WinDbg

Guangming Zhu

AGENDA

- Build Drivers
- Setup Windbg Environment
- Some Problems
- WinDbg Command

Build Drivers

Build drivers

1. Connect to shswp4p1:1665 and download the dk files to local by Pv4.
2. Download the drive code by Github.

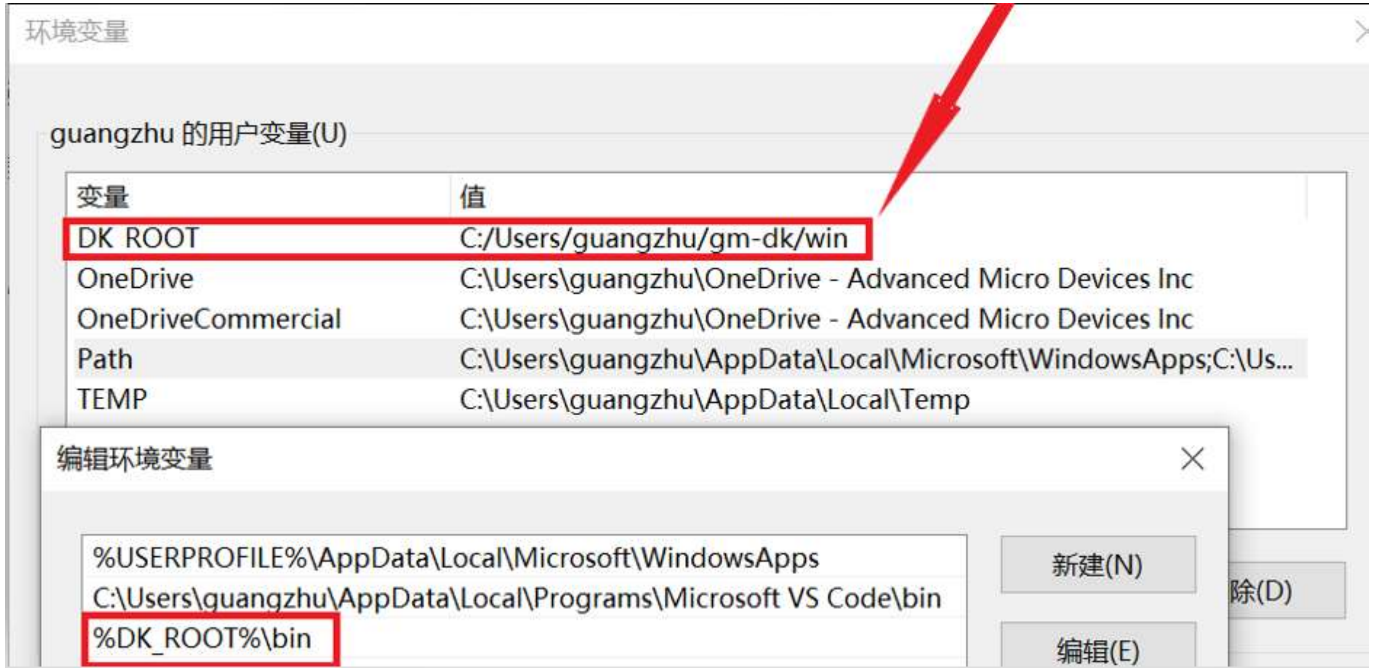
```
git clone <address>
git submodule update --init --recursive
```

3. Configure content in environment variables.
4. Execute ‘make’ command in drivers/drivers/kmd by cmd.exe.

```
make -j16 dbg.wnow64a
```

5. Get build files in drivers\drivers\kmd\build\.

Use ‘/’



guangzhu 的用户变量(U)

变量	值
DK ROOT	C:/Users/guangzhu/gm-dk/win
OneDrive	C:\Users\guangzhu\OneDrive - Advanced Micro Devices Inc
OneDriveCommercial	C:\Users\guangzhu\OneDrive - Advanced Micro Devices Inc
Path	C:\Users\guangzhu\AppData\Local\Microsoft\WindowsApps;C:\Us...
TEMP	C:\Users\guangzhu\AppData\Local\Temp

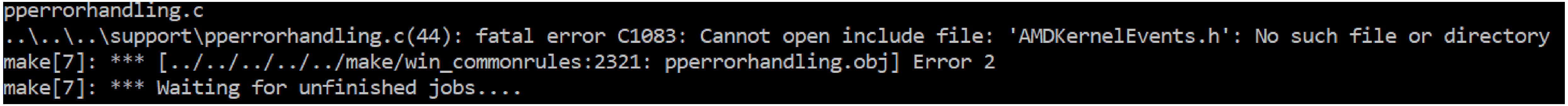
编辑环境变量

```
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\guangzhu\AppData\Local\Programs\Microsoft VS Code\bin
%DK_ROOT%\bin
```

Problem --- Can’t find ‘AMDKernelEvents.h’

If there is a problem that missing files ‘AMDKernelEvents.h’, then download depot/dk.pre/win/ms_sdk/n20246 and

depot/dk.pre/win/ms_wdk/n20246 to local.



```
pperrorhandling.c
..\..\..\support\pperrorhandling.c(44): fatal error C1083: Cannot open include file: 'AMDKernelEvents.h': No such file or directory
make[7]: *** [../../../../../make/win_commonrules:2321: pperrorhandling.obj] Error 2
make[7]: *** Waiting for unfinished jobs....
```

Setup Windbg Environment

1.1 Connect USB cable

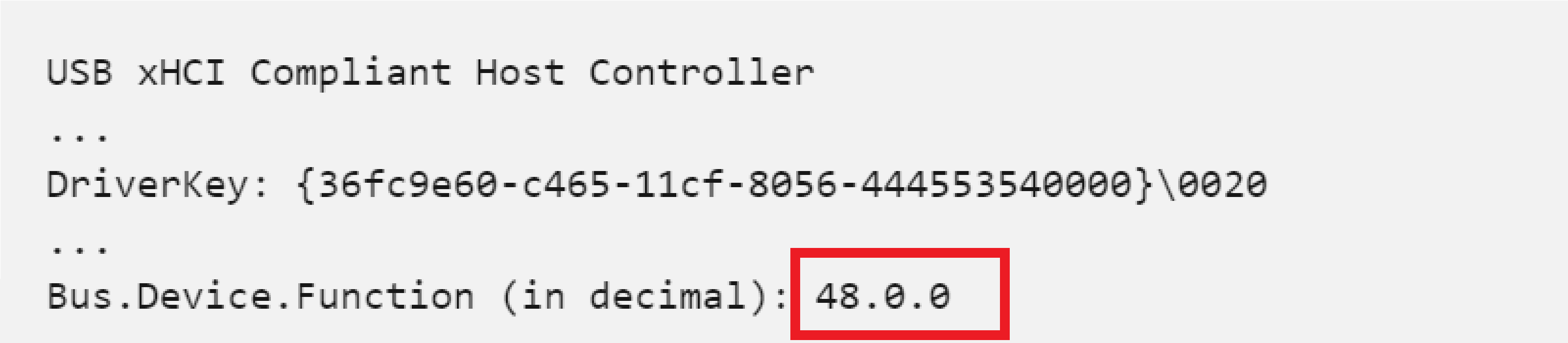
Required:

an xHCI (USB 3.0) host controller on host computer

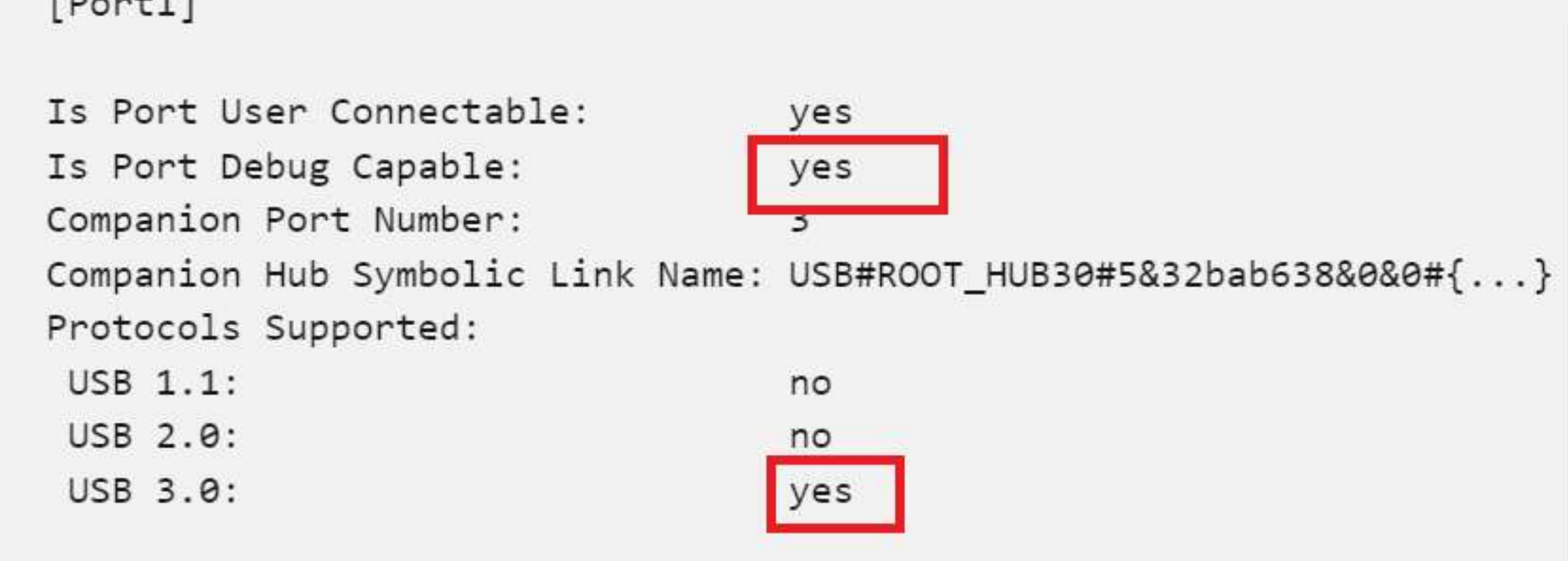
an xHCI (USB 3.0) host controller on target computer that supports debugging

a USB 3.0 cable

USBviewer



```
USB xHCI Compliant Host Controller
...
DriverKey: {36fc9e60-c465-11cf-8056-444553540000}\0020
...
Bus.Device.Function (in decimal): 48.0.0
```



```
[Port1]

Is Port User Connectable:         yes
Is Port Debug Capable:            yes
Companion Port Number:            3
Companion Hub Symbolic Link Name: USB#ROOT_HUB30#5&32bab638&0&0#{...}
Protocols Supported:
 USB 1.1:                         no
 USB 2.0:                         no
 USB 3.0:                         yes
```

