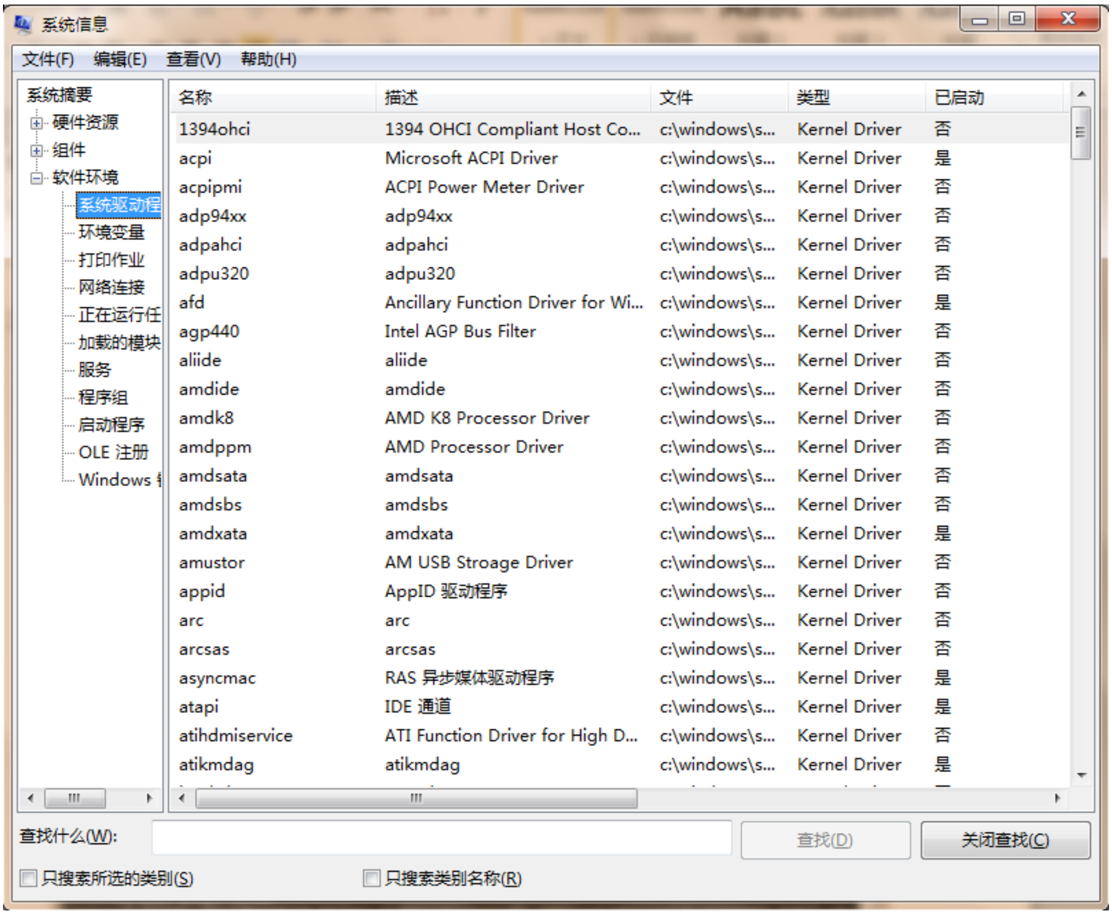
CS490 Windows Internals Labs

Oct 14th, 2013

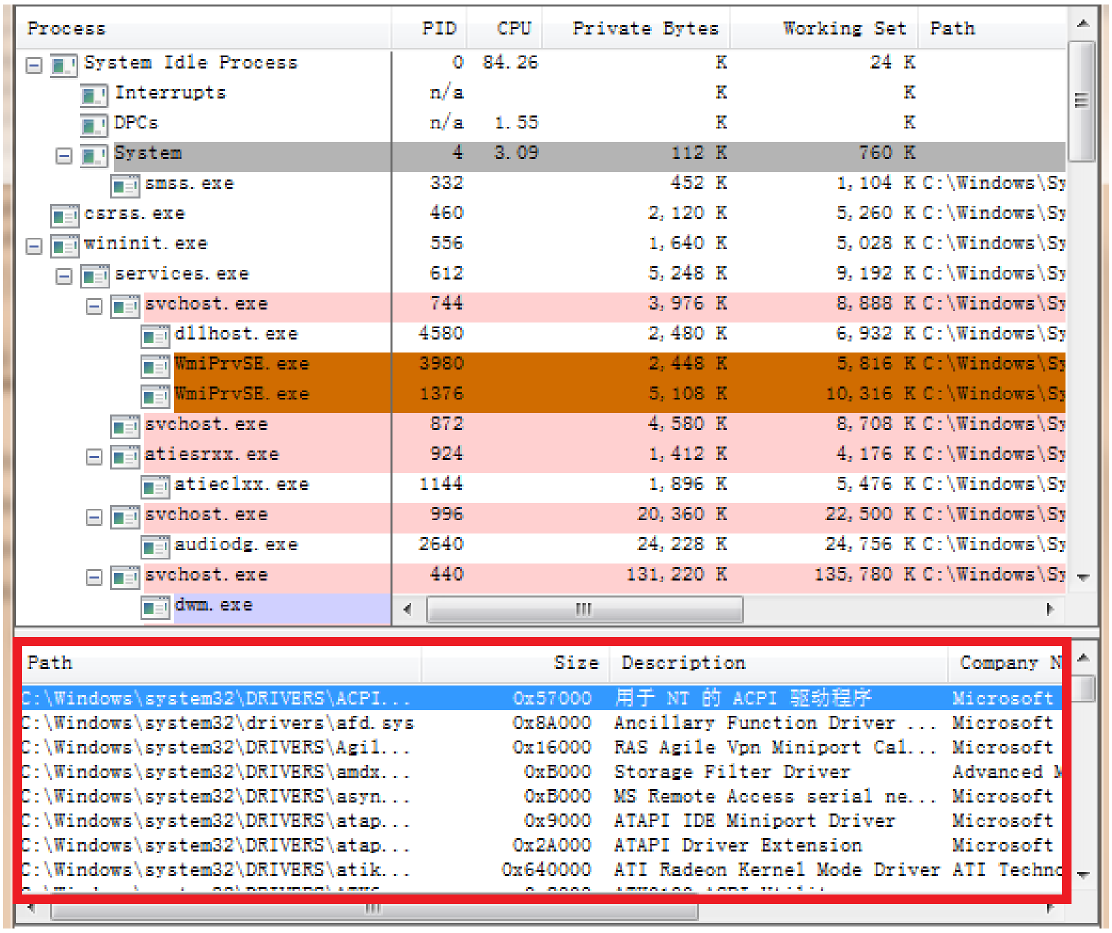
1. Viewing the Installed Driver List

In Windows XP/2003/vista/7, you can obtain the driver information by executing the **Msinfo32.exe** utility from the **Run** dialog box of the Start menu. Select the System Drivers entry under Software Environment to see the list of drivers configured on the system. Those that are loaded have the text "Yes" in the Started column.



名称	描述	文件	类型	已启动
1394ohci	1394 OHCI Compliant Host Co...	c:\windows\s...	Kernel Driver	否
acpi	Microsoft ACPI Driver	c:\windows\s...	Kernel Driver	是
acpipmi	ACPI Power Meter Driver	c:\windows\s...	Kernel Driver	否
adp94xx	adp94xx	c:\windows\s...	Kernel Driver	否
adpahci	adpahci	c:\windows\s...	Kernel Driver	否
adpu320	adpu320	c:\windows\s...	Kernel Driver	否
afd	Ancillary Function Driver for Wi...	c:\windows\s...	Kernel Driver	是
agp440	Intel AGP Bus Filter	c:\windows\s...	Kernel Driver	否
aliide	aliide	c:\windows\s...	Kernel Driver	否
amdide	amdide	c:\windows\s...	Kernel Driver	否
