Debugging KVM using Intel DCI Technology

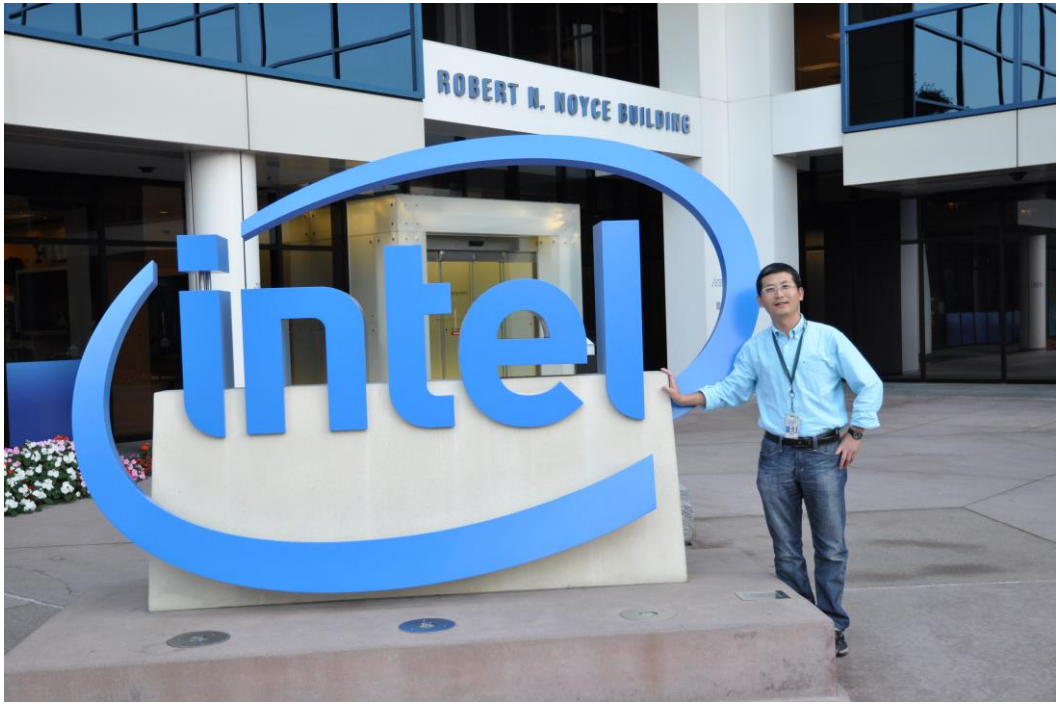
2020/10/30

Raymond Zhang

Self Introduction

- Raymond Zhang
- Worked at Intel 2003 ~ 2016
- Xen Development in 2009

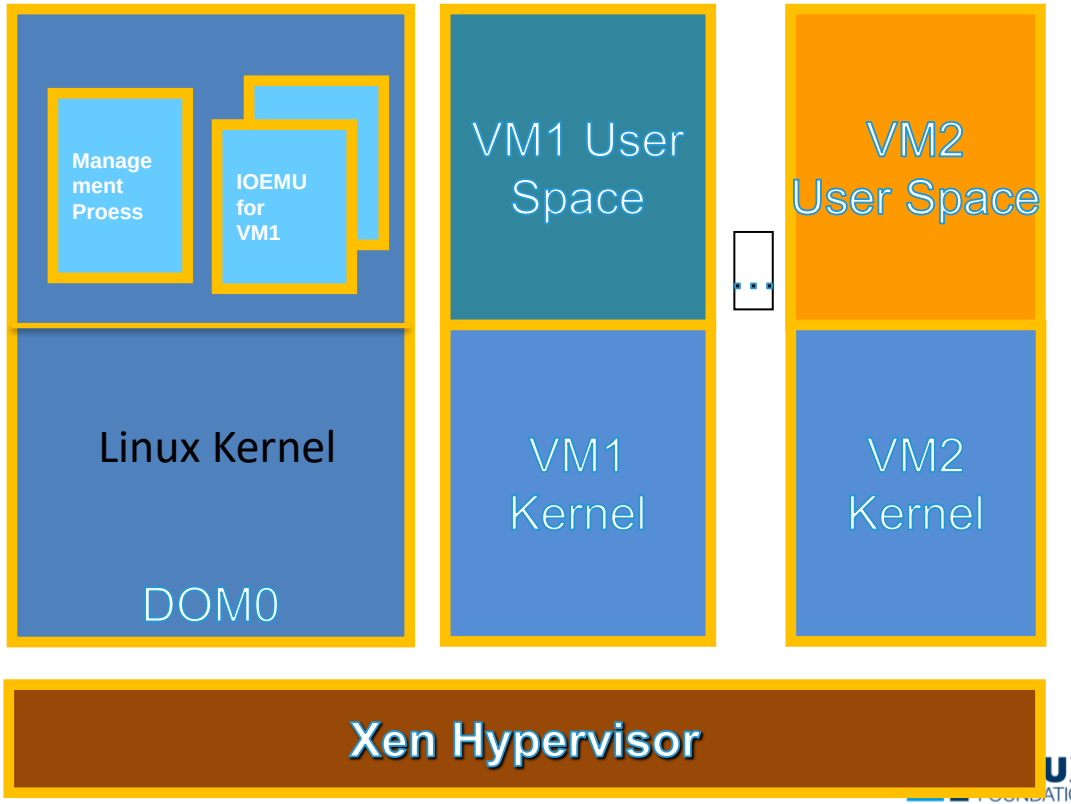
- Made Nvidia gfx worked in windows VM
- Root caused a TDR BSOD to a MMIO bug in shadow memory after 3 months debugging