1.1 Connect USB cable

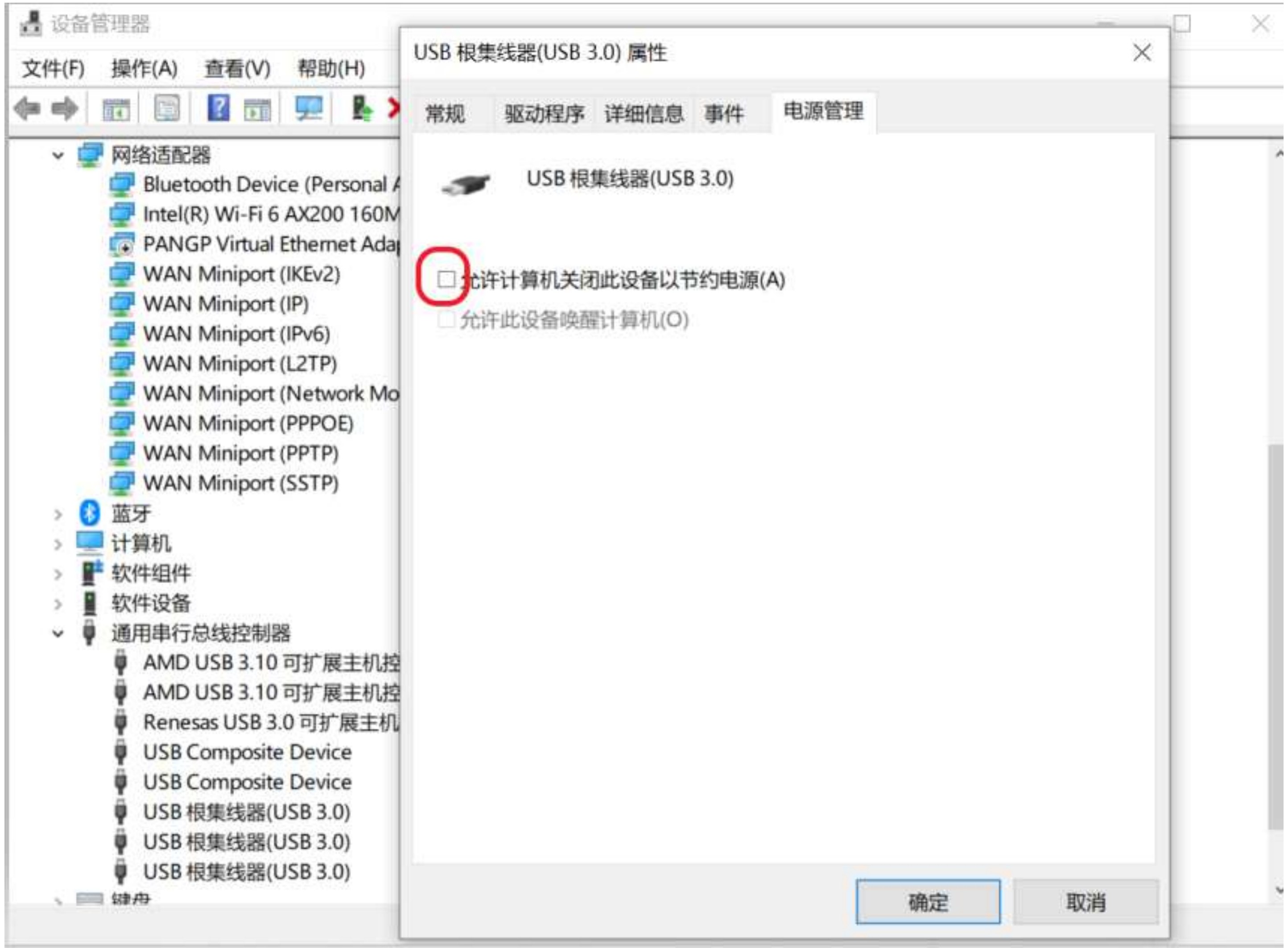
On the target computer, open a Command Prompt window as Administrator, and enter these commands:

- bcdedit /debug on
- bcdedit /dbgsettings usb targetname:TargetName
- bcdedit /set "{dbgsettings}" busparams <Bus.Device.Function>

Reboot the target computer.

1.1 Connect USB cable

In some cases, power transitions can interfere with debugging over USB 3.0. To avoid these problems, disable selective suspend for the xHCI host controller (and its root hub) that you are using for debugging.



USB 根集线器(USB 3.0)

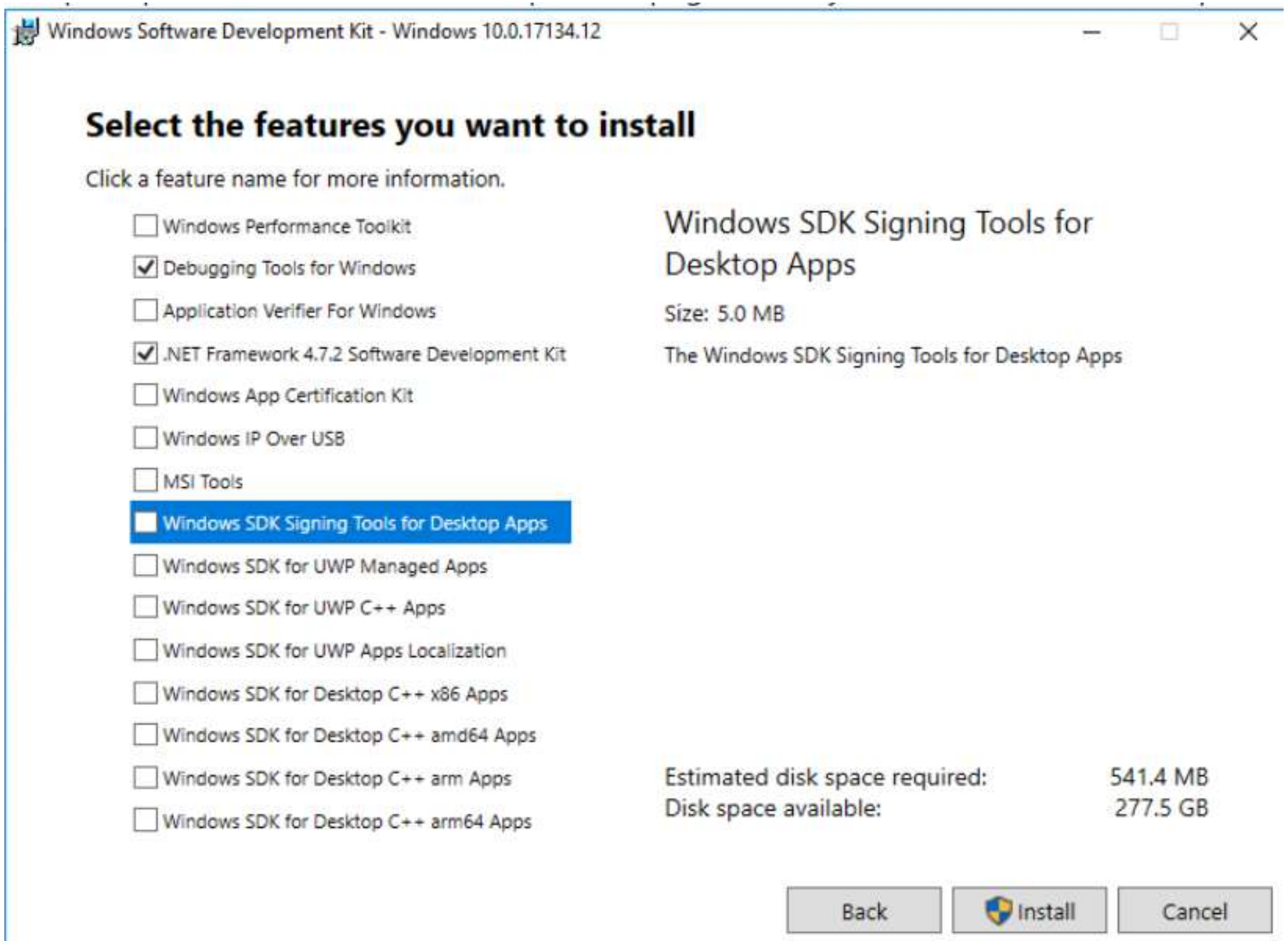
☐ 允许计算机关闭此设备以节约电源(A)

☐ 允许此设备唤醒计算机(O)

1.1 Install Windbg

Download the Windows 10 SDK:

https://developer.microsoft.com/en-us/windows/downloads/windows-10-sdk



Windows Software Development Kit - Windows 10.0.17134.12

Select the features you want to install

Click a feature name for more information.

- □ Windows Performance Toolkit
- ☑ Debugging Tools for Windows
- □ Application Verifier For Windows
- ☑ .NET Framework 4.7.2 Software Development Kit
- □ Windows App Certification Kit
- □ Windows IP Over USB
- □ MSI Tools
- □ Windows SDK Signing Tools for Desktop Apps
- □ Windows SDK for UWP Managed Apps
- □ Windows SDK for UWP C++ Apps
- □ Windows SDK for UWP Apps Localization
- □ Windows SDK for Desktop C++ x86 Apps
- □ Windows SDK for Desktop C++ amd64 Apps
- □ Windows SDK for Desktop C++ arm Apps
- □ Windows SDK for Desktop C++ arm64 Apps