amdk8	AMD K8 Processor Driver	c:\windows\s...	Kernel Driver	否
amdppm	AMD Processor Driver	c:\windows\s...	Kernel Driver	否
amdsata	amdsata	c:\windows\s...	Kernel Driver	否
amdsbs	amdsbs	c:\windows\s...	Kernel Driver	否
amdxata	amdxata	c:\windows\s...	Kernel Driver	是
amustor	AM USB Stroage Driver	c:\windows\s...	Kernel Driver	否
appid	AppID 驱动程序	c:\windows\s...	Kernel Driver	否
arc	arc	c:\windows\s...	Kernel Driver	否
arcsas	arcsas	c:\windows\s...	Kernel Driver	否
asyncmac	RAS 异步媒体驱动程序	c:\windows\s...	Kernel Driver	是
atapi	IDE 通道	c:\windows\s...	Kernel Driver	是
atihdmiservice	ATI Function Driver for High D...	c:\windows\s...	Kernel Driver	否
atikmdag	atikmdag	c:\windows\s...	Kernel Driver	是

You can also view the list of loaded kernel-mode drivers with Process Explorer from www.sysinternals.com. Run Process Explorer, select the System process, and select DLLs from the Lower Pane menu entry in the View menu. Process Explorer lists the loaded drivers, their names, version information including company and description, and load address (assuming you have configured Process Explorer to display the corresponding columns).