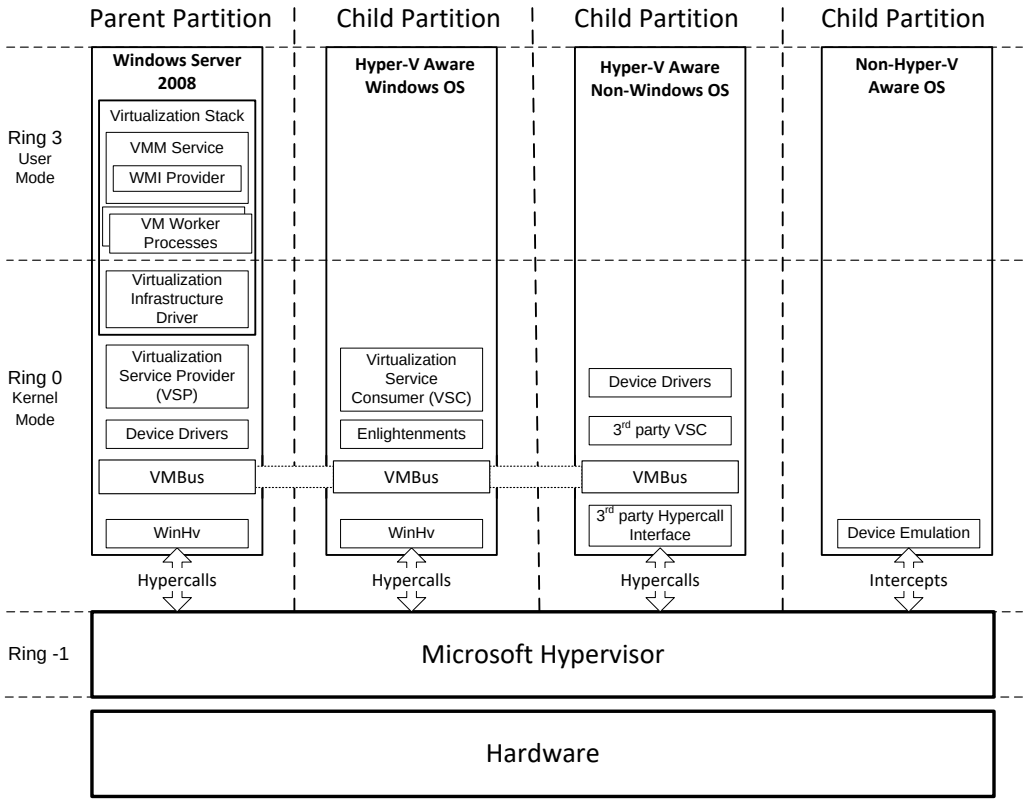
yinkui.zhang@outlook.com

The Classic Xen Architecture

- Domain 0
  - Hosts Qemu process (aka ioemu)
  - It's a VM too, but it's privileged
  - Has drivers for most real hardware



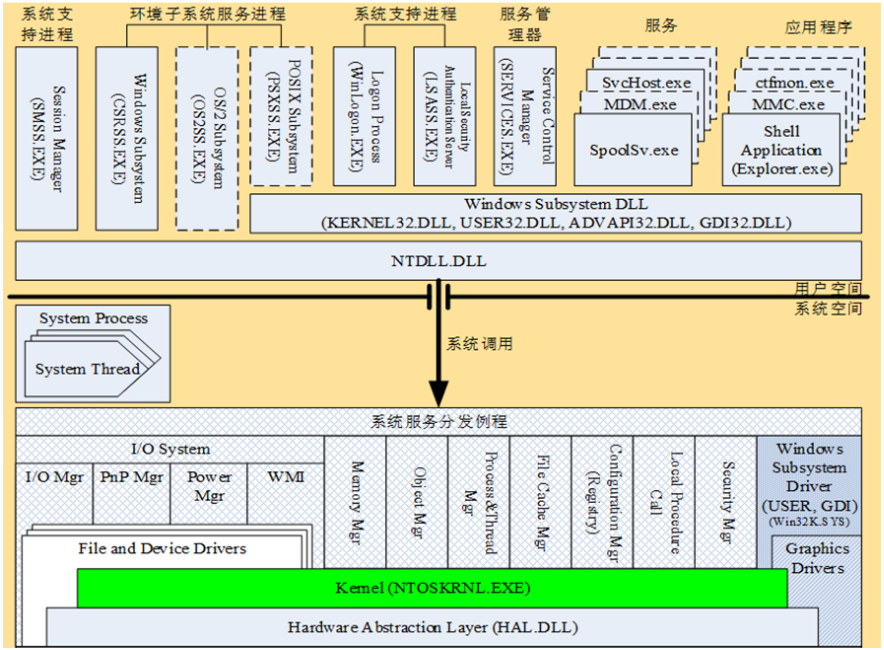
Hyper-V is Simillar

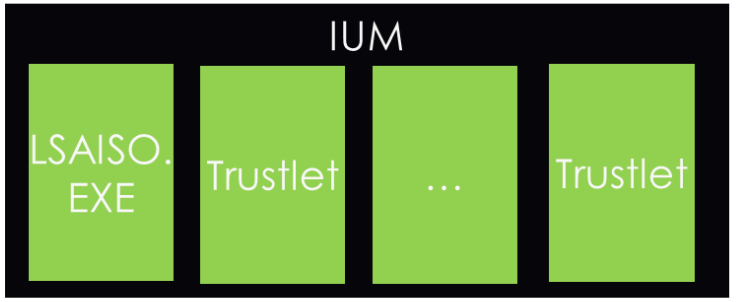


- There is a hypervisor under all VMs, including the privilege Parent Partition (domain 0 in Xen)

* From Microsoft

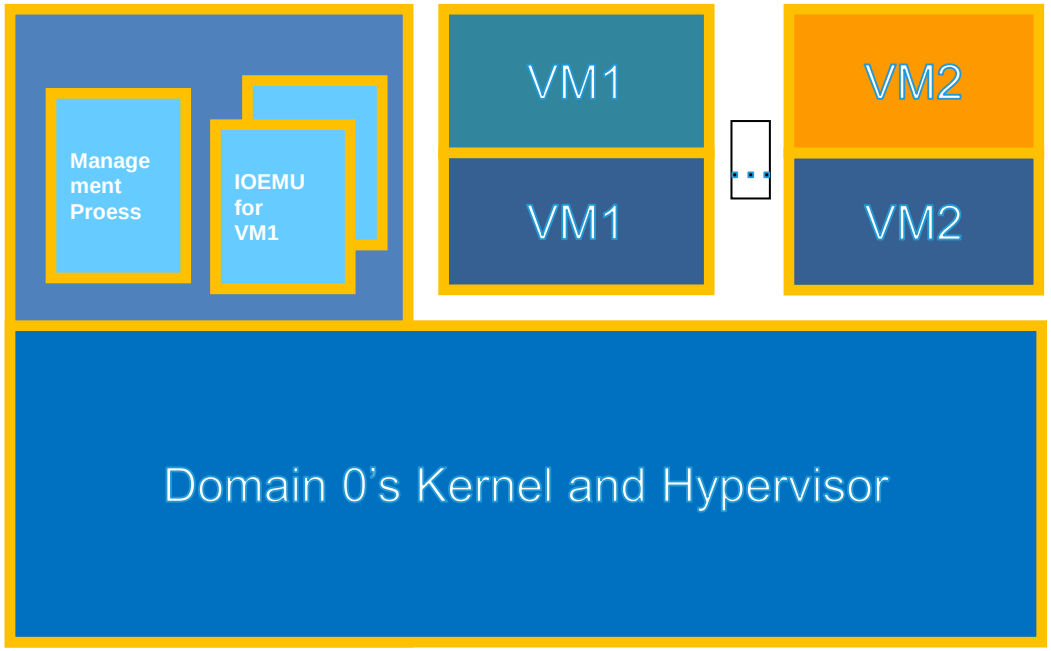
Windows 10's IUM (VBS)