Windows SDK Signing Tools for Desktop Apps

Size: 5.0 MB

The Windows SDK Signing Tools for Desktop Apps

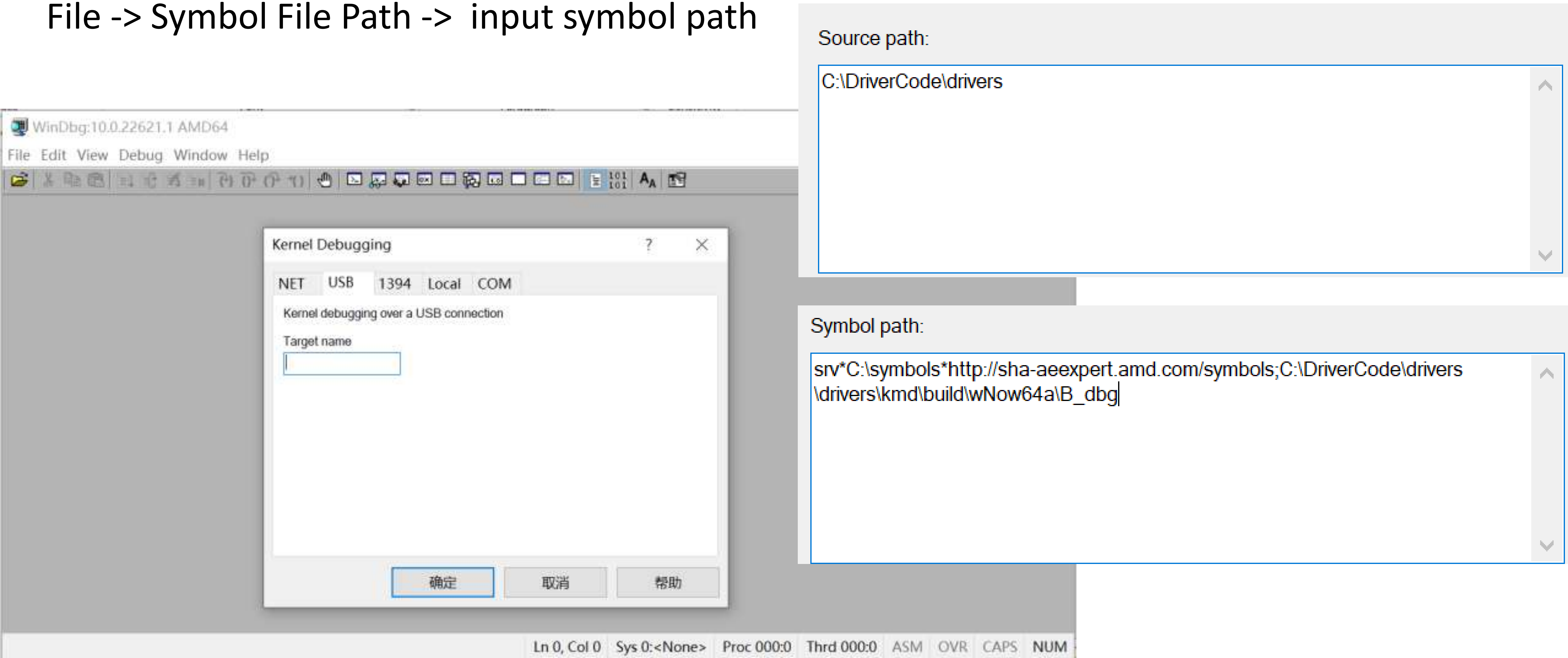
Estimated disk space required:	541.4 MB
Disk space available:	277.5 GB

1.1 Connecting a WinDbg Host System to a Test System by USB

File -> Kernel Debugging -> USB -> input Target name

File -> Source File Path -> input code path

File -> Symbol File Path -> input symbol path



WinDbg:10.0.22621.1 AMD64

Kernel Debugging

Kernel debugging over a USB connection

Target name

Source path:

C:\DriverCode\drivers

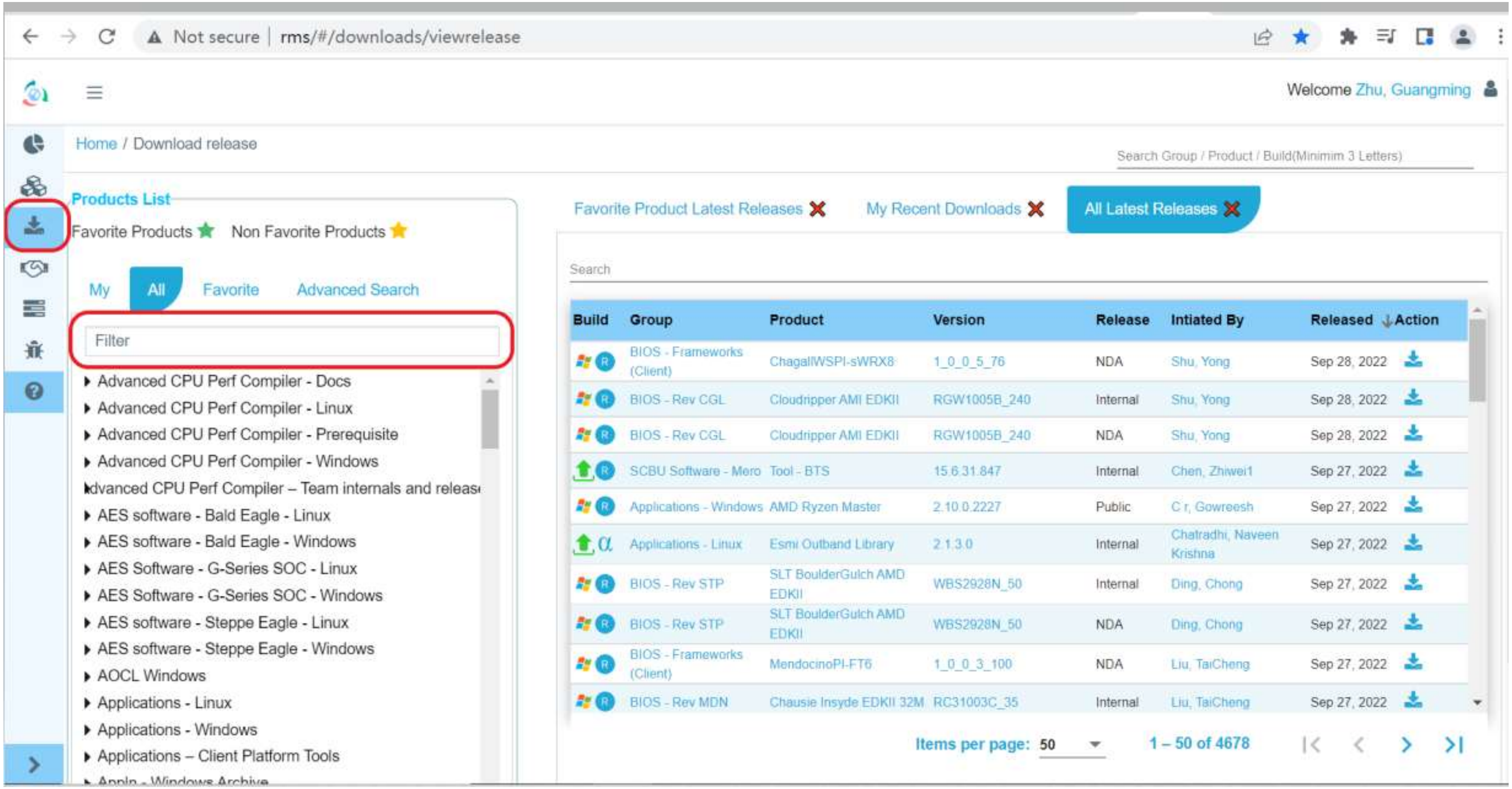
Symbol path:

srv*C:\symbols*http://sha-aeexpert.amd.com/symbols;C:\DriverCode\drivers\drivers\kmd\build\wNow64a\B_dbg

Some Problems

1.1 Reinstall VBIOS (optional)

http://rms/#/downloads/viewrelease



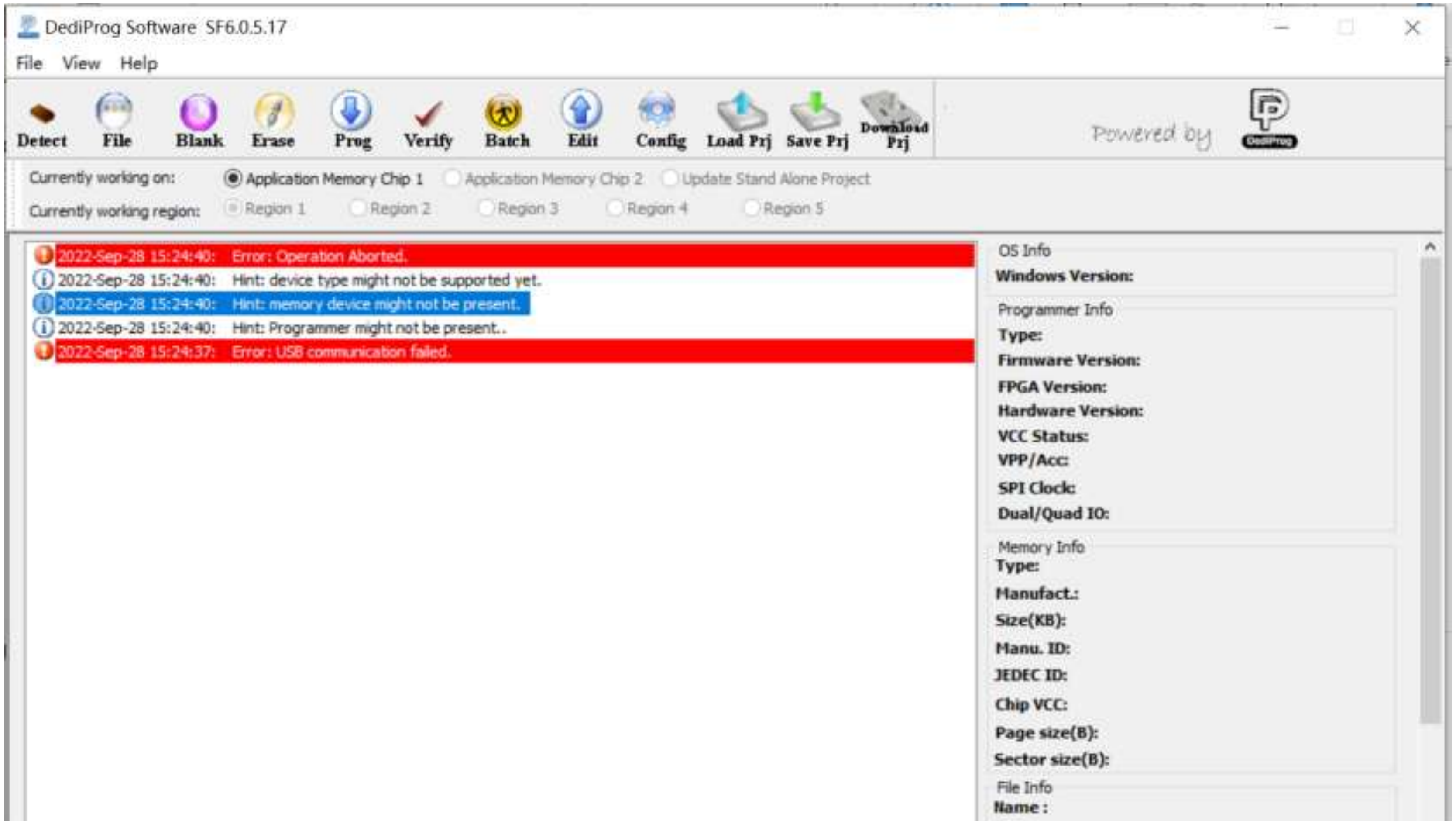
- Advanced CPU Perf Compiler - Docs
- Advanced CPU Perf Compiler - Linux
- Advanced CPU Perf Compiler - Prerequisite
- Advanced CPU Perf Compiler - Windows
- dvanced CPU Perf Compiler – Team internals and releas
- AES software - Bald Eagle - Linux
- AES software - Bald Eagle - Windows
- AES Software - G-Series SOC - Linux
- AES Software - G-Series SOC - Windows
- AES software - Steppe Eagle - Linux
- AES software - Steppe Eagle - Windows
- AOCL Windows
- Applications - Linux
- Applications - Windows
- Applications – Client Platform Tools
- Appln - Windows Archive