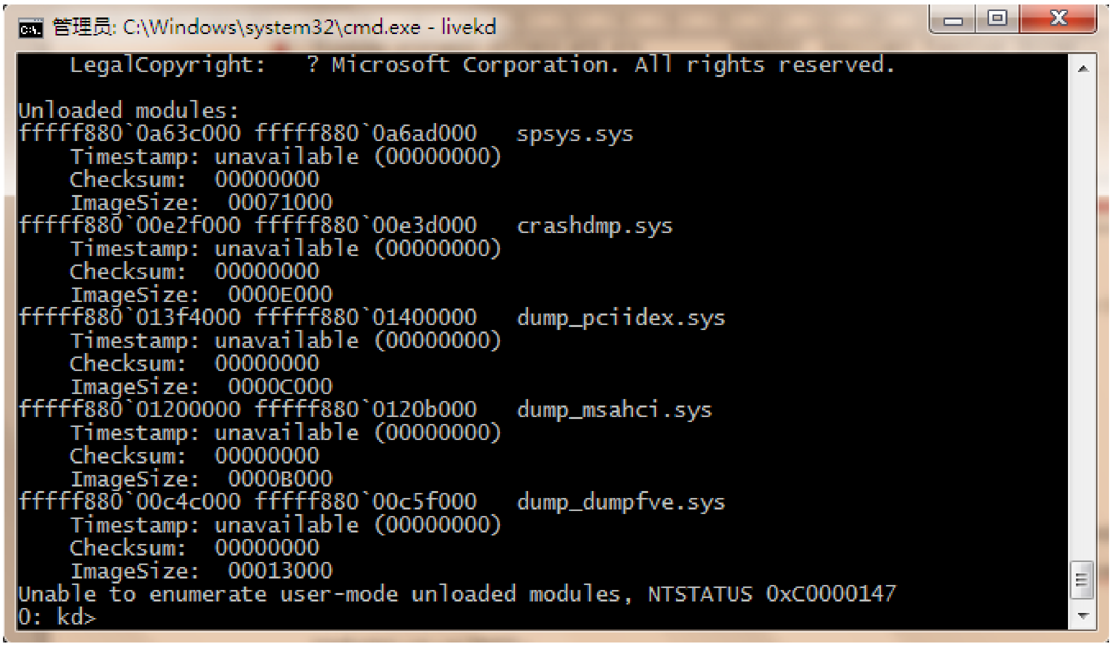


Process	PID	CPU	Private Bytes	Working Set	Path
System Idle Process	0	84. 26	K	24 K	
Interrupts	n/a		K	K	
DPCs	n/a	1. 55	K	K	
System	4	3. 09	112 K	760 K	
smss. exe	332		452 K	1, 104 K	C:\Windows\Sy
csrss. exe	460		2, 120 K	5, 260 K	C:\Windows\Sy
wininit. exe	556		1, 640 K	5, 028 K	C:\Windows\Sy
services. exe	612		5, 248 K	9, 192 K	C:\Windows\Sy
svchost. exe	744		3, 976 K	8, 888 K	C:\Windows\Sy
dllhost. exe	4580		2, 480 K	6, 932 K	C:\Windows\Sy
WmiPrvSE. exe	3980		2, 448 K	5, 816 K	C:\Windows\Sy
WmiPrvSE. exe	1376		5, 108 K	10, 316 K	C:\Windows\Sy
svchost. exe	872		4, 580 K	8, 708 K	C:\Windows\Sy
atiesrxx. exe	924		1, 412 K	4, 176 K	C:\Windows\Sy
atieclxx. exe	1144		1, 896 K	5, 476 K	C:\Windows\Sy
svchost. exe	996		20, 360 K	22, 500 K	C:\Windows\Sy
audiodg. exe	2640		24, 228 K	24, 756 K	C:\Windows\Sy
svchost. exe	440		131, 220 K	135, 780 K	C:\Windows\Sy
dwm. exe					

Path	Size	Description	Company N
C:\Windows\system32\DRIVERS\ACPI...	0x57000	用于 NT 的 ACPI 驱动程序	Microsoft
C:\Windows\system32\drivers\afd. sys	0x8A000	Ancillary Function Driver ...	Microsoft
C:\Windows\system32\DRIVERS\Agil...	0x16000	RAS Agile Vpn Miniport Cal...	Microsoft
C:\Windows\system32\DRIVERS\amdx...	0xB000	Storage Filter Driver	Advanced M
C:\Windows\system32\DRIVERS\asyn...	0xB000	MS Remote Access serial ne...	Microsoft
C:\Windows\system32\DRIVERS\atap...	0x9000	ATAPI IDE Miniport Driver	Microsoft
C:\Windows\system32\DRIVERS\atap...	0x2A000	ATAPI Driver Extension	Microsoft
C:\Windows\system32\DRIVERS\atik...	0x640000	ATI Radeon Kernel Mode Driver	ATI Techno

To view loaded driver, you can get a similar display with the kernel debugger lm kv command:

kd>lm kv

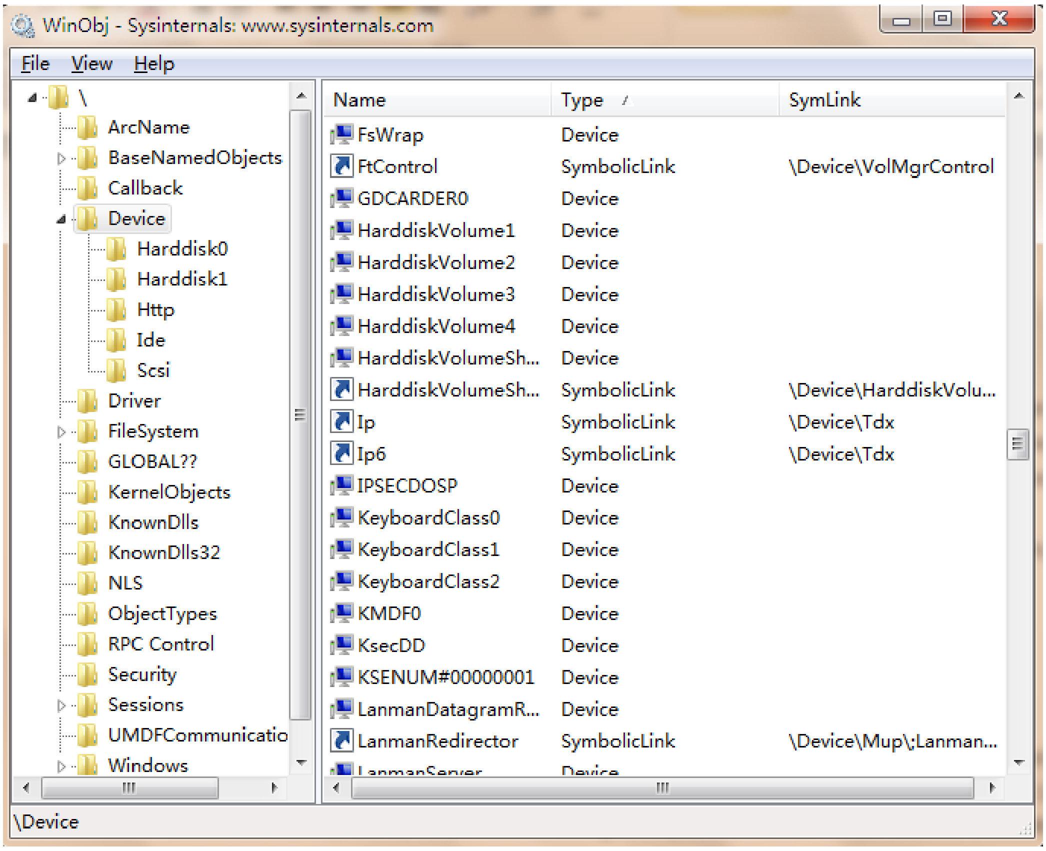


管理员: C:\Windows\system32\cmd.exe - livekd

```
    LegalCopyright:   ? Microsoft Corporation. All rights reserved.

Unloaded modules:
fffff880`0a63c000 fffff880`0a6ad000   spsys.sys
    Timestamp: unavailable (00000000)
    Checksum:  00000000
    ImageSize:  00071000
fffff880`00e2f000 fffff880`00e3d000   crashdmp.sys
    Timestamp: unavailable (00000000)
    Checksum:  00000000
    ImageSize:  0000E000
fffff880`013f4000 fffff880`01400000   dump_pciidex.sys
    Timestamp: unavailable (00000000)
    Checksum:  00000000
    ImageSize:  0000C000
fffff880`01200000 fffff880`0120b000   dump_msahci.sys
    Timestamp: unavailable (00000000)
    Checksum:  00000000
    ImageSize:  0000B000
fffff880`00c4c000 fffff880`00c5f000   dump_dumpfve.sys
    Timestamp: unavailable (00000000)
    Checksum:  00000000
    ImageSize:  00013000
Unable to enumerate user-mode unloaded modules, NTSTATUS 0xC0000147
0: kd>
```