HVIX64.EXE

KVM is Smarter



- Combine Dom0 Kernel and hypervisor into 1
- It has a lot of benefits

kvm_cpu_vmxon

```
static void kvm_cpu_vmxon(u64 addr)
{
        cr4_set_bits(X86_CR4_VMXE);
        intel_pt_handle_vmx(1);

        asm volatile ("vmxon %0" : : "m"(addr));
}
```

linux-5.0.7\arch\x86\kvm\vmx\vmx.c

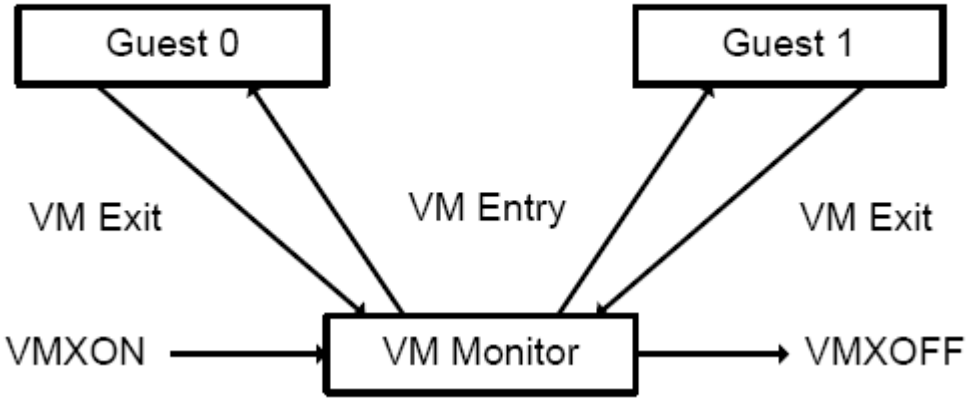
It’s inlined into kvm_intel!hardware_enable

```
kvm_intel!hardware_enable+0xca
  298 ffffffff`c0c8f80a 4889df        mov     rdi,rbx
  299 ffffffff`c0c8f80d 57            push    rdi
  299 ffffffff`c0c8f80e 9d            popfq
  299 ffffffff`c0c8f80f 0f1f440000    nop     dword ptr [rax+rax]
  299 ffffffff`c0c8f814 bf01000000    mov     edi,1
  299 ffffffff`c0c8f819 e8a24db8d8    call    lk!intel_pt_handle_vmx (ffffffff`998145c0)

  288 ffffffff`c0c8f81e f30fc775d0    vmxon   qword ptr [rbp-30h]

  139 ffffffff`c0c8f823 31c0          xor     eax,eax
  139 ffffffff`c0c8f825 803d980c030000  cmp     byte ptr [kvm_intel!nested_vmx_hardware_unse
  768 ffffffff`c0c8f82c 7538          jne     kvm_intel!hardware_enable+0x126 (ffffffff`c0c8f866)
```

VMXON



The first one who executes VMXON wins the hypervisor/king role! The second one is a traitor.

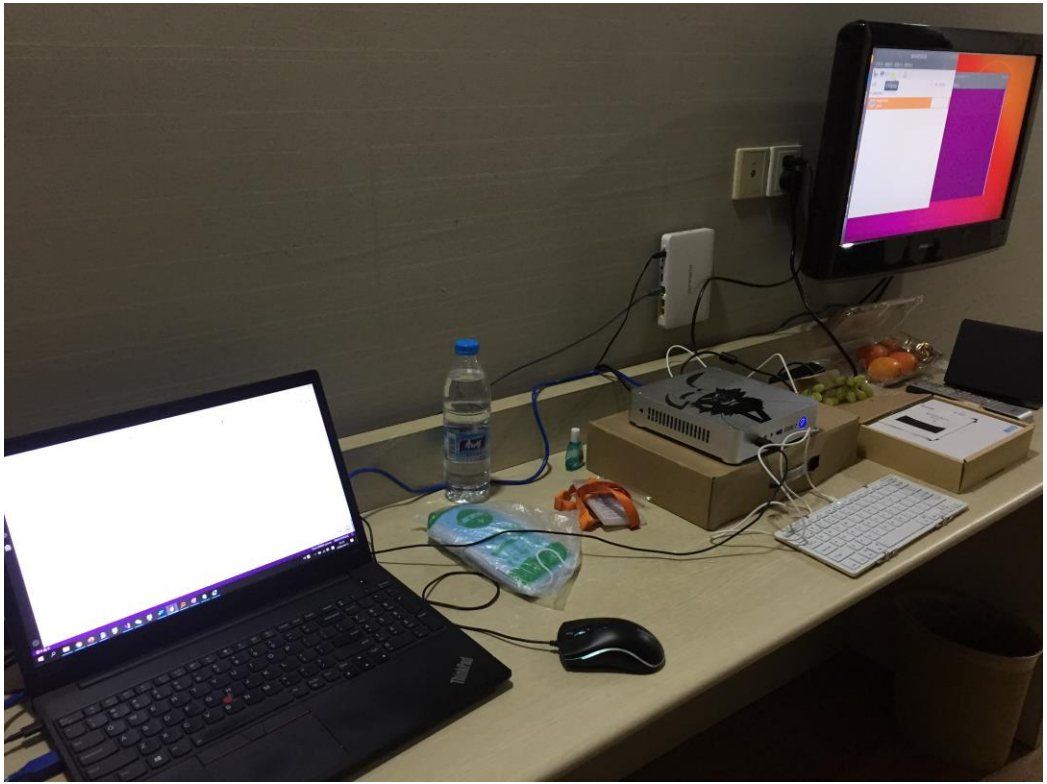
Examine in debugger

```
bp ffffffff`c0c8f81e
g
Breakpoint 0 hit
kvm_intel!hardware_enable+0xde:
0010:ffffffff`c0c8f81e f30fc775d0      vmxon   qword ptr [rbp-30h]
k
Child-SP          RetAddr           Call Site
ffffb232`80140ef0 ffffffff`c0aea599 kvm_intel!hardware_enable+0xfe [/build/lin
ffffb232`80140f30 ffffffff`c0ac98ba kvm!kvm_arch_hardware_enable+0x99 [/build/
ffffb232`80140f78 ffffffff`a51468ad kvm!hardware_enable_nolock+0x3a [/build/li
ffffb232`80140f98 ffffffff`a5147483 lk!flush_smp_call_function_queue+0x5d [/bu
ffffb232`80140fd0 ffffffff`a5c025be lk!generic_smp_call_function_single_interr
ffffb232`80140fe0 ffffffff`a5c01d0f lk!smp_call_function_interrupt+0x3e [/buil
ffffb232`80140ff8 00000000`00020001 lk!_paravirt_nop+0xd9 [/build/linux-hwe-p0
```