Build	Group	Product	Version	Release	Intiated By	Released	Action
R	BIOS - Frameworks (Client)	ChagallWSPI-sWRX8	1_0_0_5_76	NDA	Shu, Yong	Sep 28, 2022	
R	BIOS - Rev CGL	Cloudripper AMI EDKII	RGW1005B_240	Internal	Shu, Yong	Sep 28, 2022	
R	BIOS - Rev CGL	Cloudripper AMI EDKII	RGW1005B_240	NDA	Shu, Yong	Sep 28, 2022	
R	SCBU Software - Mero	Tool - BTS	15.6.31.847	Internal	Chen, Zhiwei1	Sep 27, 2022	
R	Applications - Windows	AMD Ryzen Master	2.10.0.2227	Public	C r, Gowreesh	Sep 27, 2022	
α	Applications - Linux	Esmi Outband Library	2.1.3.0	Internal	Chatradhi, Naveen Krishna	Sep 27, 2022	
R	BIOS - Rev STP	SLT BoulderGulch AMD EDKII	WBS2928N_50	Internal	Ding, Chong	Sep 27, 2022	
R	BIOS - Rev STP	SLT BoulderGulch AMD EDKII	WBS2928N_50	NDA	Ding, Chong	Sep 27, 2022	
R	BIOS - Frameworks (Client)	MendocinoPI-FT6	1_0_0_3_100	NDA	Liu, TaiCheng	Sep 27, 2022	
R	BIOS - Rev MDN	Chausie Insyde EDKII 32M	RC31003C_35	Internal	Liu, TaiCheng	Sep 27, 2022	

Items per page: 50 1 – 50 of 4678

1.1 Reinstall VBIOS (optional)

1. Click on detect and it should verify that the pins are connected
2. Click on file and open the file with the sbios in it. Load the .FD file. If there are miltiple FD files, click on the the .FD file with the same name as the sBIOS
3. Click Batch. This step does the entire flash.



Currently working on: ● Application Memory Chip 1 ○ Application Memory Chip 2 ○ Update Stand Alone Project

Currently working region: ○ Region 1 ○ Region 2 ○ Region 3 ○ Region 4 ○ Region 5

```
2022-Sep-28 15:24:40:   Error: Operation Aborted.
2022-Sep-28 15:24:40:   Hint: device type might not be supported yet.
2022-Sep-28 15:24:40:   Hint: memory device might not be present.
2022-Sep-28 15:24:40:   Hint: Programmer might not be present..
2022-Sep-28 15:24:37:   Error: USB communication failed.
```

OS Info
Windows Version:

Programmer Info
Type:
Firmware Version:
FPGA Version:
Hardware Version:
VCC Status:
VPP/Acc:
SPI Clock:
Dual/Quad IO:

Memory Info
Type:
Manufact.:
Size(KB):
Manu. ID:
JEDEC ID:
Chip VCC:
Page size(B):
Sector size(B):

File Info
Name :

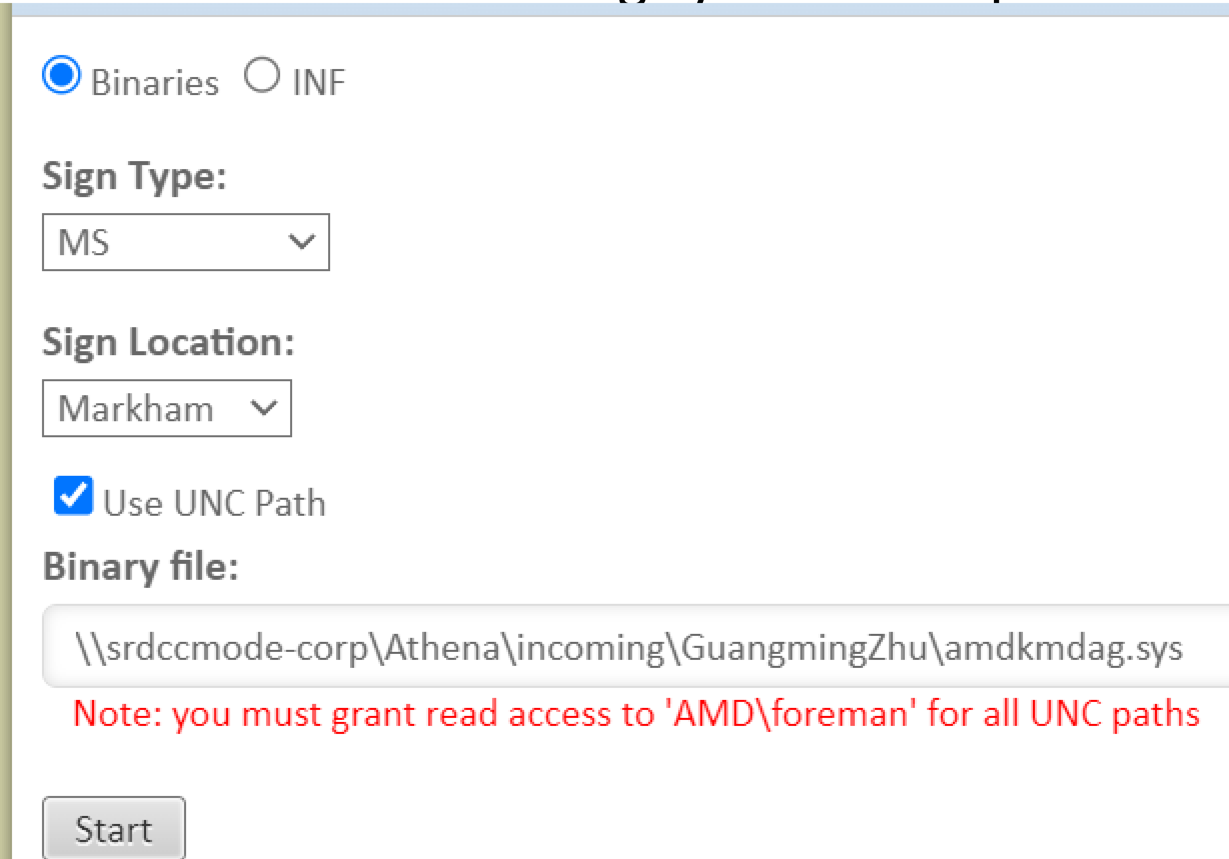
1.2 Sign the amdkmdag.sys File

Phenomenon: Yellow Bang in GPU or APU at Device Manager, and the specific explanation includes error code 52.

Happens in sv2 (newer version of win11).

http://osibuilds.amd.com/dashboard/#/job/1486840

.kdfiles -m amdkmdag.sys <des full path>



● Binaries ○ INF

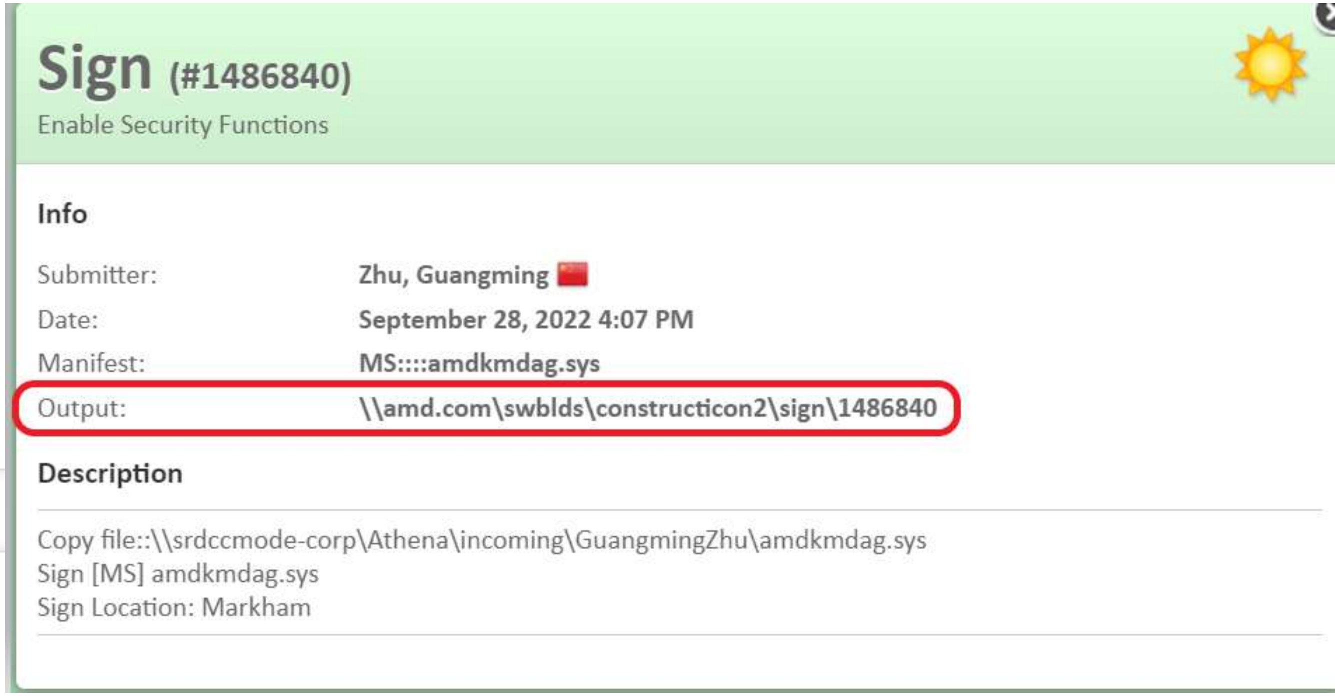
Sign Type: MS

Sign Location: Markham

☑ Use UNC Path

Binary file: \\srdccmode-corp\Athena\incoming\GuangmingZhu\amdkmdag.sys

Note: you must grant read access to 'AMD\foreman' for all UNC paths



Sign (#1486840)

Enable Security Functions

Info

Submitter:	Zhu, Guangming 🇨🇳
Date:	September 28, 2022 4:07 PM
Manifest:	MS::::amdkmdag.sys
Output:	\\amd.com\swblds\constructicon2\sign\1486840

Description

Copy file::\\srdccmode-corp\Athena\incoming\GuangmingZhu\amdkmdag.sys
Sign [MS] amdkmdag.sys
Sign Location: Markham