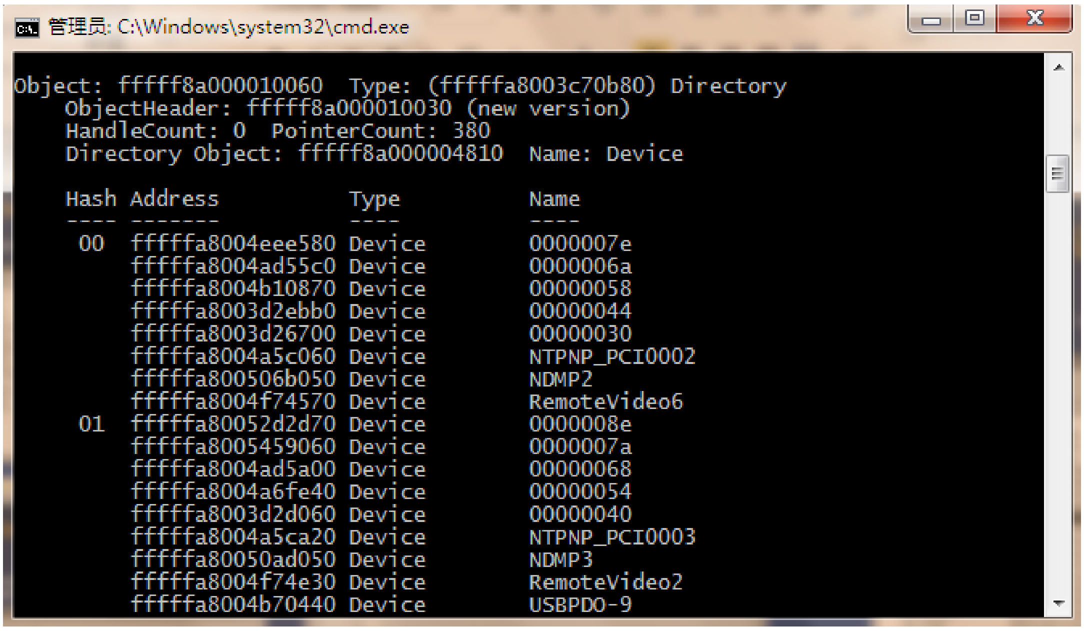
2. Viewing \Device Directory

You can use the Winobj tool from www.sysinternals.com or the !object kernel debugger command to view the device names under \Device in the object manager namespace. The following screen shot shows an I/O manager–assigned symbolic link that points to a device object in \Device with an auto-generated name.



Name	Type	SymLink
FsWrap	Device	
FtControl	SymbolicLink	\Device\VolMgrControl
GDCARDER0	Device	
HarddiskVolume1	Device	
HarddiskVolume2	Device	
HarddiskVolume3	Device	
HarddiskVolume4	Device	
HarddiskVolumeSh...	Device	
HarddiskVolumeSh...	SymbolicLink	\Device\HarddiskVolu...
Ip	SymbolicLink	\Device\Tdx
Ip6	SymbolicLink	\Device\Tdx
IPSECDOSP	Device	
KeyboardClass0	Device	
KeyboardClass1	Device	
KeyboardClass2	Device	
KMDF0	Device	
KsecDD	Device	
KSENUM#00000001	Device	
LanmanDatagramR...	Device	
LanmanRedirector	SymbolicLink	\Device\Mup\;Lanman...
LanmanServer	Device	