Linux Kernel


Quick view of KVM
DCI
Debug KVM using DCI

My debugging setup

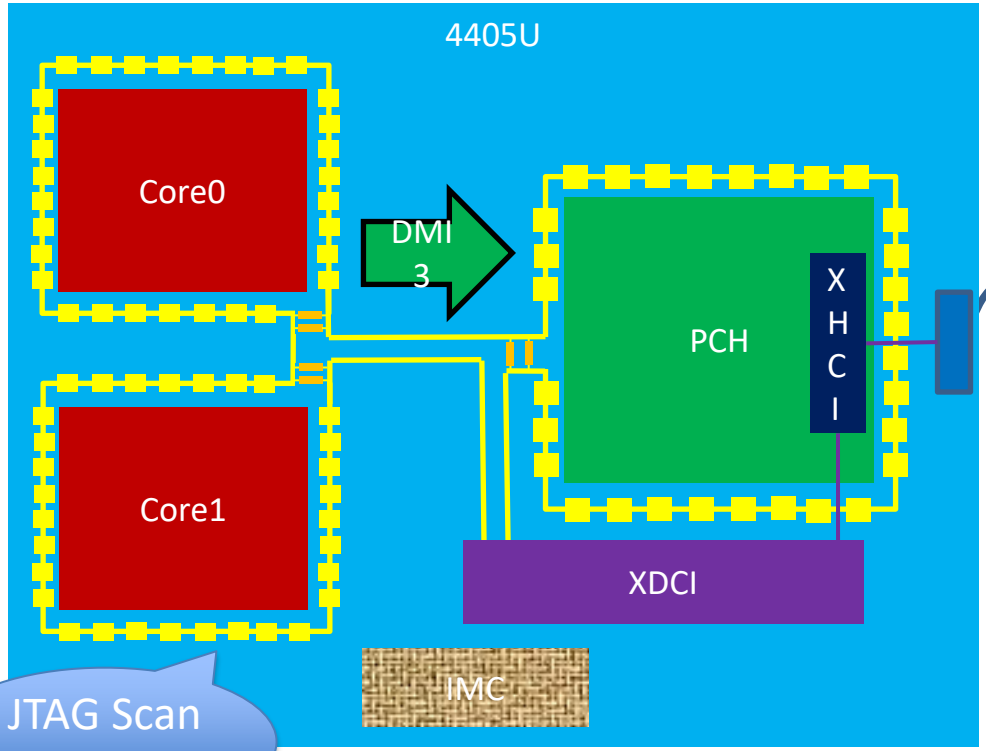


Host：
- Windows 10 PC

Target:
- Ubuntu 18.04
- KVM enabled
- A Ubuntu VM
- Pentium CPU(4405U)
- Customized BIOS

Connection:
- DCI – DbC USB3.0

DCI = Direct Connect Interface



USB3.0

DCI

调试主机

JTAG Scan Chain

Device List

IPython: C:IntelSWTools/system_studio_2020

```
-------------------------------------------------------------------------------------------------------
0    0x00004000  SPT0                SPT                C1   0x9A506013   -/-/ -/-  Yes
1    0x00004001  SPT_MASTER0          SPT_MASTER         C1   0x02080001   -/-/ -/-  Yes
2    0x00004004  SPT_MASTER_RETIME0   SPT_MASTER_RETIME  C1   0x00082017   -/-/ -/-  Yes
3    0x00004005  SPT_RGNLB0           SPT_RGNLB          C1   0x02080005   -/-/ -/-  Yes
4    0x00004008  SPT_AGG0             SPT_AGG            C1   0x0008000B   -/-/ -/-  Yes
5    0x00004009  SPT_CLTAP_RETIME0    SPT_CLTAP_RETIME   C1   0x0008000F   -/-/ -/-  Yes
6    0x00003000  SKL_U_UC0            SKL_U_UC           D0   0x3A76D013   0/-/ -/-  Yes
8    0x00005000  SKL_CBO0             SKL_CBO            D0                0/-/ -/-  Yes
10   0x00005001  SKL_CBO1             SKL_CBO            D0                0/-/ -/-  Yes
12   0x00002000  SKL_CORE0            SKL_C              D0                0/-/ 0/-  Yes
13   0x00001000  SKL_C0_T0            SKL                D0                0/-/ 0/0  Yes
14   0x00001001  SKL_C0_T1            SKL                D0                0/-/ 0/1  Yes
16   0x00002001  SKL_CORE1            SKL_C              D0                0/-/ 1/-  Yes
17   0x00001002  SKL_C1_T0            SKL                D0                0/-/ 1/0  Yes
18   0x00001003  SKL_C1_T1            SKL                D0                0/-/ 1/1  Yes
19   0x00010000  GroupDomain          LogicalGroupDomain                   -/-/ -/-  Yes
20   0x00010001  GPC                  LogicalGroupCore                     -/-/ -/-  Yes
21   0x00011000  DebugPort0           Debugport                            -/-/ -/-  Yes
22   0x00014000  DCI_USB_DFX          InterfacePort                        -/-/ -/-  Yes
23   0x00016000  PinsInterface0       PinsInterface                        -/-/ -/-  Yes
24   0x00014001  DCI_RAW              InterfacePort                        -/-/ -/-  Yes
25   0x00014002  DCI_PACKETS          InterfacePort                        -/-/ -/-  Yes
26   0x00012000  JtagScanChain0       JTAGScanChain                        -/-/ -/-  Yes
27   0x00012001  JtagScanChain1       JTAGScanChain                        -/-/ -/-  Yes
28   0x00019000  dci_iosf             StatePortInterface                   -/-/ -/-  Yes
29   0x00014003  DCI_USB_DMA          InterfacePort                        -/-/ -/-  Yes
30   0x00014004  DCI_USB_TRACE        InterfacePort                        -/-/ -/-  Yes

In [3]:
```

Two types of DCI

Closed Chassis Adapter (CCA)

BSSB Hosted DCI

Pro: Debug early wake up
Con: not USB 3 speed

USB Hosted DCI

Pro: low cost
Con: S0 only

IA32_DEBUG_INTERFACE_MSR (0xC80)

12.1.452 (C80h) IA32_DEBUG_INTERFACE_MSR

This register provides controls to enable/disable and lock different processor debug features. CPUID.(EAX=1):ECX[11] when set indicates the availability of this MSR.