WinDbg Command

Some Basic Command

g : Continue

.reboot

.cls : Clear screen

sxn bpe : Temporarily cancel int 3 (cc) breakpoint

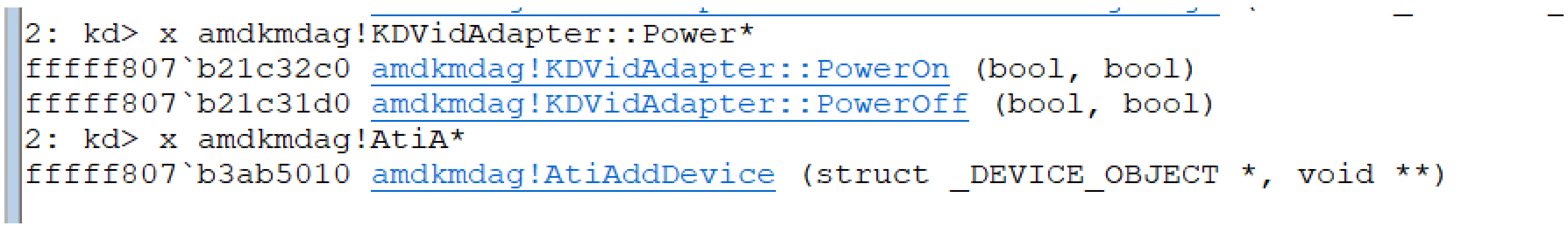
.kdfiles -m amdkmdag.sys <des full path> : Replace the drive of target machine

.reload /f amdkmdag.sys : Load symbols

.load <path> : Load dbg extension

!ghelp : Debugging help commands for arkdbg

x <module>!<function> : View function addresses and parameter types



```
2: kd> x amdkmdag!KDVidAdapter::Power*
fffff807`b21c32c0 amdkmdag!KDVidAdapter::PowerOn (bool, bool)
fffff807`b21c31d0 amdkmdag!KDVidAdapter::PowerOff (bool, bool)
2: kd> x amdkmdag!AtiA*
fffff807`b3ab5010 amdkmdag!AtiAddDevice (struct _DEVICE_OBJECT *, void **)
```

Access Memory by Arkdbgx.dll

Beside register access, ArkDbgX is powerful tool to access memory in different memory space.

CPU virtual --- in command line, it is specified by “/v <address>”.

CPU physical --- in command line, it is specified by “/p <address>”

GPU physical --- in command line, it is specified by “/x 0 <address>” (GART or MC address)

GPU virtual --- in command line, it is specified by “/x <hub>.<vmid> <address>”.

GPU offset --- in command line, it is specified by “/g <address>”

Display Memory

da – ASCII characters

db – Byte values and ASCII characters

dc – Double-word values(4 bytes) and ASCII characters

dd – Double-word values(4 bytes)

df – Single-precision floating-point numbers (4 bytes)

dp – Pointer-sized values

dq – Quad-word valued (8 bytes)

du – Unicode characters

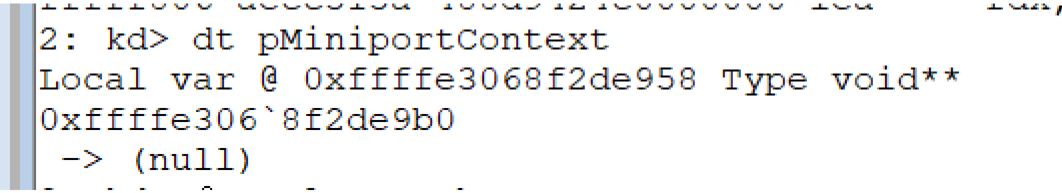
dw – Word values (2 bytes)

dW – Word values (2 bytes) and ASCII characters

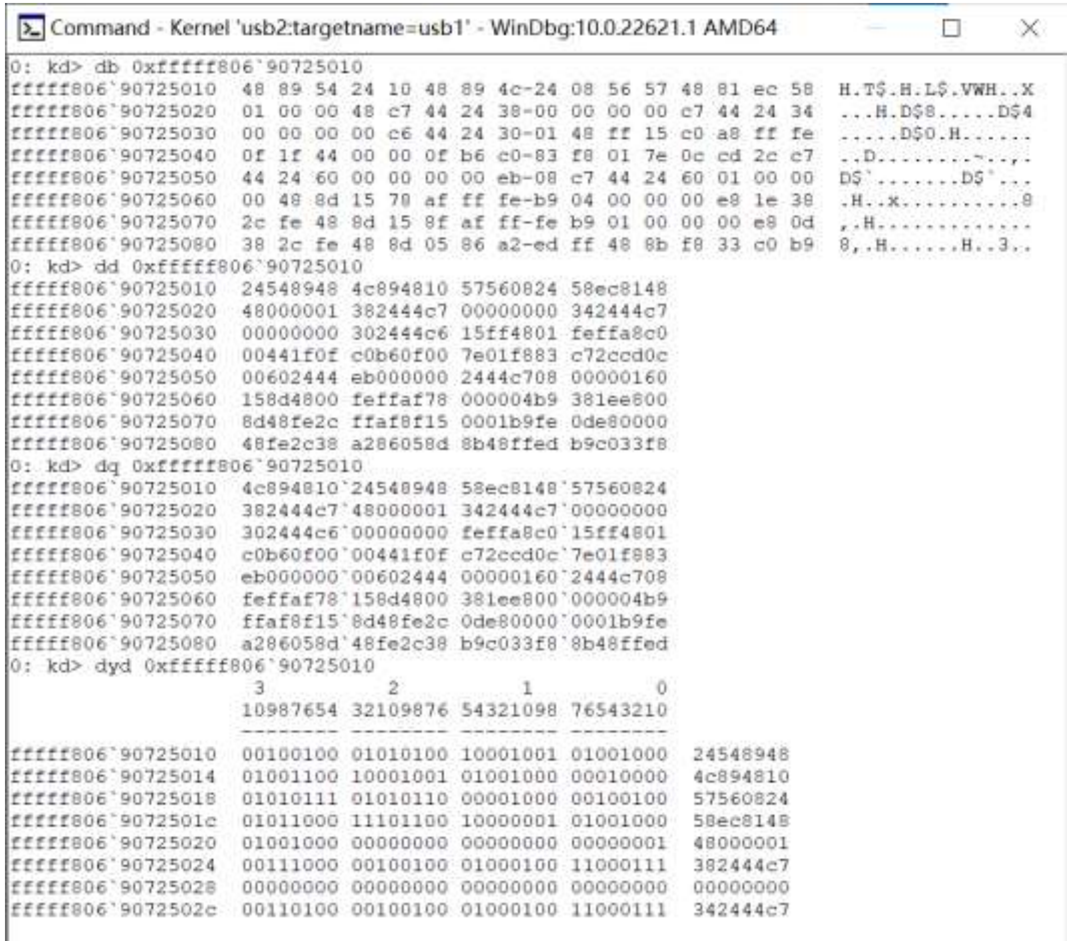
dyb – Binary values and byte values

dyd – Binary valued and double-word values (4 bytes)

eb | ew | ed | ef |eq – Memory editing



```
2: kd> dt pMiniportContext
Local var @ 0xffffe3068f2de958 Type void**
0xffffe306`8f2de9b0
 -> (null)
```



Command - Kernel 'usb2:targetname=usb1' - WinDbg:10.0.22621.1 AMD64

```
0: kd> db 0xfffff806`90725010
fffff806`90725010  48 89 54 24 10 48 89 4c-24 08 56 57 48 81 ec 58  H.T$.H.L$.VWH..X
fffff806`90725020  01 00 00 48 c7 44 24 38-00 00 00 00 c7 44 24 34  ...H.D$8.....D$4
fffff806`90725030  00 00 00 00 c6 44 24 30-01 48 ff 15 c0 a8 ff fe  .....D$0.H......
fffff806`90725040  0f 1f 44 00 00 0f b6 c0-83 f8 01 7e 0c cd 2c c7  ..D........~..,.
fffff806`90725050  44 24 60 00 00 00 00 eb-08 c7 44 24 60 01 00 00  D$`.......D$`...
fffff806`90725060  00 48 8d 15 78 af ff fe-b9 04 00 00 00 e8 1e 38  .H..x..........8
fffff806`90725070  2c fe 48 8d 15 8f af ff-fe b9 01 00 00 00 e8 0d  ,.H.............
fffff806`90725080  38 2c fe 48 8d 05 86 a2-ed ff 48 8b f8 33 c0 b9  8,.H......H..3..
0: kd> dd 0xfffff806`90725010
fffff806`90725010  24548948 4c894810 57560824 58ec8148
fffff806`90725020  48000001 382444c7 00000000 342444c7
fffff806`90725030  00000000 302444c6 15ff4801 feffa8c0
fffff806`90725040  00441f0f c0b60f00 7e01f883 c72ccd0c
fffff806`90725050  00602444 eb000000 2444c708 00000160
fffff806`90725060  158d4800 feffaf78 000004b9 381ee800
fffff806`90725070  8d48fe2c ffaf8f15 0001b9fe 0de80000
fffff806`90725080  48fe2c38 a286058d 8b48ffed b9c033f8
0: kd> dq 0xfffff806`90725010
fffff806`90725010  4c894810`24548948 58ec8148`57560824
fffff806`90725020  382444c7`48000001 342444c7`00000000
fffff806`90725030  302444c6`00000000 feffa8c0`15ff4801
fffff806`90725040  c0b60f00`00441f0f c72ccd0c`7e01f883
fffff806`90725050  eb000000`00602444 00000160`2444c708
fffff806`90725060  feffaf78`158d4800 381ee800`000004b9
fffff806`90725070  ffaf8f15`8d48fe2c 0de80000`0001b9fe
fffff806`90725080  a286058d`48fe2c38 b9c033f8`8b48ffed
0: kd> dyd 0xfffff806`90725010
                    3        2        1        0
                   10987654 32109876 54321098 76543210
                   -------- -------- -------- --------
fffff806`90725010  00100100 01010100 10001001 01001000  24548948
fffff806`90725014  01001100 10001001 01001000 00010000  4c894810
fffff806`90725018  01010111 01010110 00001000 00100100  57560824
fffff806`9072501c  01011000 11101100 10000001 01001000  58ec8148
fffff806`90725020  01001000 00000000 00000000 00000001  48000001
fffff806`90725024  00111000 00100100 01000100 11000111  382444c7
fffff806`90725028  00000000 00000000 00000000 00000000  00000000
fffff806`9072502c  00110100 00100100 01000100 11000111  342444c7
```