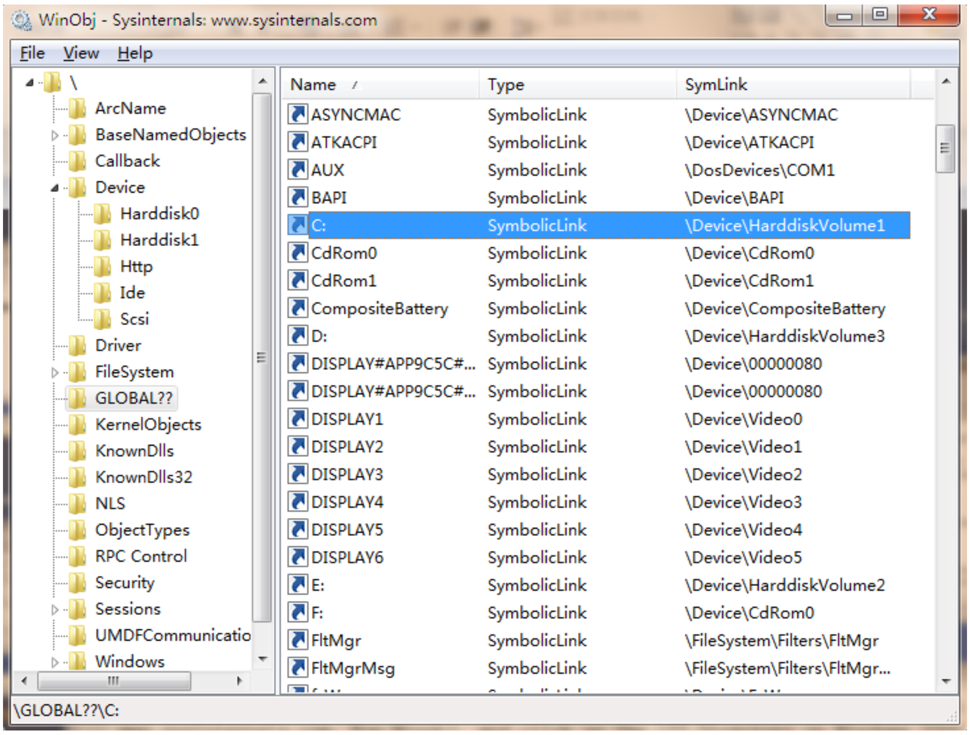
When you run the !object kernel debugger command and specify the \Device directory, you should see output similar to the following:



```
Object: fffff8a000010060  Type: (fffffa8003c70b80) Directory
    ObjectHeader: fffff8a000010030 (new version)
    HandleCount: 0  PointerCount: 380
    Directory Object: fffff8a000004810  Name: Device

    Hash Address          Type          Name
    ---- --------         ----          ----
     00  fffffa8004eee580 Device        0000007e
         fffffa8004ad55c0 Device        0000006a
         fffffa8004b10870 Device        00000058
         fffffa8003d2ebb0 Device        00000044
         fffffa8003d26700 Device        00000030
         fffffa8004a5c060 Device        NTPNP_PCI0002
         fffffa800506b050 Device        NDMP2
         fffffa8004f74570 Device        RemoteVideo6
     01  fffffa80052d2d70 Device        0000008e
         fffffa8005459060 Device        0000007a
         fffffa8004ad5a00 Device        00000068
         fffffa8004a6fe40 Device        00000054
         fffffa8003d2d060 Device        00000040
         fffffa8004a5ca20 Device        NTPNP_PCI0003
         fffffa80050ad050 Device        NDMP3
         fffffa8004f74e30 Device        RemoteVideo2
         fffffa8004b70440 Device        USBPDO-9
```

3. Device Name Mappings

You can examine the symbolic links that define the Windows device namespace with the Winobj utility from www.sysinternals.com. Run Winobj, and click on the **\Global??** on Windows XP or later version. Notice the symbolic links on the right. Try double-clicking on the device C:. C: is a symbolic link to the internal device named \Device\HarddiskVolume1, or the first volume on the first hard drive in the system.

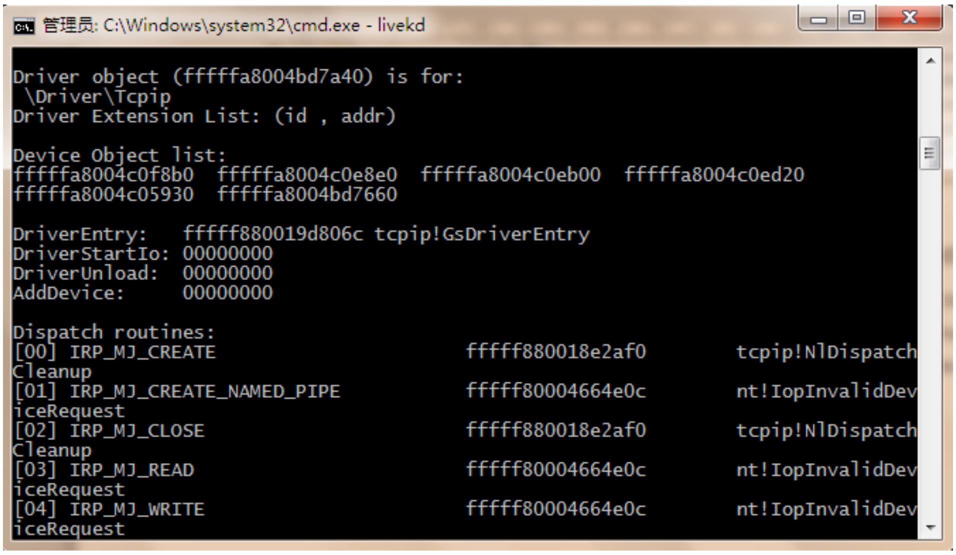


Name	Type	SymLink
ASYNCMAC	SymbolicLink	\Device\ASYNCMAC
ATKACPI	SymbolicLink	\Device\ATKACPI
AUX	SymbolicLink	\DosDevices\COM1
BAPI	SymbolicLink	\Device\BAPI
C:	SymbolicLink	\Device\HarddiskVolume1
CdRom0	SymbolicLink	\Device\CdRom0
CdRom1	SymbolicLink	\Device\CdRom1
CompositeBattery	SymbolicLink	\Device\CompositeBattery
D:	SymbolicLink	\Device\HarddiskVolume3
DISPLAY#APP9C5C#...	SymbolicLink	\Device\00000080
DISPLAY#APP9C5C#...	SymbolicLink	\Device\00000080
DISPLAY1	SymbolicLink	\Device\Video0
DISPLAY2	SymbolicLink	\Device\Video1
DISPLAY3	SymbolicLink	\Device\Video2
DISPLAY4	SymbolicLink	\Device\Video3
DISPLAY5	SymbolicLink	\Device\Video4
DISPLAY6	SymbolicLink	\Device\Video5
E:	SymbolicLink	\Device\HarddiskVolume2
F:	SymbolicLink	\Device\CdRom0
FltMgr	SymbolicLink	\FileSystem\Filters\FltMgr
FltMgrMsg	SymbolicLink	\FileSystem\Filters\FltMgr...

\GLOBAL??\C:

4. Viewing the TCP/IP Driver Object and its Device Objects

Using the kernel debugger to look at a live system, you can examine TCP/IP's device objects. After performing the !drvobj command to see the addresses of each of the driver's device objects, execute !devobj tcpip 7 to view the name and other details about the device object.