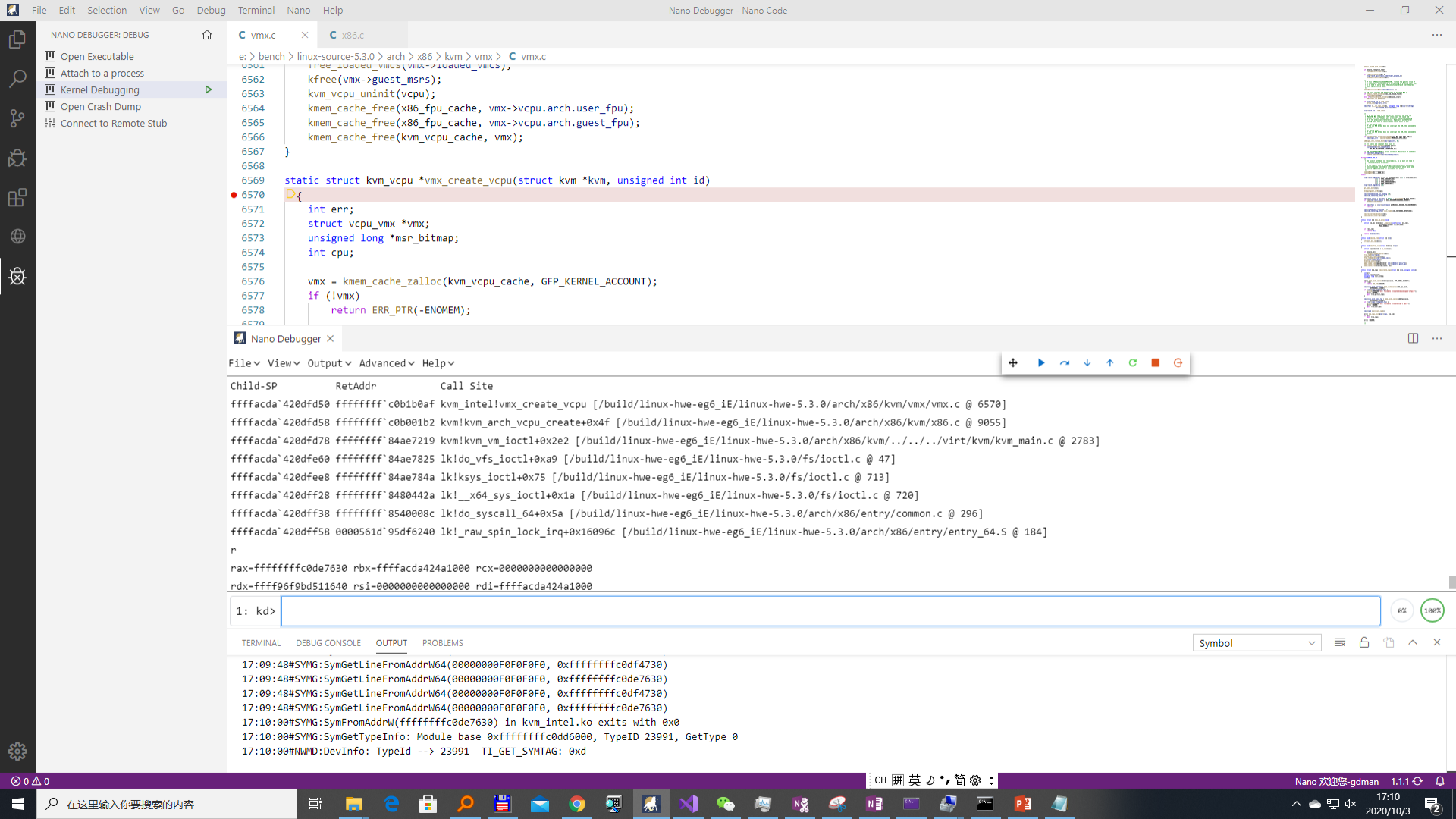
MSR Address: C80h				
Bit	**Scope**	**Default**	**Attribute**	**Description**
63:32	-	-	-	**RSVD_63_32**—Reserved
31	Package	-	RO	**DEBUG_OCCURRED**—This sticky bit is set by hardware to indicate the status of the enable bit. Note: On Skylake Server this bit status is retained in the RTC well through persitent PCH bit setting. On reboot the previous value is sent back to the processor through a reset message.
30	Package	-	RW	**LOCK**—When set locks any further changes to enable bit Note: The lock bit is set automatically on the first SMI assertion even if not explicitly set by BIOS
29:1	-	-	-	**RSVD_29_1**—Reserved
0	Package	-	RW	**ENABLE**—When set enables the debug features

For usual commercial machine, BIOS locks it


Quick view of KVM
DCI
Debug KVM using DCI

VM Create

- Create virtual CPU
- Create virtual MMU
- Create local APIC
- Hyper-V emulation
- Programmable Interrupt Timer

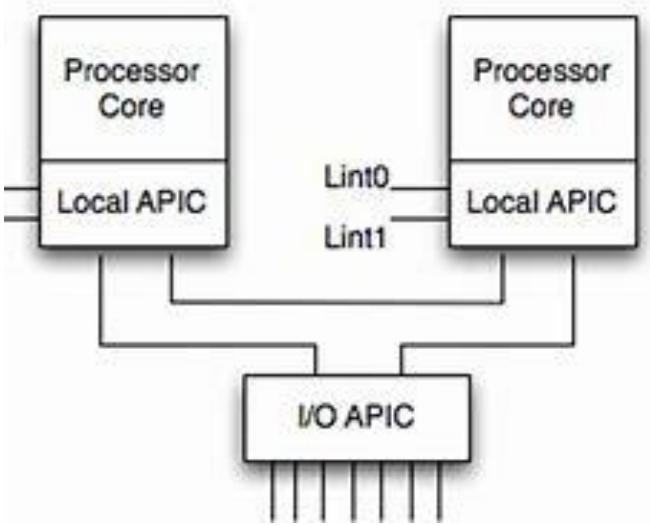

File Edit Selection View Go Debug Terminal Nano Help
Nano Debugger - Nano Code
NANO DEBUGGER: DEBUG
Open Executable
Attach to a process
Kernel Debugging
Open Crash Dump
Connect to Remote Stub
vmx.c
x86.c
e: > bench > linux-source-5.3.0 > arch > x86 > kvm > vmx > vmx.c
6562 kfree(vmx->guest_msrs);
6563 kvm_vcpu_uninit(vcpu);
6564 kmem_cache_free(x86_fpu_cache, vmx->vcpu.arch.user_fpu);
6565 kmem_cache_free(x86_fpu_cache, vmx->vcpu.arch.guest_fpu);
6566 kmem_cache_free(kvm_vcpu_cache, vmx);
6567 }
6568
6569 static struct kvm_vcpu *vmx_create_vcpu(struct kvm *kvm, unsigned int id)
6570 {
6571 int err;
6572 struct vcpu_vmx *vmx;
6573 unsigned long *msr_bitmap;
6574 int cpu;
6575
6576 vmx = kmem_cache_zalloc(kvm_vcpu_cache, GFP_KERNEL_ACCOUNT);
6577 if (!vmx)
6578 return ERR_PTR(-ENOMEM);
Nano Debugger
File View Output Advanced Help
Child-SP RetAddr Call Site
ffffacda`420dfd50 ffffffff`c0b1b0af kvm_intel!vmx_create_vcpu [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kvm/vmx/vmx.c @ 6570]
ffffacda`420dfd58 ffffffff`c0b001b2 kvm!kvm_arch_vcpu_create+0x4f [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kvm/x86.c @ 9055]
ffffacda`420dfd78 ffffffff`84ae7219 kvm!kvm_vm_ioctl+0x2e2 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kvm/../../../virt/kvm/kvm_main.c @ 2783]
ffffacda`420dfe60 ffffffff`84ae7825 lk!do_vfs_ioctl+0xa9 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/fs/ioctl.c @ 47]
ffffacda`420dfee8 ffffffff`84ae784a lk!ksys_ioctl+0x75 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/fs/ioctl.c @ 713]
ffffacda`420dff28 ffffffff`8480442a lk!__x64_sys_ioctl+0x1a [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/fs/ioctl.c @ 720]
ffffacda`420dff38 ffffffff`8540008c lk!do_syscall_64+0x5a [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/entry/common.c @ 296]
ffffacda`420dff58 0000561d`95df6240 lk!_raw_spin_lock_irq+0x16096c [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/entry/entry_64.S @ 184]
r
rax=ffffffffc0de7630 rbx=ffffacda424a1000 rcx=0000000000000000
rdx=ffff96f9bd511640 rsi=0000000000000000 rdi=ffffacda424a1000
1: kd>
0%
100%
TERMINAL DEBUG CONSOLE OUTPUT PROBLEMS
Symbol
17:09:48#SYMG:SymGetLineFromAddrW64(00000000F0F0F0F0, 0xffffffffc0df4730)
17:09:48#SYMG:SymGetLineFromAddrW64(00000000F0F0F0F0, 0xffffffffc0de7630)
17:09:48#SYMG:SymGetLineFromAddrW64(00000000F0F0F0F0, 0xffffffffc0df4730)
17:09:48#SYMG:SymGetLineFromAddrW64(00000000F0F0F0F0, 0xffffffffc0de7630)
17:10:00#SYMG:SymFromAddrW(ffffffffc0de7630) in kvm_intel.ko exits with 0x0
17:10:00#SYMG:SymGetTypeInfo: Module base 0xffffffffc0dd6000, TypeID 23991, GetType 0
17:10:00#NWMD:DevInfo: TypeId --> 23991 TI_GET_SYMTAG: 0xd
0 0
Nano 欢迎您-gdman 1.1.1
在这里输入你要搜索的内容
17:10
2020/10/3