Breakpoint Types

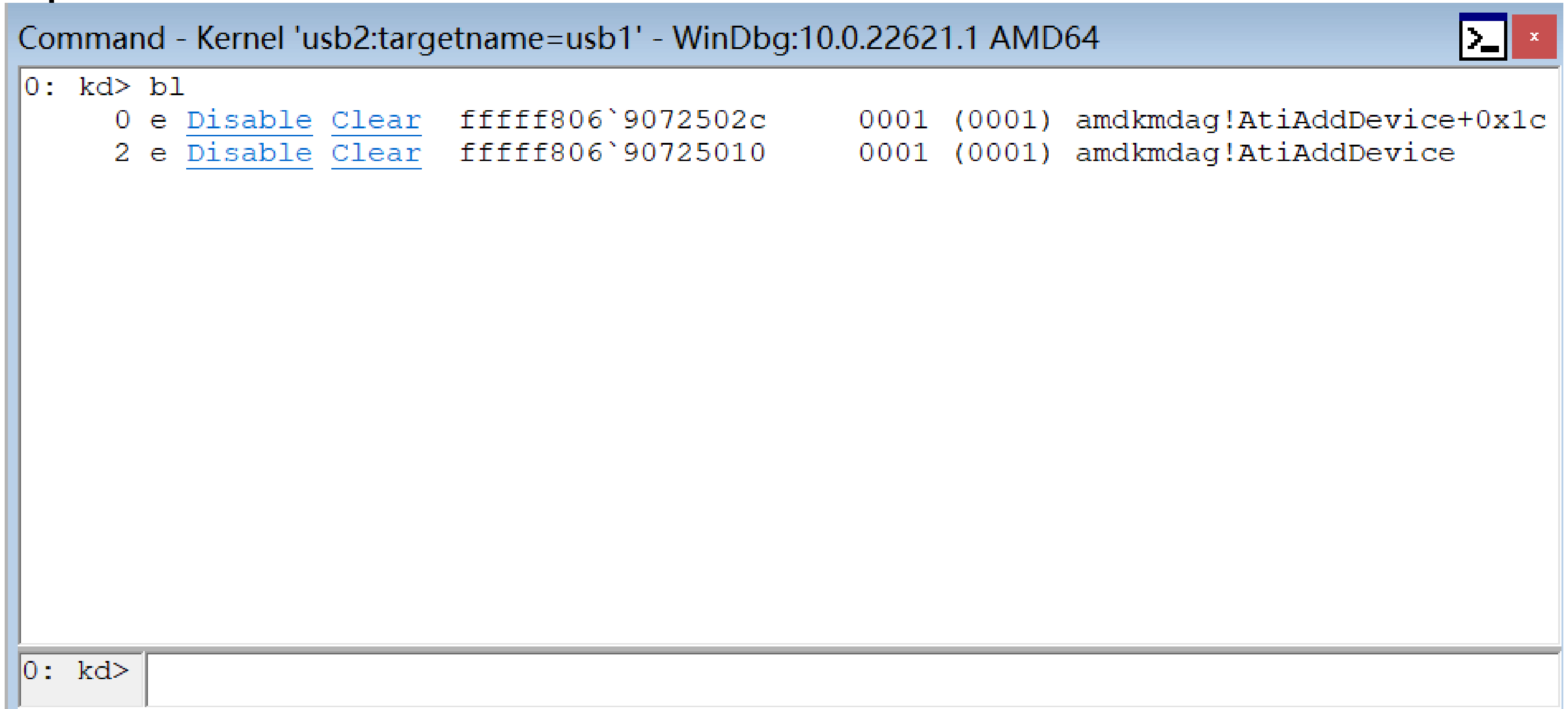
bp (F9) software style of breakpoint (interrupt) does not use hardware debugging registers – cleared at unload

bu – Unresolved breakpoint – persistent after unload

ba – Breakpoint on memory access. Uses hardware debug when available on processor .

bm – Break by symbol pattern. (bm amdkmdag!Ati*)

bl – list of breakpoints


Command - Kernel 'usb2:targetname=usb1' - WinDbg:10.0.22621.1 AMD64

```
0: kd> bl
     0 e Disable Clear  fffff806`9072502c     0001 (0001) amdkmdag!AtiAddDevice+0x1c
     2 e Disable Clear  fffff806`90725010     0001 (0001) amdkmdag!AtiAddDevice
```

```
0: kd>
```

Conditional Statement

?<var>: display the pointer of var

??<var>: display the value of var

```
bu amdkmdag!AtiAddDevice+0x29 "j(poi(ResultLength)>0x0)'.echo wide blt';'.echo small blt;g'"
```

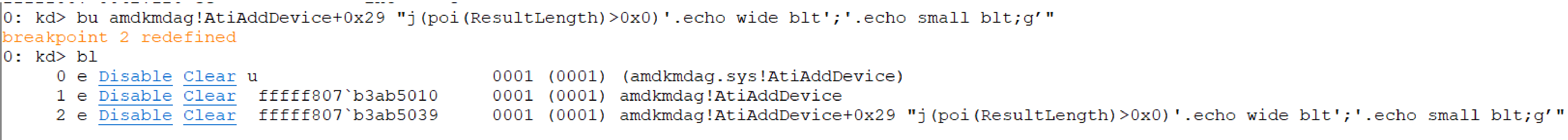
j : Conditional statement,

; : Conditional segmentation

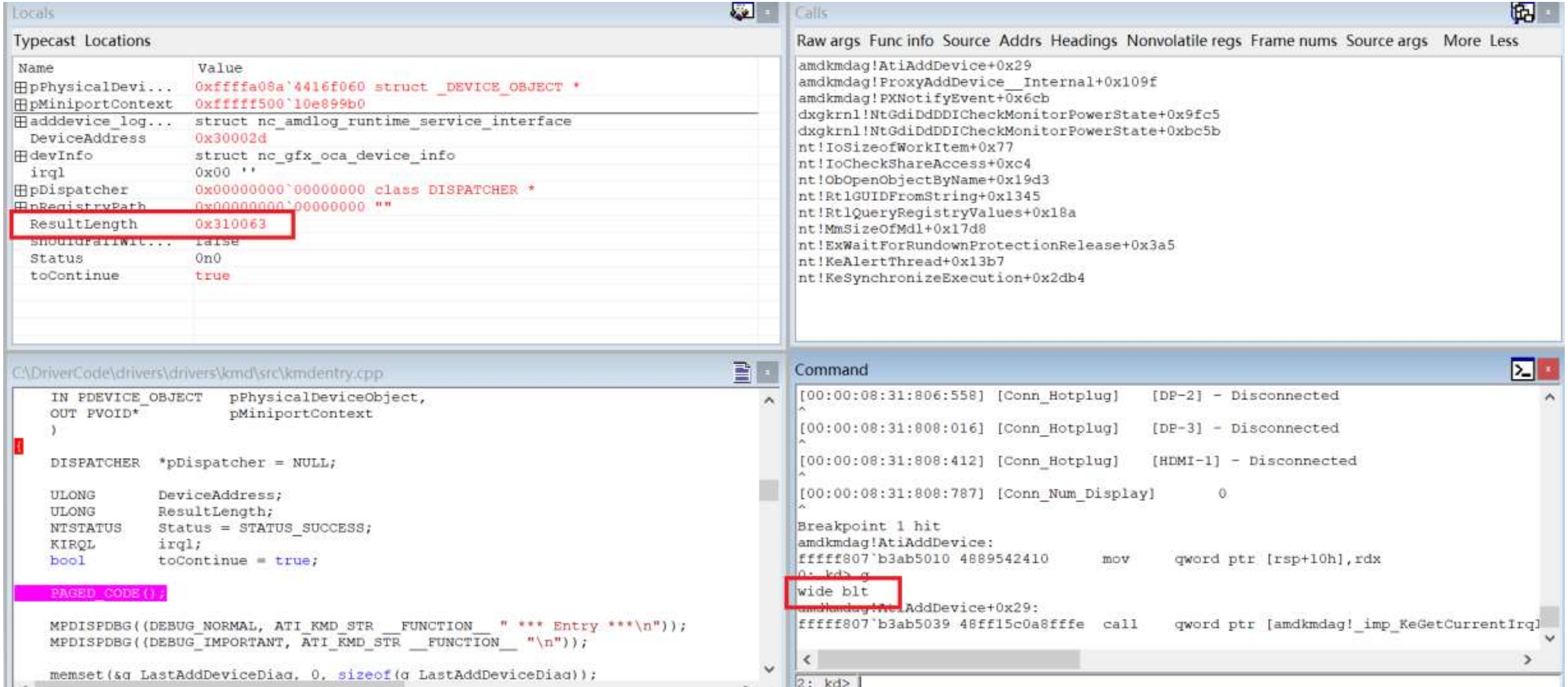
.echo : Print information

```
bu amdkmdag!AtiAddDevice+0x29 ".if@@c++(poi(ResultLength)>0x0){??var}.else{.echo short blt}"
```

Conditional Statement



```
0: kd> bu amdkmdag!AtiAddDevice+0x29 "j(poi(ResultLength)>0x0)'.echo wide blt';'.echo small blt;g'"
breakpoint 2 redefined
0: kd> bl
     0 e Disable Clear u             0001 (0001) (amdkmdag.sys!AtiAddDevice)
     1 e Disable Clear  fffff807`b3ab5010     0001 (0001) amdkmdag!AtiAddDevice
     2 e Disable Clear  fffff807`b3ab5039     0001 (0001) amdkmdag!AtiAddDevice+0x29 "j(poi(ResultLength)>0x0)'.echo wide blt';'.echo small blt;g'"
```



```
Name                Value
⊞pPhysicalDevi...   0xffffa08a`4416f060 struct _DEVICE_OBJECT *
⊞pMiniportContext   0xfffff500`10e899b0
⊞adddevice_log...   struct nc_amdlog_runtime_service_interface
  DeviceAddress     0x30002d
⊞devInfo            struct nc_gfx_oca_device_info
  irql              0x00 ''
⊞pDispatcher        0x00000000`00000000 class DISPATCHER *
⊞pRegistryPath      0x00000000`00000000 ""
  ResultLength      0x310063
  shouldFailWit...  false
  Status            0n0
  toContinue        true
```

Raw args Func info Source Addrs Headings Nonvolatile regs Frame nums Source args More Less

```
amdkmdag!AtiAddDevice+0x29
amdkmdag!ProxyAddDevice__Internal+0x109f
amdkmdag!PXNotifyEvent+0x6cb
dxgkrnl!NtGdiDdDDICheckMonitorPowerState+0x9fc5
dxgkrnl!NtGdiDdDDICheckMonitorPowerState+0xbc5b
nt!IoSizeofWorkItem+0x77
nt!IoCheckShareAccess+0xc4
nt!ObOpenObjectByName+0x19d3
nt!RtlGUIDFromString+0x1345
nt!RtlQueryRegistryValues+0x18a
nt!MmSizeOfMdl+0x17d8
nt!ExWaitForRundownProtectionRelease+0x3a5
nt!KeAlertThread+0x13b7
nt!KeSynchronizeExecution+0x2db4
```

```
    IN PDEVICE_OBJECT   pPhysicalDeviceObject,
    OUT PVOID*          pMiniportContext
    )
{

    DISPATCHER  *pDispatcher = NULL;

    ULONG        DeviceAddress;
    ULONG        ResultLength;
    NTSTATUS     Status = STATUS_SUCCESS;
    KIRQL        irql;
    bool         toContinue = true;

    PAGED_CODE();

    MPDISPDBG((DEBUG_NORMAL, ATI_KMD_STR __FUNCTION__ " *** Entry ***\n"));
    MPDISPDBG((DEBUG_IMPORTANT, ATI_KMD_STR __FUNCTION__ "\n"));

    memset(&g_LastAddDeviceDiag, 0, sizeof(g_LastAddDeviceDiag));
```