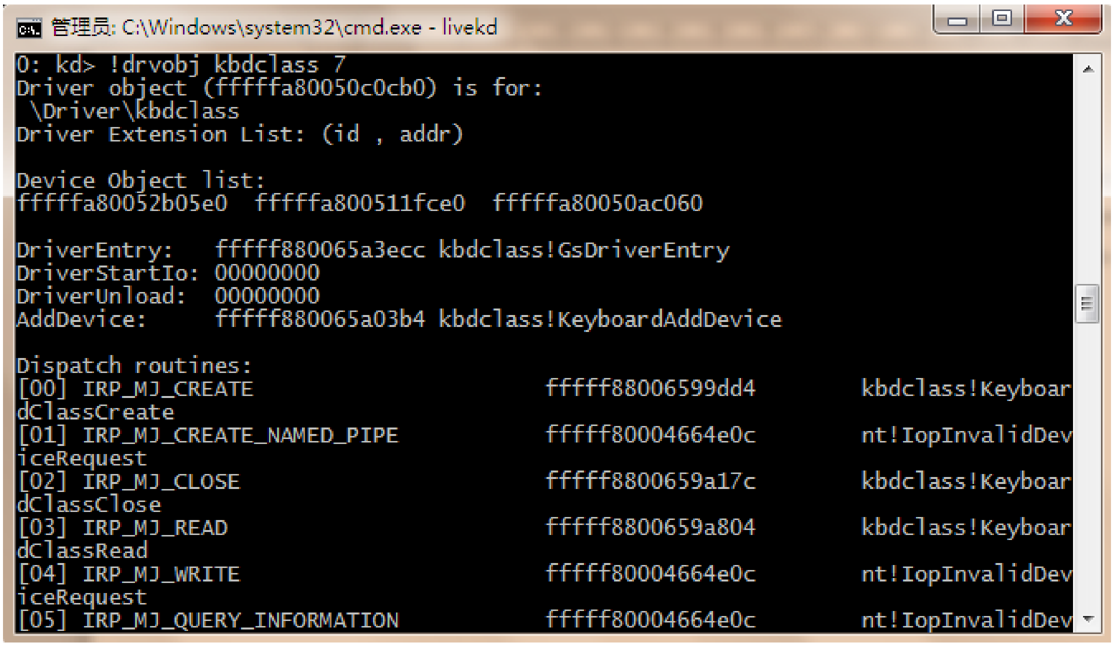


```
Driver object (fffffa8004bd7a40) is for:
 \Driver\Tcpip
Driver Extension List: (id , addr)

Device Object list:
fffffa8004c0f8b0  fffffa8004c0e8e0  fffffa8004c0eb00  fffffa8004c0ed20
fffffa8004c05930  fffffa8004bd7660

DriverEntry:   fffff880019d806c tcpip!GsDriverEntry
DriverStartIo: 00000000
DriverUnload:  00000000
AddDevice:     00000000

Dispatch routines:
[00] IRP_MJ_CREATE                      fffff880018e2af0        tcpip!NlDispatch
Cleanup
[01] IRP_MJ_CREATE_NAMED_PIPE           fffff80004664e0c        nt!IopInvalidDev
iceRequest
[02] IRP_MJ_CLOSE                       fffff880018e2af0        tcpip!NlDispatch
Cleanup
[03] IRP_MJ_READ                        fffff80004664e0c        nt!IopInvalidDev
iceRequest
[04] IRP_MJ_WRITE                       fffff80004664e0c        nt!IopInvalidDev
iceRequest
```

5. Looking at Driver's Dispatch Routines

You can obtain a listing of the functions a driver has defined for its dispatch routines by entering a 7 after the driver object's name (or address) in the !drvobj kernel debugger command. The following output shows that drivers support 28 IRP types.

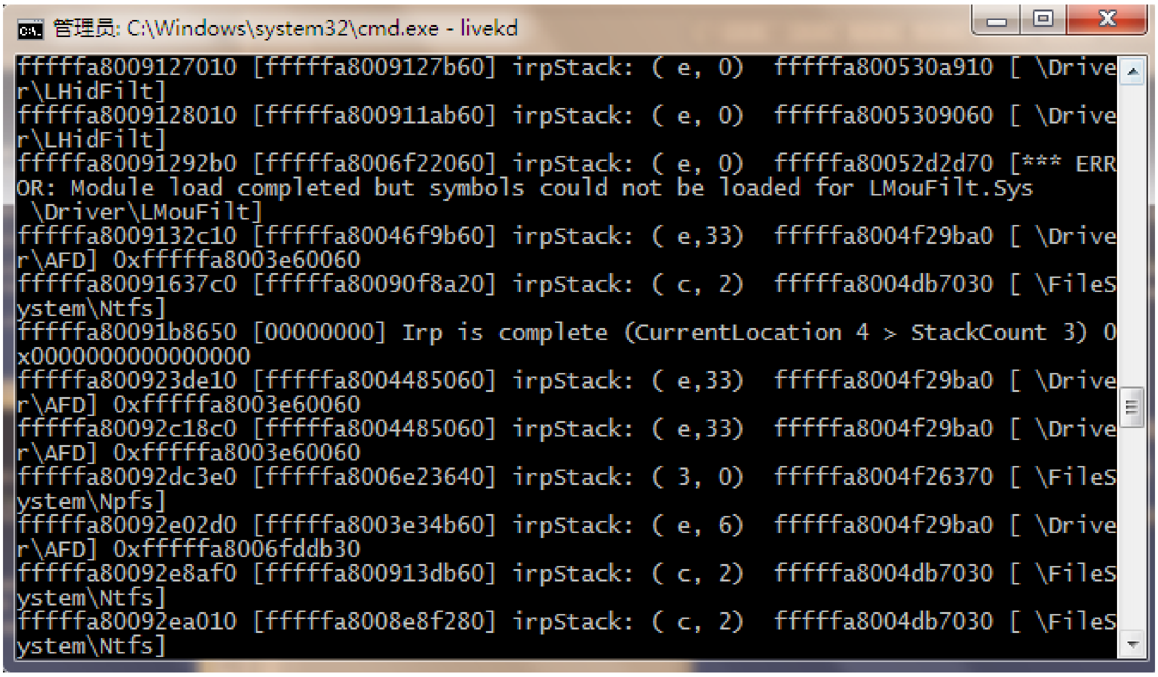
Kd>!drvobj kbdclass 7



```
0: kd> !drvobj kbdclass 7
Driver object (fffffa80050c0cb0) is for:
 \Driver\kbdclass
Driver Extension List: (id , addr)

Device Object list:
fffffa80052b05e0  fffffa800511fce0  fffffa80050ac060

DriverEntry:   fffff880065a3ecc kbdclass!GsDriverEntry
DriverStartIo: 00000000
DriverUnload:  00000000
AddDevice:     fffff880065a03b4 kbdclass!KeyboardAddDevice

Dispatch routines:
[00] IRP_MJ_CREATE                      fffff88006599dd4        kbdclass!Keyboar
dClassCreate
[01] IRP_MJ_CREATE_NAMED_PIPE           fffff80004664e0c        nt!IopInvalidDev
iceRequest
[02] IRP_MJ_CLOSE                       fffff8800659a17c        kbdclass!Keyboar
dClassClose
[03] IRP_MJ_READ                        fffff8800659a804        kbdclass!Keyboar
dClassRead
[04] IRP_MJ_WRITE                       fffff80004664e0c        nt!IopInvalidDev
iceRequest
[05] IRP_MJ_QUERY_INFORMATION           fffff80004664e0c        nt!IopInvalidDev
```

