PointerCount: 1

......

01a0: Object: 81dc3118  GrantedAccess: 00100003 Entry: e1033340
Object: 81dc3118  Type: (81f96040) Semaphore
    ObjectHeader: 81dc3100 (old version)
        HandleCount: 1  PointerCount: 1

......

01c4: Object: 81d77568  GrantedAccess: 00100020 (Inherit) Entry: e1033388
Object: 81d77568  Type: (81fce040) File
    ObjectHeader: 81d77550 (old version)
        HandleCount: 1  PointerCount: 1
        Directory  Object:  00000000    Name:  \Documents  and  Settings\Administrator\ 桌 面 \
{HarddiskVolume1}

kd> !handle 1a0 3 81eb91d8
processor number 0, process 81eb91d8
PROCESS 81eb91d8  SessionId: 0  Cid: 0174    Peb: 7ffdf000  ParentCid: 0674
    DirBase: 00075000  ObjectTable: e1650888  HandleCount:  88.
    Image: notepad.exe

Handle table at e1033000 with 88 Entries in use
01a0: Object: 81dc3118  GrantedAccess: 00100003 Entry: e1033340
Object: 81dc3118  Type: (81f96040) Semaphore
    ObjectHeader: 81dc3100 (old version)
        HandleCount: 1  PointerCount: 1
```

!object

!object 命令显示对象信息。

!object xxxxxxxx 命令显示指定对象的信息，xxxxxxxx 表示对象地址。

```
kd> !handle 1a0
processor number 0, process 808a0000
PROCESS 808a0000  SessionId: none  Cid: 0000    Peb: 00000000  ParentCid: 0000
    DirBase: 00039000  ObjectTable: e1000d50  HandleCount: 203.
    Image: Idle

Handle table at e1002000 with 203 Entries in use
01a0: Object: 81f130d0  GrantedAccess: 00140003 Entry: e1002340
Object: 81f130d0  Type: (81fce040) File
    ObjectHeader: 81f130b8 (old version)
        HandleCount: 1  PointerCount: 2
        Directory Object: 00000000  Name: \pagefile.sys {HarddiskVolume1}

kd> !object 81f130d0
Object: 81f130d0  Type: (81fce040) File
    ObjectHeader: 81f130b8 (old version)
    HandleCount: 1  PointerCount: 2
    Directory Object: 00000000  Name: \pagefile.sys {HarddiskVolume1}
```

!object xxxx\yyyy\zzzz 显示指定路径的对象的信息。

```
kd> !object \
Object: e1000868  Type: (81fd05d0) Directory
    ObjectHeader: e1000850 (old version)
    HandleCount: 0  PointerCount: 40
    Directory Object: 00000000  Name: \

    Hash Address  Type         Name
    ---- -------  ----         ----
     00  e1007b88 Directory    ArcName
         81eb1758 Device       Ntfs
     01  e1627aa8 Port         SeLsaCommandPort
     ......
     35  e12b90e0 Directory    KnownDlls
     36  81fb1e20 Device       DfsServer
kd> !object \Driver
Object: e1006708  Type: (81fd05d0) Directory
    ObjectHeader: e10066f0 (old version)
    HandleCount: 0  PointerCount: 65
    Directory Object: e1000868  Name: Driver

    Hash Address  Type         Name
    ---- -------  ----         ----
```

```
    00  81f06328 Driver       Beep
        81fb1410 Driver       NDIS
        81fb1620 Driver       KSecDD
    ......
        81f0b778 Driver       i8042prt
        81f6b598 Driver       IntelIde
kd> !object \Driver\Tcpip
Object: 81d68a38  Type: (81f933b0) Driver
    ObjectHeader: 81d68a20 (old version)
    HandleCount: 0  PointerCount: 7
    Directory Object: e1006708  Name: Tcpip
```

驱动设备相关

!drvobj

!drvobj 命令显示驱动信息，主要是显示 DRIVER_OBJECT 结构的信息。

!drvobj xxxxxxxx yy 命令显示指定驱动的信息，xxxxxxxx 可以指定为 DRIVER_OBJECT 结构的地址，也可以指定驱动名，yy 表示显示掩码。

```
lkd> !object \Driver
Object: 88a6ae30  Type: (84042ce0) Directory
    ObjectHeader: 88a6ae18 (old version)
    HandleCount: 0  PointerCount: 106
    Directory Object: 88a034d0  Name: Driver

    Hash Address  Type         Name
    ---- -------  ----         ----
     00  84f18b18 Driver       KSecDD
         84f02d50 Driver       NDIS
     ......
     20  848659f0 Driver       VlkdKnl
     ......
     36  84b20a28 Driver       PROCEXP113
         85dd9f38 Driver       i8042prt
lkd> !drvobj 848659f0 7
Driver object (848659f0) is for:
 \Driver\VlkdKnl
Driver Extension List: (id , addr)

Device Object list:
86f29d88

DriverEntry:   a98b1385 VlkdKnl!GsDriverEntry
DriverStartIo: 00000000
DriverUnload:  a98b080e VlkdKnl!OnUnload
AddDevice:     00000000
```

```
Dispatch routines:
[00] IRP_MJ_CREATE                    a98b07ea   VlkdKnl!DispatchCreateClose
[01] IRP_MJ_CREATE_NAMED_PIPE          81e39fdf nt!IopInvalidDeviceRequest
[02] IRP_MJ_CLOSE                     a98b07ea   VlkdKnl!DispatchCreateClose
......
[0e] IRP_MJ_DEVICE_CONTROL             a98b0788 VlkdKnl!DispatchIoctl
......
[1b] IRP_MJ_PNP                       81e39fdf   nt!IopInvalidDeviceRequest

lkd> !drvobj VlkdKnl 7
Driver object (848659f0) is for:
 \Driver\VlkdKnl
Driver Extension List: (id , addr)