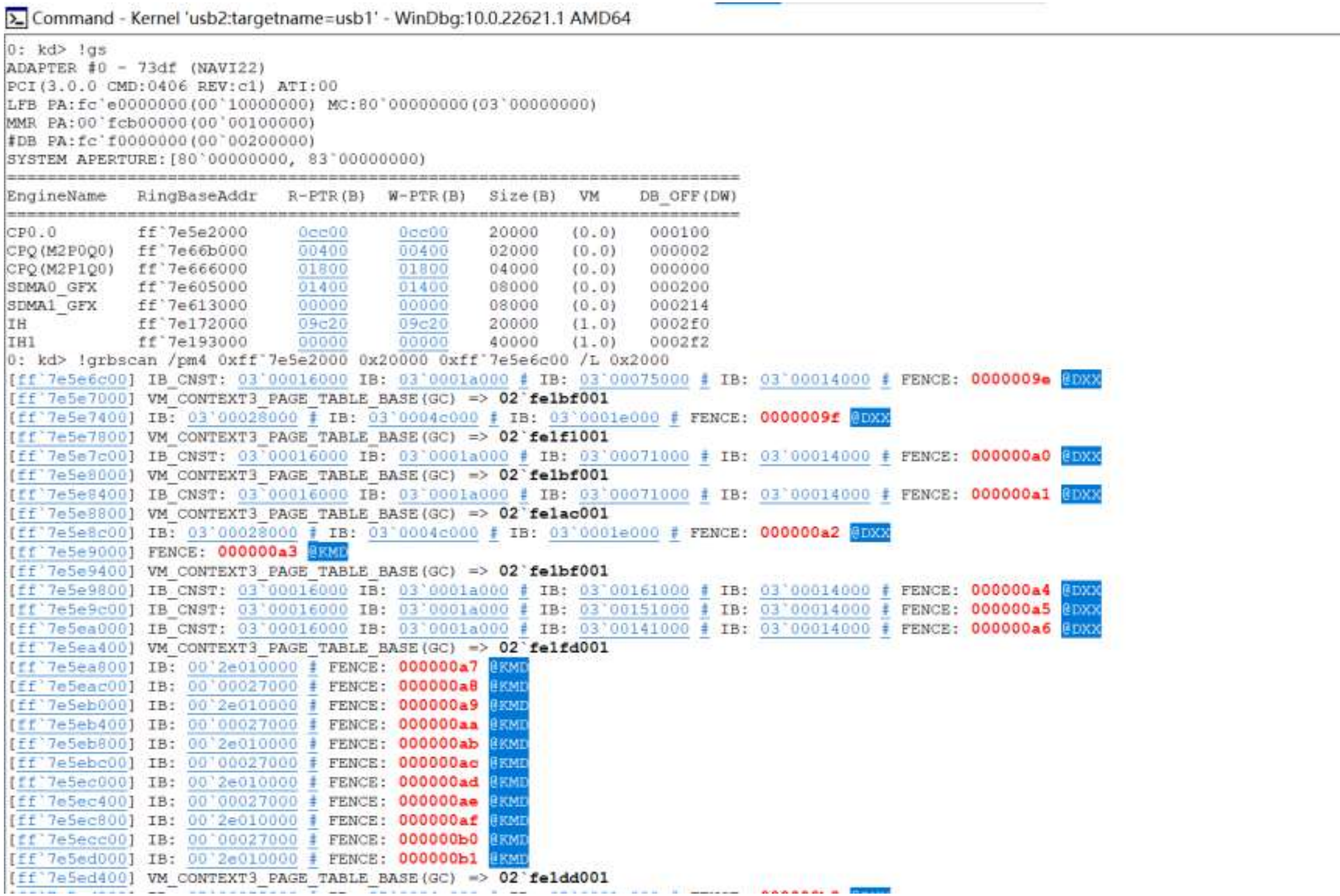
```
[00:00:08:31:806:558] [Conn_Hotplug]    [DP-2] - Disconnected
^
[00:00:08:31:808:016] [Conn_Hotplug]    [DP-3] - Disconnected
^
[00:00:08:31:808:412] [Conn_Hotplug]    [HDMI-1] - Disconnected
^
[00:00:08:31:808:787] [Conn_Num_Display]       0
^
Breakpoint 1 hit
amdkmdag!AtiAddDevice:
fffff807`b3ab5010 4889542410      mov     qword ptr [rsp+10h],rdx
0: kd> g
wide blt
amdkmdag!AtiAddDevice+0x29:
fffff807`b3ab5039 48ff15c0a8fffe  call    qword ptr [amdkmdag!_imp_KeGetCurrentIrql]

2: kd>
```

Arkdbgx.dll---!gs

File Position: \\valfs\shareAll\F\felixwei\tools\windbg\ext

.load <path>

Unload: .unload <fileName>



Command - Kernel 'usb2:targetname=usb1' - WinDbg:10.0.22621.1 AMD64

```
0: kd> !gs
ADAPTER #0 - 73df (NAVI22)
PCI(3.0.0 CMD:0406 REV:c1) ATI:00
LFB PA:fc`e0000000(00`10000000) MC:80`00000000(03`00000000)
MMR PA:00`fcb00000(00`00100000)
#DB PA:fc`f0000000(00`00200000)
SYSTEM APERTURE:[80`00000000, 83`00000000)
=========================================================================
EngineName    RingBaseAddr    R-PTR(B)   W-PTR(B)   Size(B)   VM     DB_OFF(DW)
=========================================================================
CP0.0         ff`7e5e2000     0cc00      0cc00      20000     (0.0)  000100
CPQ(M2P0Q0)   ff`7e66b000     00400      00400      02000     (0.0)  000002
CPQ(M2P1Q0)   ff`7e666000     01800      01800      04000     (0.0)  000000
SDMA0_GFX     ff`7e605000     01400      01400      08000     (0.0)  000200
SDMA1_GFX     ff`7e613000     00000      00000      08000     (0.0)  000214
IH            ff`7e172000     09c20      09c20      20000     (1.0)  0002f0
IH1           ff`7e193000     00000      00000      40000     (1.0)  0002f2
0: kd> !grbscan /pm4 0xff`7e5e2000 0x20000 0xff`7e5e6c00 /L 0x2000
[ff`7e5e6c00] IB_CNST: 03`00016000 IB: 03`0001a000 # IB: 03`00075000 # IB: 03`00014000 # FENCE: 0000009e @DXX
[ff`7e5e7000] VM_CONTEXT3_PAGE_TABLE_BASE(GC) => 02`felbf001
[ff`7e5e7400] IB: 03`00028000 # IB: 03`0004c000 # IB: 03`0001e000 # FENCE: 0000009f @DXX
[ff`7e5e7800] VM_CONTEXT3_PAGE_TABLE_BASE(GC) => 02`felf1001
[ff`7e5e7c00] IB_CNST: 03`00016000 IB: 03`0001a000 # IB: 03`00071000 # IB: 03`00014000 # FENCE: 000000a0 @DXX
[ff`7e5e8000] VM_CONTEXT3_PAGE_TABLE_BASE(GC) => 02`felbf001
[ff`7e5e8400] IB_CNST: 03`00016000 IB: 03`0001a000 # IB: 03`00071000 # IB: 03`00014000 # FENCE: 000000a1 @DXX
[ff`7e5e8800] VM_CONTEXT3_PAGE_TABLE_BASE(GC) => 02`felac001
[ff`7e5e8c00] IB: 03`00028000 # IB: 03`0004c000 # IB: 03`0001e000 # FENCE: 000000a2 @DXX
[ff`7e5e9000] FENCE: 000000a3 @KMD
[ff`7e5e9400] VM_CONTEXT3_PAGE_TABLE_BASE(GC) => 02`felbf001
[ff`7e5e9800] IB_CNST: 03`00016000 IB: 03`0001a000 # IB: 03`00161000 # IB: 03`00014000 # FENCE: 000000a4 @DXX
[ff`7e5e9c00] IB_CNST: 03`00016000 IB: 03`0001a000 # IB: 03`00151000 # IB: 03`00014000 # FENCE: 000000a5 @DXX
[ff`7e5ea000] IB_CNST: 03`00016000 IB: 03`0001a000 # IB: 03`00141000 # IB: 03`00014000 # FENCE: 000000a6 @DXX
[ff`7e5ea400] VM_CONTEXT3_PAGE_TABLE_BASE(GC) => 02`felfd001
[ff`7e5ea800] IB: 00`2e010000 # FENCE: 000000a7 @KMD
[ff`7e5eac00] IB: 00`00027000 # FENCE: 000000a8 @KMD
[ff`7e5eb000] IB: 00`2e010000 # FENCE: 000000a9 @KMD
[ff`7e5eb400] IB: 00`00027000 # FENCE: 000000aa @KMD
[ff`7e5eb800] IB: 00`2e010000 # FENCE: 000000ab @KMD
[ff`7e5ebc00] IB: 00`00027000 # FENCE: 000000ac @KMD
[ff`7e5ec000] IB: 00`2e010000 # FENCE: 000000ad @KMD
[ff`7e5ec400] IB: 00`00027000 # FENCE: 000000ae @KMD
[ff`7e5ec800] IB: 00`2e010000 # FENCE: 000000af @KMD
[ff`7e5ecc00] IB: 00`00027000 # FENCE: 000000b0 @KMD
[ff`7e5ed000] IB: 00`2e010000 # FENCE: 000000b1 @KMD
[ff`7e5ed400] VM_CONTEXT3_PAGE_TABLE_BASE(GC) => 02`feldd001
```