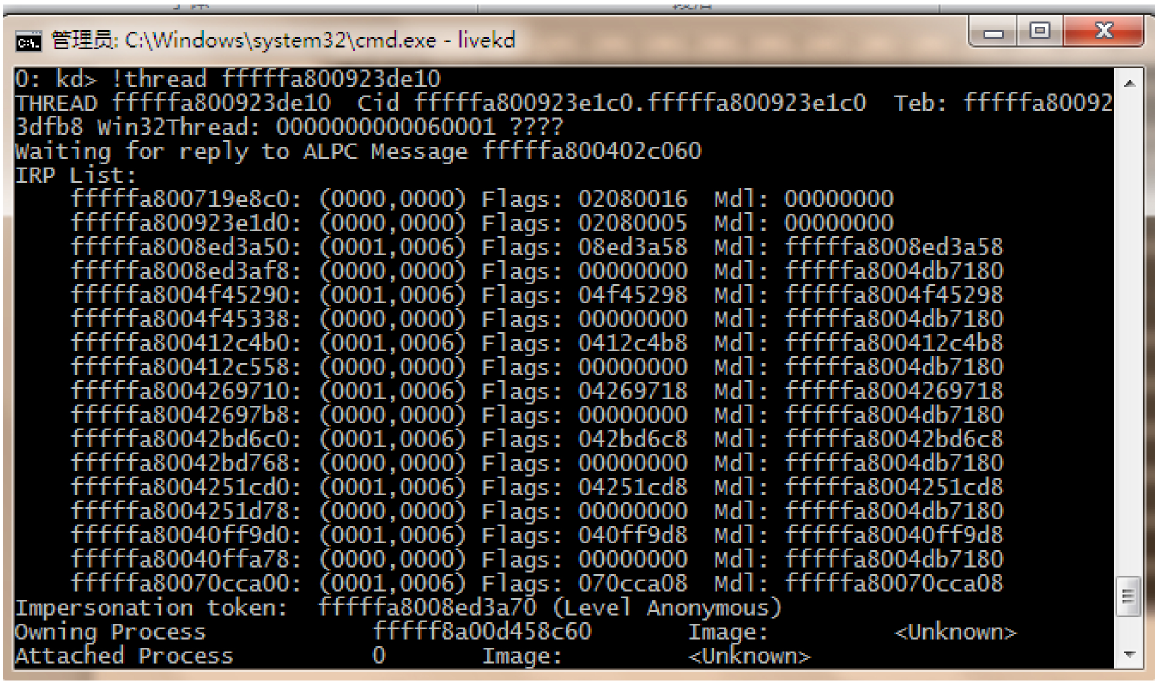
6. Find an IRP

In this experiment, you'll find an uncompleted IRP on the system, and you'll determine the IRP type, the device at which it's directed, the driver that manages the device, the thread that issued the IRP, and what process the thread belongs to. At any point in time, there are at least a few uncompleted IRPs on a system. This is because there are many devices to which applications can issue IRPs that a driver will only complete when a particular event occurs, such as data becoming available. One example is a blocking read from a network endpoint. You can see the outstanding IRPs on a system with the !irpfind kernel debugger command:



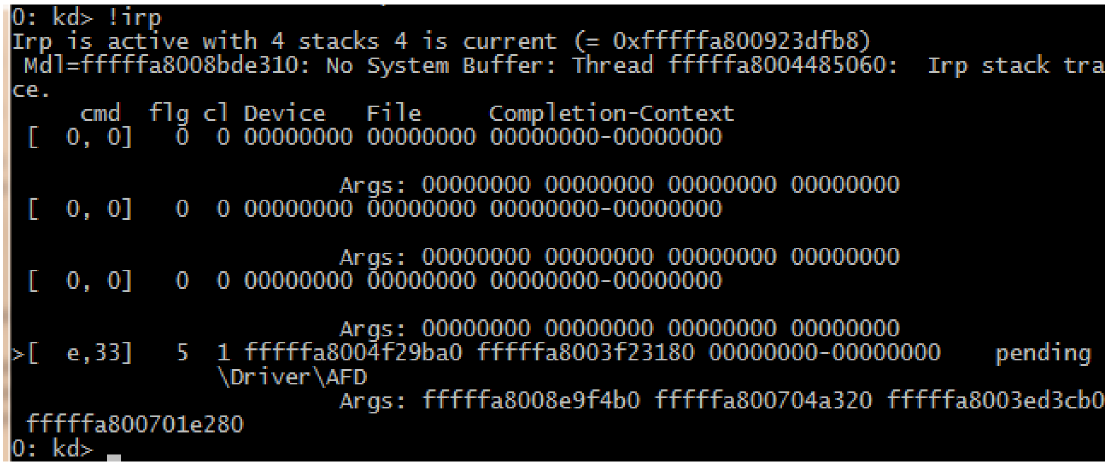
```
fffffa8009127010 [fffffa8009127b60] irpStack: ( e, 0)  fffffa800530a910 [ \Drive
r\LHidFilt]
fffffa8009128010 [fffffa800911ab60] irpStack: ( e, 0)  fffffa8005309060 [ \Drive
r\LHidFilt]
fffffa80091292b0 [fffffa8006f22060] irpStack: ( e, 0)  fffffa80052d2d70 [*** ERR
OR: Module load completed but symbols could not be loaded for LMouFilt.Sys
 \Driver\LMouFilt]
fffffa8009132c10 [fffffa80046f9b60] irpStack: ( e,33)  fffffa8004f29ba0 [ \Drive
r\AFD] 0xfffffa8003e60060
fffffa80091637c0 [fffffa80090f8a20] irpStack: ( c, 2)  fffffa8004db7030 [ \FileS
ystem\Ntfs]
fffffa80091b8650 [00000000] Irp is complete (CurrentLocation 4 > StackCount 3) 0
x0000000000000000
fffffa800923de10 [fffffa8004485060] irpStack: ( e,33)  fffffa8004f29ba0 [ \Drive
r\AFD] 0xfffffa8003e60060
fffffa80092c18c0 [fffffa8004485060] irpStack: ( e,33)  fffffa8004f29ba0 [ \Drive
r\AFD] 0xfffffa8003e60060
fffffa80092dc3e0 [fffffa8006e23640] irpStack: ( 3, 0)  fffffa8004f26370 [ \FileS
ystem\Npfs]
fffffa80092e02d0 [fffffa8003e34b60] irpStack: ( e, 6)  fffffa8004f29ba0 [ \Drive
r\AFD] 0xfffffa8006fddb30
fffffa80092e8af0 [fffffa800913db60] irpStack: ( c, 2)  fffffa8004db7030 [ \FileS
ystem\Ntfs]
fffffa80092ea010 [fffffa8008e8f280] irpStack: ( c, 2)  fffffa8004db7030 [ \FileS
ystem\Ntfs]
```

When you use the !thread command, it prints any IRPs associated with the thread.



```
0: kd> !thread fffffa800923de10
THREAD fffffa800923de10  Cid fffffa800923e1c0.fffffa800923e1c0  Teb: fffffa80092
3dfb8 Win32Thread: 0000000000060001 ????
Waiting for reply to ALPC Message fffffa800402c060
IRP List:
    fffffa800719e8c0: (0000,0000) Flags: 02080016  Mdl: 00000000
    fffffa800923e1d0: (0000,0000) Flags: 02080005  Mdl: 00000000
    fffffa8008ed3a50: (0001,0006) Flags: 08ed3a58  Mdl: fffffa8008ed3a58
    fffffa8008ed3af8: (0000,0000) Flags: 00000000  Mdl: fffffa8004db7180
    fffffa8004f45290: (0001,0006) Flags: 04f45298  Mdl: fffffa8004f45298
    fffffa8004f45338: (0000,0000) Flags: 00000000  Mdl: fffffa8004db7180
    fffffa800412c4b0: (0001,0006) Flags: 0412c4b8  Mdl: fffffa800412c4b8
    fffffa800412c558: (0000,0000) Flags: 00000000  Mdl: fffffa8004db7180
    fffffa8004269710: (0001,0006) Flags: 04269718  Mdl: fffffa8004269718
    fffffa80042697b8: (0000,0000) Flags: 00000000  Mdl: fffffa8004db7180
    fffffa80042bd6c0: (0001,0006) Flags: 042bd6c8  Mdl: fffffa80042bd6c8
    fffffa80042bd768: (0000,0000) Flags: 00000000  Mdl: fffffa8004db7180
    fffffa8004251cd0: (0001,0006) Flags: 04251cd8  Mdl: fffffa8004251cd8
    fffffa8004251d78: (0000,0000) Flags: 00000000  Mdl: fffffa8004db7180
    fffffa80040ff9d0: (0001,0006) Flags: 040ff9d8  Mdl: fffffa80040ff9d8
    fffffa80040ffa78: (0000,0000) Flags: 00000000  Mdl: fffffa8004db7180
    fffffa80070cca00: (0001,0006) Flags: 070cca08  Mdl: fffffa80070cca08
Impersonation token:  fffffa8008ed3a70 (Level Anonymous)
Owning Process            fffff8a00d458c60       Image:         <Unknown>
Attached Process          0       Image:         <Unknown>
```

If you want to see the current IRP, use !irp after you scan the IRPs by using !irpfind. You can get result similar to the following screenshot.



```
0: kd> !irp
Irp is active with 4 stacks 4 is current (= 0xfffffa800923dfb8)
 Mdl=fffffa8008bde310: No System Buffer: Thread fffffa8004485060:  Irp stack tra
ce.
     cmd  flg cl Device   File     Completion-Context
 [  0, 0]   0  0 00000000 00000000 00000000-00000000

                        Args: 00000000 00000000 00000000 00000000
 [  0, 0]   0  0 00000000 00000000 00000000-00000000

                        Args: 00000000 00000000 00000000 00000000
 [  0, 0]   0  0 00000000 00000000 00000000-00000000

                        Args: 00000000 00000000 00000000 00000000
>[  e,33]   5  1 fffffa8004f29ba0 fffffa8003f23180 00000000-00000000    pending
               \Driver\AFD
                        Args: fffffa8008e9f4b0 fffffa800704a320 fffffa8003ed3cb0
 fffffa800701e280
0: kd> 
```