Creating Virtual MMU

- Memory Management Unit
- Address Translation

```
Child-SP          RetAddr           Call Site
ffff9e0a`02c4bca0 ffffffff`c08ccd63 kvm!kvm_mmu_create(
                                    struct kvm_vcpu * vcpu = 0xffff8e73`25740000)
ffff9e0a`02c4bca8 ffffffff`c08a9a8f kvm!kvm_arch_vcpu_init(
                                    struct kvm_vcpu * vcpu = 0xffff8e73`25740000)+0x93
ffff9e0a`02c4bcc8 ffffffff`c0bfe6e5 kvm!kvm_vcpu_init(
                                    struct kvm_vcpu * vcpu = 0xffff8e73`25740000,
                                    struct kvm * kvm = 0xffff9e0a`02c01000,
                                    unsigned int id = 0)+0xcf
ffff9e0a`02c4bcf8 ffffffff`c08cc0af kvm_intel!vmx_create_vcpu(
                                    struct kvm * kvm = 0xffff8e73`25740000,
                                    unsigned int id = 0x2c01000)+0xb5
ffff9e0a`02c4bd58 ffffffff`c08b11b2 kvm!kvm_arch_vcpu_create(
                                    struct kvm * kvm = 0xffff8e73`25740000)+0x4f
ffff9e0a`02c4bd78 ffffffff`916e7219 kvm!kvm_vm_ioctl(
                                    unsigned int ioctl = 0x25740000,
                                    long unsigned int arg = 0n-107709143707648)+0x2e2
ffff9e0a`02c4be60 ffffffff`916e7825 lk!do_vfs_ioctl(
                                    struct file * filp = 0xffff8e73`25740000,
                                    long unsigned int arg = 0n2111471)+0xa9
ffff9e0a`02c4bee8 ffffffff`916e784a lk!ksys_ioctl(
                                    unsigned int fd = 0x2c01000,
                                    unsigned int cmd = 0x25740000,
                                    long unsigned int arg = 0n0)+0x75
ffff9e0a`02c4bf28 ffffffff`9140442a lk!__x64_sys_ioctl(void)+0x1a
ffff9e0a`02c4bf38 ffffffff`9200008c lk!do_syscall_64(
                                    struct pt_regs * regs = 0x00000000`002020ff)+0x5a
```

Create Local APIC

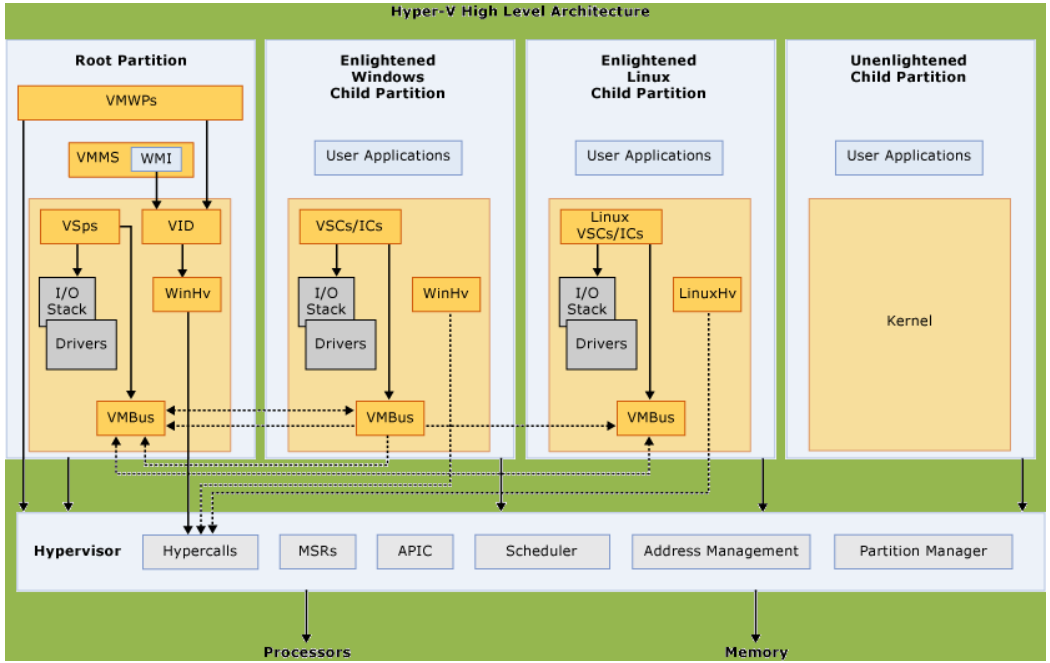


Call Site
kvm!kvm_create_lapic
kvm!kvm_arch_vcpu_init
kvm!kvm_vcpu_init
kvm_intel!vmx_create_vcpu
kvm!kvm_arch_vcpu_create
kvm!kvm_vm_ioctl

```
vcpu->arch.apic = apic;

apic->vcpu = vcpu;
```

KVM Microsoft Hyper-V emulation



kvm!kvm_hv_vcpu_postcreate
kvm!kvm_arch_vcpu_postcreate
kvm!kvm_vm_ioctl
lk!do_vfs_ioctl
lk!ksys_ioctl
lk!__x64_sys_ioctl
lk!do_syscall_64

Programmable Interval Timer (PIT)