Device Object list:
86f29d88

DriverEntry:   a98b1385 VlkdKnl!GsDriverEntry
DriverStartIo: 00000000
DriverUnload:  a98b080e VlkdKnl!OnUnload
AddDevice:     00000000

Dispatch routines:
[00] IRP_MJ_CREATE                    a98b07ea   VlkdKnl!DispatchCreateClose
[01] IRP_MJ_CREATE_NAMED_PIPE          81e39fdf nt!IopInvalidDeviceRequest
[02] IRP_MJ_CLOSE                     a98b07ea   VlkdKnl!DispatchCreateClose
......
[0e] IRP_MJ_DEVICE_CONTROL             a98b0788 VlkdKnl!DispatchIoctl
......
[1b] IRP_MJ_PNP                       81e39fdf   nt!IopInvalidDeviceRequest
```

!devobj

!devobj 命令显示驱动设备信息，主要是显示 DEVICE_OBJECT 结构信息。

!devobj xxxxxxxx 命令显示指定驱动设备的信息，xxxxxxxx 可以指定为 DEVICE_OBJECT 结构的地址，也可以指定驱动设备名。

```
lkd> !drvobj VlkdKnl
Driver object (848659f0) is for:
 \Driver\VlkdKnl
Driver Extension List: (id , addr)

Device Object list:
86f29d88
lkd> !devobj 86f29d88
Device object (86f29d88) is for:
```

```
 VlkdKnl \Driver\VlkdKnl DriverObject 848659f0
Current Irp 00000000 RefCount 1 Type 00000022 Flags 00000040
Dacl 88a69d60 DevExt 00000000 DevObjExt 86f29e40
ExtensionFlags (0x00000800)
                        Unknown flags 0x00000800
Device queue is not busy.
```

蓝屏 Dump 相关

分析蓝屏 dump 文件是非常有挑战的事情，因为蓝屏经常在内核被破坏了一段时间之后发生，要找到出错时的原因已经很难了。分析时用到的命令也没有标准可循，基本上是尽量多查看蓝屏时的状态，比如驱动列表、Hook 情况、内存占用情况、IRQL、本驱动状态、进程列表、堆栈情况、线程等待情况、IRP 等，希望从中能找到一点线索，然后继续分析。

!analyze -v

!analyze -v 命令是分析蓝屏 dump 时首先应该执行的命令，该命令会显示蓝屏原因、蓝屏上下文等，并会根据蓝屏原因自动进行相关的分析，给出进一步分析的建议。

```
kd> !analyze -v
*******************************************************************************
*                                                                             *
*                        Bugcheck Analysis                                    *
*                                                                             *
*******************************************************************************

CRITICAL_OBJECT_TERMINATION (f4)
A process or thread crucial to system operation has unexpectedly exited or been
terminated.
Several processes and threads are necessary for the operation of the
system; when they are terminated (for any reason), the system can no
longer function.
Arguments:
Arg1: 00000003, Process
Arg2: 81f0e020, Terminating object
Arg3: 81f0e184, Process image file name
Arg4: 8095aa0e, Explanatory message (ascii)

Debugging Details:
------------------

Page 1e167 not present in the dump file. Type ".hh dbgerr004" for details
Page 1dffc not present in the dump file. Type ".hh dbgerr004" for details
PEB is paged out (Peb.Ldr = 7ffde00c).  Type ".hh dbgerr001" for details
PEB is paged out (Peb.Ldr = 7ffde00c).  Type ".hh dbgerr001" for details

PROCESS_OBJECT: 81f0e020
```

```
IMAGE_NAME:  csrss.exe

DEBUG_FLR_IMAGE_TIMESTAMP:  0

MODULE_NAME: csrss

FAULTING_MODULE: 00000000

PROCESS_NAME:  procexp.exe

BUGCHECK_STR:  0xF4_procexp.exe

DEFAULT_BUCKET_ID:  DRIVER_FAULT

CURRENT_IRQL:  0

LAST_CONTROL_TRANSFER:  from 80996ecf to 80875d0e

STACK_TEXT:
f51ad7f0 80996ecf 000000f4 00000003 81f0e020 nt!KeBugCheckEx+0x1b
f51ad814 8095a9d2 8095aa0e 81f0e020 81f0e184 nt!PspCatchCriticalBreak+0x75
f51ad844 8082337b 000002d8 00000001 0012f800 nt!NtTerminateProcess+0x7a
f51ad844 7c95ed54 000002d8 00000001 0012f800 nt!KiFastCallEntry+0xf8
WARNING: Frame IP not in any known module. Following frames may be wrong.
0012f800 00000000 00000000 00000000 00000000 0x7c95ed54


STACK_COMMAND:  kb

FOLLOWUP_NAME:  MachineOwner

FAILURE_BUCKET_ID:  0xF4_procexp.exe_IMAGE_csrss.exe

BUCKET_ID:  0xF4_procexp.exe_IMAGE_csrss.exe

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

比如上面这个蓝屏，!analyze -v 命令分析出蓝屏错误码是 0xF4，蓝屏原因是关键对象被终止了。进一步分析出了是进程对象被结束了，并且知道是 csrss.exe 进程。接着下面显示蓝屏时刻的堆栈信息，从堆栈可以看出是因为关键进程被结束，系统主动调用 KeBugCheckEx 函数蓝屏的。然后再根据当前进程是 procexp.exe，可以知道这次蓝屏是因为 procexp.exe 进程把 csrss.exe 进程结束导致的。