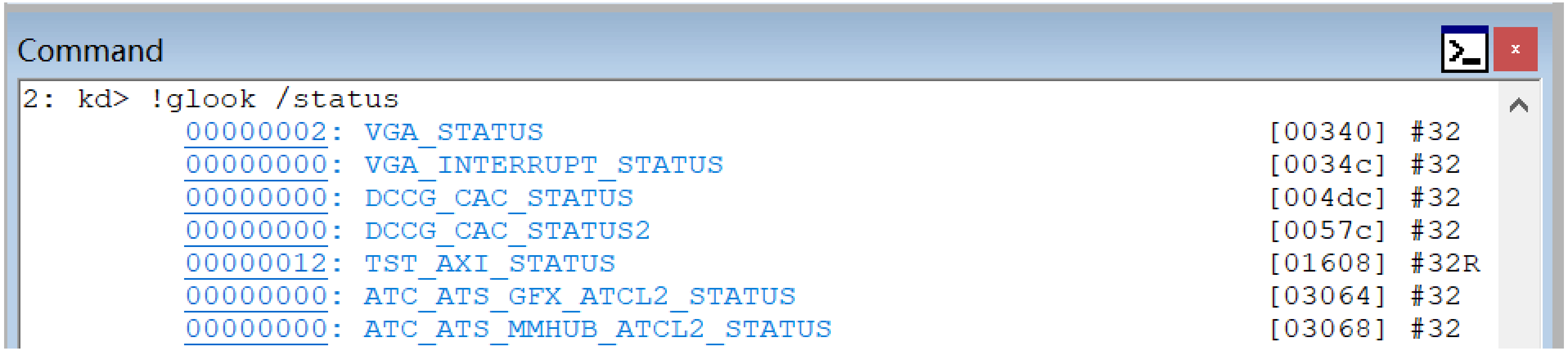
Arkdbgx.dll

!ghelp

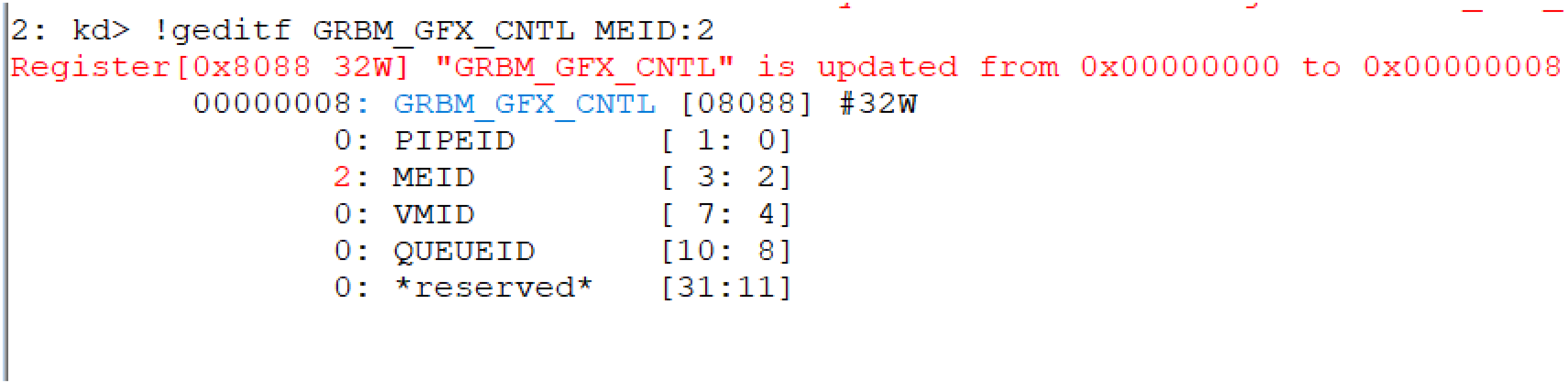
!glook: read register.

!gedit: update register. !gedit GRBM_GFX_CNTL (2 << 2) | 1 | (3 << 8).

!geditf: update register with field directly. !geditf GRBM_GFX_CNTL MEID:2 PIPEID:1 QUEUEID:3



```
2: kd> !glook /status
         00000002: VGA_STATUS                                          [00340] #32
         00000000: VGA_INTERRUPT_STATUS                                [0034c] #32
         00000000: DCCG_CAC_STATUS                                     [004dc] #32
         00000000: DCCG_CAC_STATUS2                                    [0057c] #32
         00000012: TST_AXI_STATUS                                      [01608] #32R
         00000000: ATC_ATS_GFX_ATCL2_STATUS                            [03064] #32
         00000000: ATC_ATS_MMHUB_ATCL2_STATUS                          [03068] #32
```

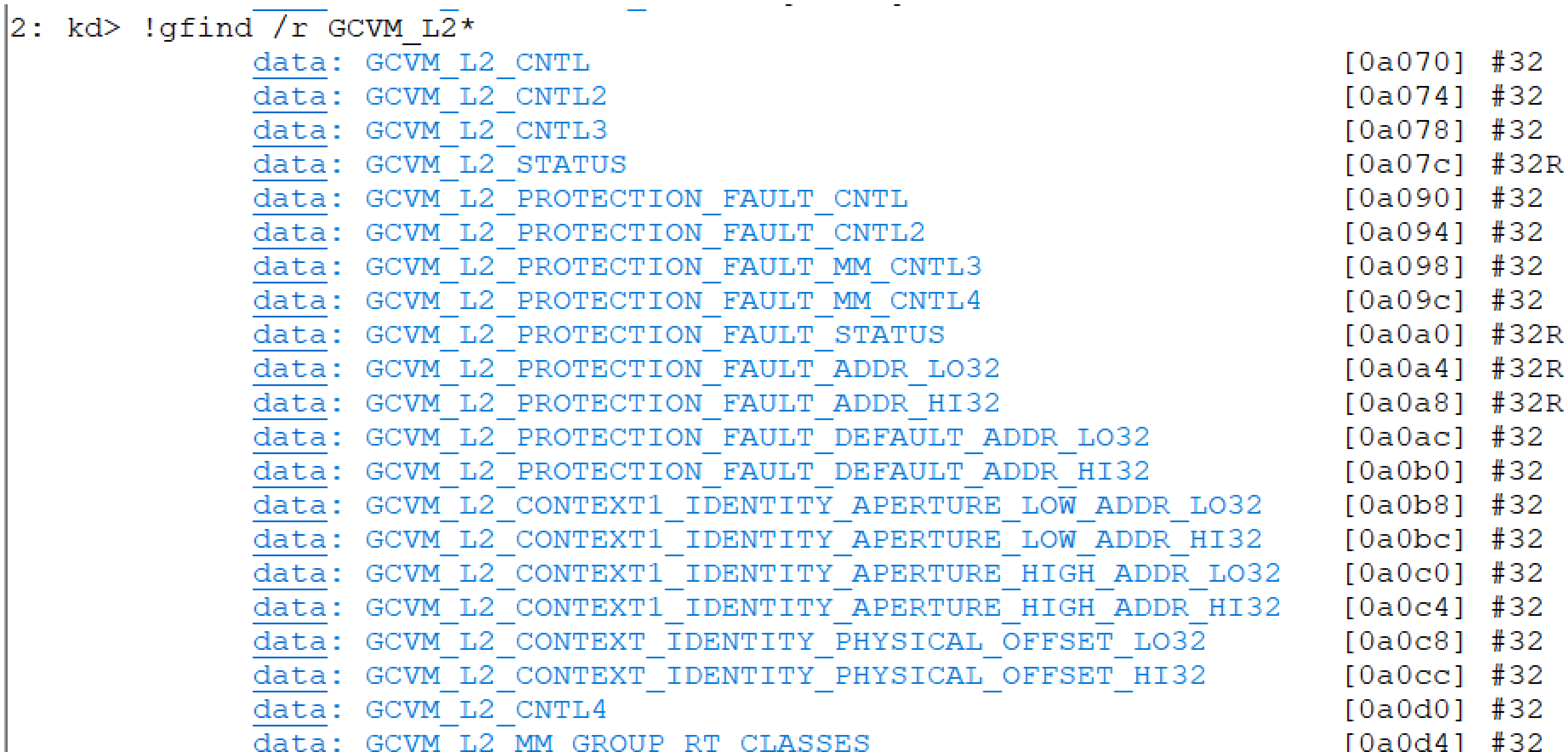


```
2: kd> !geditf GRBM_GFX_CNTL MEID:2
Register[0x8088 32W] "GRBM_GFX_CNTL" is updated from 0x00000000 to 0x00000008
         00000008: GRBM_GFX_CNTL [08088] #32W
                0: PIPEID       [ 1: 0]
                2: MEID         [ 3: 2]
                0: VMID         [ 7: 4]
                0: QUEUEID      [10: 8]
                0: *reserved*   [31:11]
```

Arkdbgx.dll

!gfind: sometimes reading some registers cause asic hard hang, but you forget the register name. You can use it to find registers with matching name or field. For example, “!gfind /fields *bigk*” command will find all the regiters with part of its field name has “bigk”.

!geditor: this is very handy Windows GUI, but you can’t use it in remote debugging, and in non-Windows platform.



```
2: kd> !gfind /r GCVM_L2*
             data: GCVM_L2_CNTL                                         [0a070] #32
             data: GCVM_L2_CNTL2                                        [0a074] #32
             data: GCVM_L2_CNTL3                                        [0a078] #32
             data: GCVM_L2_STATUS                                       [0a07c] #32R
             data: GCVM_L2_PROTECTION_FAULT_CNTL                        [0a090] #32
             data: GCVM_L2_PROTECTION_FAULT_CNTL2                       [0a094] #32
             data: GCVM_L2_PROTECTION_FAULT_MM_CNTL3                    [0a098] #32
             data: GCVM_L2_PROTECTION_FAULT_MM_CNTL4                    [0a09c] #32
             data: GCVM_L2_PROTECTION_FAULT_STATUS                      [0a0a0] #32R
             data: GCVM_L2_PROTECTION_FAULT_ADDR_LO32                   [0a0a4] #32R
             data: GCVM_L2_PROTECTION_FAULT_ADDR_HI32                   [0a0a8] #32R
             data: GCVM_L2_PROTECTION_FAULT_DEFAULT_ADDR_LO32           [0a0ac] #32
             data: GCVM_L2_PROTECTION_FAULT_DEFAULT_ADDR_HI32           [0a0b0] #32
             data: GCVM_L2_CONTEXT1_IDENTITY_APERTURE_LOW_ADDR_LO32     [0a0b8] #32
             data: GCVM_L2_CONTEXT1_IDENTITY_APERTURE_LOW_ADDR_HI32     [0a0bc] #32
             data: GCVM_L2_CONTEXT1_IDENTITY_APERTURE_HIGH_ADDR_LO32    [0a0c0] #32
             data: GCVM_L2_CONTEXT1_IDENTITY_APERTURE_HIGH_ADDR_HI32    [0a0c4] #32
             data: GCVM_L2_CONTEXT_IDENTITY_PHYSICAL_OFFSET_LO32        [0a0c8] #32
             data: GCVM_L2_CONTEXT_IDENTITY_PHYSICAL_OFFSET_HI32        [0a0cc] #32
             data: GCVM_L2_CNTL4                                        [0a0d0] #32
             data: GCVM_L2_MM_GROUP_RT_CLASSES                          [0a0d4] #32
```

Q & A

Thanks