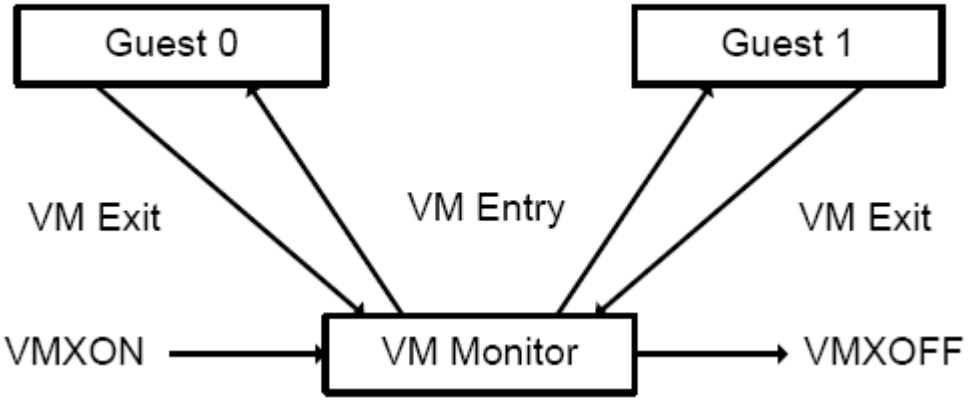
- Run in qemu-system-x86

Call Site
kvm!create_pit_timer.part.6
kvm!pit_load_count
kvm!kvm_pit_load_count
kvm!kvm_arch_vm_ioctl

```
kvm!kvm_create_pit [/build/linux-
hwe-eg6_iE/linux-hwe-
5.3.0/arch/x86/kvm/i8254.c @ 649]
```

VM Exit

- It doesn’t mean VM shutdown
- VM exits when it executes sensitive instruction
  - I/O access
  - Some page fault
  - Exception



Exit for I/O

- Primary way to stop VM destroy hardware

```
 # Child-SP          RetAddr           Call Site
00 ffff9e0a`02c4bc70 ffffffff`c0bf14aa kvm!kvm_fast_pio [/build/linux-hwe-eg6_iE/linux-hwe-5.3.
01 ffff9e0a`02c4bc78 ffffffff`c0bfee15 kvm_intel!handle_io+0x4a [/build/linux-hwe-eg6_iE/linux-h
02 ffff9e0a`02c4bc90 ffffffff`c08c65b8 kvm_intel!vmx_handle_exit+0xa5 [/build/linux-hwe-eg6_iE
03 ffff9e0a`02c4bcd0 cccccccc`cccccccc kvm!vcpu_enter_guest+0x4c8 [/build/linux-hwe-eg6_iE/li
```

x86.c

e: > bench > linux-source-5.3.0 > arch > x86 > kvm > x86.c

```
5688    gpa = vcpu->mmio_fragments[0].gpa;
5689
5690    vcpu->mmio_needed = 1;
5691    vcpu->mmio_cur_fragment = 0;
5692
5693    vcpu->run->mmio.len = min(8u, vcpu->mmio_fragments[0].len);
5694    vcpu->run->mmio.is_write = vcpu->mmio_is_write = ops->write;
5695    vcpu->run->exit_reason = KVM_EXIT_MMIO;
5696    vcpu->run->mmio.phys_addr = gpa;
5697
5698    return ops->read_write_exit_mmio(vcpu, gpa, val, bytes);
5699 }
5700
5701 static int emulator_read_emulated(struct x86_emulate_ctxt *ctxt,
5702                                   unsigned long addr,
5703                                   void *val,
5704                                   unsigned int bytes,
5705                                   struct x86_exception *exception)
```

Nano Home | Nano Debugger

File View Output Advanced Help

```
Child-SP          RetAddr           Call Site
ffffa660`4210f9d8 ffffffff`c0ac7546 kvm!write_exit_mmio [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kvm/x86.c @ 5574]
ffffa660`4210f9e0 ffffffff`c0ac75d5 kvm!emulator_read_write+0x126 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kvm/x86.c @ 5699]
ffffa660`4210fa30 ffffffff`c0aed99f kvm!emulator_write_emulated+0x15 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kvm/x86.c @ 5719]
ffffa660`4210fa40 ffffffff`c0aee328 kvm!segmented_write+0x5f [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kvm/emulate.c @ 1493]
ffffa660`4210fa80 ffffffff`c0af23a5 kvm!writeback+0x1c8 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kvm/emulate.c @ 1853]
ffffa660`4210fad0 ffffffff`c0ad2c78 kvm!x86_emulate_insn+0x625 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kvm/emulate.c @ 5771]
ffffa660`4210fb20 ffffffff`c0adfbde kvm!x86_emulate_instruction+0x338 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kvm/x86.c @ 6688]
ffffa660`4210fb88 ffffffff`c0b54dbe kvm!kvm_mmu_page_fault+0x47e [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kvm/mmu.c @ 5561]
ffffa660`4210fc68 ffffffff`c0b61e15 kvm_intel!handle_ept_misconfig+0x5e [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kvm/vmx/vmx.c @ 5124]
ffffa660`4210fc90 ffffffff`c0ace5b8 kvm_intel!vmx_handle_exit+0xa5 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kvm/vmx/vmx.c @ 5864]
ffffa660`4210fcd0 cccccccc`cccccc4 kvm!vcpu_enter_guest+0x4c8 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kvm/x86.c @ 8230]
```

2: kd>

TERMINAL DEBUG CONSOLE OUTPUT PROBLEMS

Nano Code

```
16:47:34#EXDI:vread 0xccccccccccccccc4 - elements 128 width 1 exits with hr=0xee000006, read=0 [REAL]
16:47:34#EVCB:ChangeDebuggeeState callback with 0x4, 0x2
16:47:34#NB:Getting BP by source for x86.c, 0
16:47:37#NB:Getting BP by source for x86.c, 0
16:47:37#NB:Getting BP by source for x86.c, 0
16:47:43#JTAG:The 64-bit paging PML4E is not valid. Address can't be translated.
16:47:43#EXDI:vread 0xccccccccccccccc4 - elements 128 width 1 exits with hr=0xee000006, read=0 [REAL]
```

Two types of I/O

(1) PIO

- Port IO
- Classic PC Ports

(2) MIO

- Memory Mapped IO
- More common

Register I/O handler

```
int register_ioport_read(pio_addr_t start, int length, int
size,IOPortReadFunc *func, void *opaque);

int register_ioport_write(pio_addr_t start, int length, int size,
IOPortWriteFunc *func, void *opaque);
```

- The main job to do device emulation

kvm_io_bus

```
dt bus -r
Local var @ r14 Type kvm_io_bus*
   +0x000 dev_count        : 0n4
   +0x004 ioeventfd_count  : 0n0
   +0x008 range            : [0]kvm_io_range[]
kvm_io_range
      +0x000 addr           : 0x20
      +0x008 len            : 0n2
      +0x010 dev            : 0xffff8e72`dd084960
kvm_io_device
         +0x000 ops         : 0xffffffff`c0905cc0
kvm_io_device_ops
```

Dispatch/Service I/O Access

```
Child-SP          RetAddr           Call Site
ffff9e0a`02c4bbd0 ffffffff`c08bc30e kvm!kvm_io_bus_read(
                                    struct kvm_vcpu * vcpu = 0xffff8e73`25740000,
                                    kvm_bus bus_idx = KVM_PIO_BUS (0n1),
                                    gpa_t addr = 0x71,
                                    int len = 0n1,
                                    void * val = 0xffff8e73`21174000)
ffff9e0a`02c4bbd8 ffffffff`c08bd714 kvm!kernel_pio(
                                    struct kvm_vcpu * vcpu = 0xffff8e73`25740000,
                                    void * pd = 0xffff8e73`21174000)+0x2e
ffff9e0a`02c4bc00 ffffffff`c08c4594 kvm!emulator_pio_in_emulated(
                                    struct x86_emulate_ctxt * ctxt = 0xffff8e73`25740000,
                                    int size = 0n1,
                                    unsigned int count = 0x21174000)+0x84
ffff9e0a`02c4bc40 ffffffff`c0bf14aa kvm!kvm_fast_pio(
                                    struct kvm_vcpu * vcpu = 0xffff8e73`25740000,
                                    short unsigned int port = 0)+0x54
ffff9e0a`02c4bc78 ffffffff`c0bfee15 kvm_intel!handle_io(
                                    struct kvm_vcpu * vcpu = 0xffff8e73`25740000)+0x4a
ffff9e0a`02c4bc90 ffffffff`c08c65b8 kvm_intel!vmx_handle_exit(
                                    struct kvm_vcpu * vcpu = 0xffff8e73`25740000)+0xa5
Amd64VtoP: Virt ccccccccccccccc4, pagedir 00000258bc03d000
Amd64VtoP: Non-canonical address
ffff9e0a`02c4bcd0 cccccccc`ccccccc4 kvm!vcpu_enter_guest(
                                    struct kvm_vcpu * vcpu = 0xffff8e73`25740000)+0x4c8
```

Exit for MMIO

Call Site
kvm!apic_mmio_read
kvm!vcpu_mmio_read
kvm!emulator_read_write_onepage
kvm!emulator_read_write
kvm!emulator_read_emulated
kvm!segmented_read
kvm!x86_emulate_insn
kvm!x86_emulate_instruction
kvm!kvm_mmu_page_fault
kvm_intel!handle_ept_misconfig
kvm_intel!vmx_handle_exit
kvm!vcpu_enter_guest

APIC Emulation

```
1367  static int apic_mmio_read(struct kvm_vcpu *vcpu, struct kvm_io_device *this,
1368                            gpa_t address, int len, void *data)
1369  {
1370      struct kvm_lapic *apic = to_lapic(this);
1371      u32 offset = address - apic->base_address;
1372
1373      if (!apic_mmio_in_range(apic, address))
1374          return -EOPNOTSUPP;
1375
1376      if (!kvm_apic_hw_enabled(apic) || apic_x2apic_mode(apic)) {
1377          if (!kvm_check_has_quirk(vcpu->kvm,
1378                                   KVM_X86_QUIRK_LAPIC_MMIO_HOLE))
1379              return -EOPNOTSUPP;
1380
1381          memset(data, 0xff, len);
1382          return 0;
1383      }
1384
1385      kvm_lapic_reg_read(apic, offset, len, data);
1386
1387      return 0;
```

Service MMIO Write by APIC

```
Child-SP          RetAddr           Call Site
ffff9e0a`02c4b928 ffffffff`c08be89a kvm!apic_mmio_write(
                                    struct kvm_vcpu * vcpu = 0xffff8e73`25740000,
                                    struct kvm_io_device * this = 0xffff8e73`22455008,
                                    gpa_t address = 0xfc097024,
                                    int len = 0n4,
                                    void * data = 0xffff8e73`25741a40)+0xa
ffff9e0a`02c4b940 ffffffff`c08bf259 kvm!write_mmio(
                                    struct kvm_vcpu * vcpu = 0xffff8e73`25740000,
                                    gpa_t gpa = 0xffff8e73`22455008,
                                    int bytes = 0n-66490332,
                                    void * val = 0x00000000`00000004)+0x6a
ffff9e0a`02c4b980 ffffffff`c08bf4b0 kvm!emulator_read_write_onepage(
                                    void * val = 0x00000000`00000004,
                                    unsigned int bytes = 0x25741a40,
                                    struct kvm_vcpu * vcpu = 0x00000000`00000004,
                                    const read_write_emulator_ops * ops = 0xffff8e73`25740000)+0x119
ffff9e0a`02c4b9e0 ffffffff`c08bf5d5 kvm!emulator_read_write(
                                    struct x86_emulate_ctxt * ctxt = 0x00000000`00000004,
                                    long unsigned int addr = 0n4,
                                    void * val = 0xffff8e73`25741a40,
                                    unsigned int bytes = 0xfc097024,
                                    const read_write_emulator_ops * ops = 0xffff8e73`25740000)+0x90
ffff9e0a`02c4ba30 ffffffff`c08e599f kvm!emulator_write_emulated(void)+0x15
ffff9e0a`02c4ba40 ffffffff`c08e6328 kvm!segmented_write(
```

APIC Regsiter

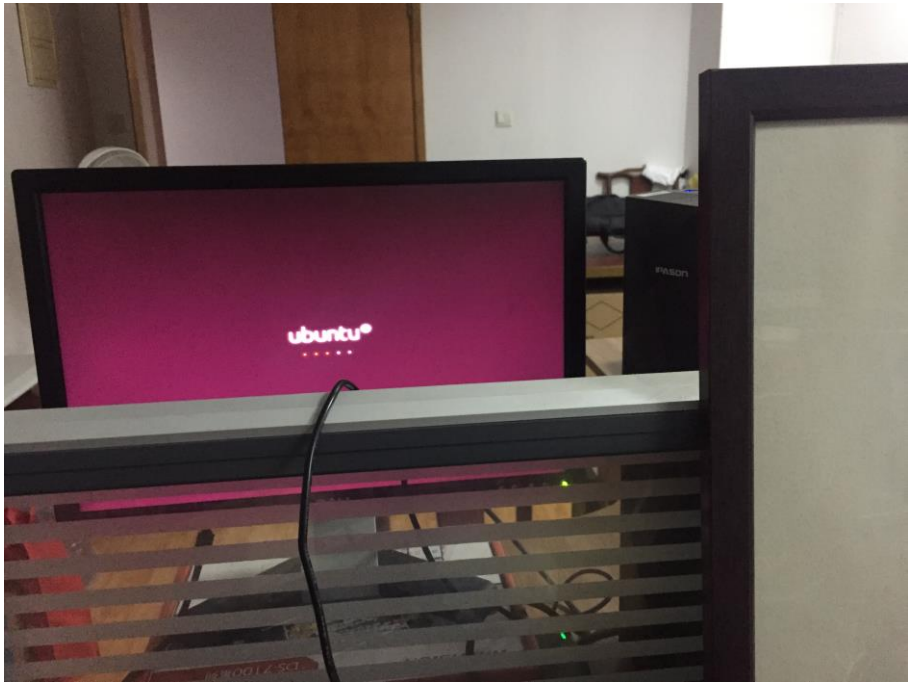
```
@rdi        vcpu = 0xffff8e73`25740000
@rsi        this = 0xffff8e73`22455008
@rdx        address = 0xfc097024
@rcx        len = 0n4
@r8         data = 0xffff8e73`25741a40
```

- Quite frequent

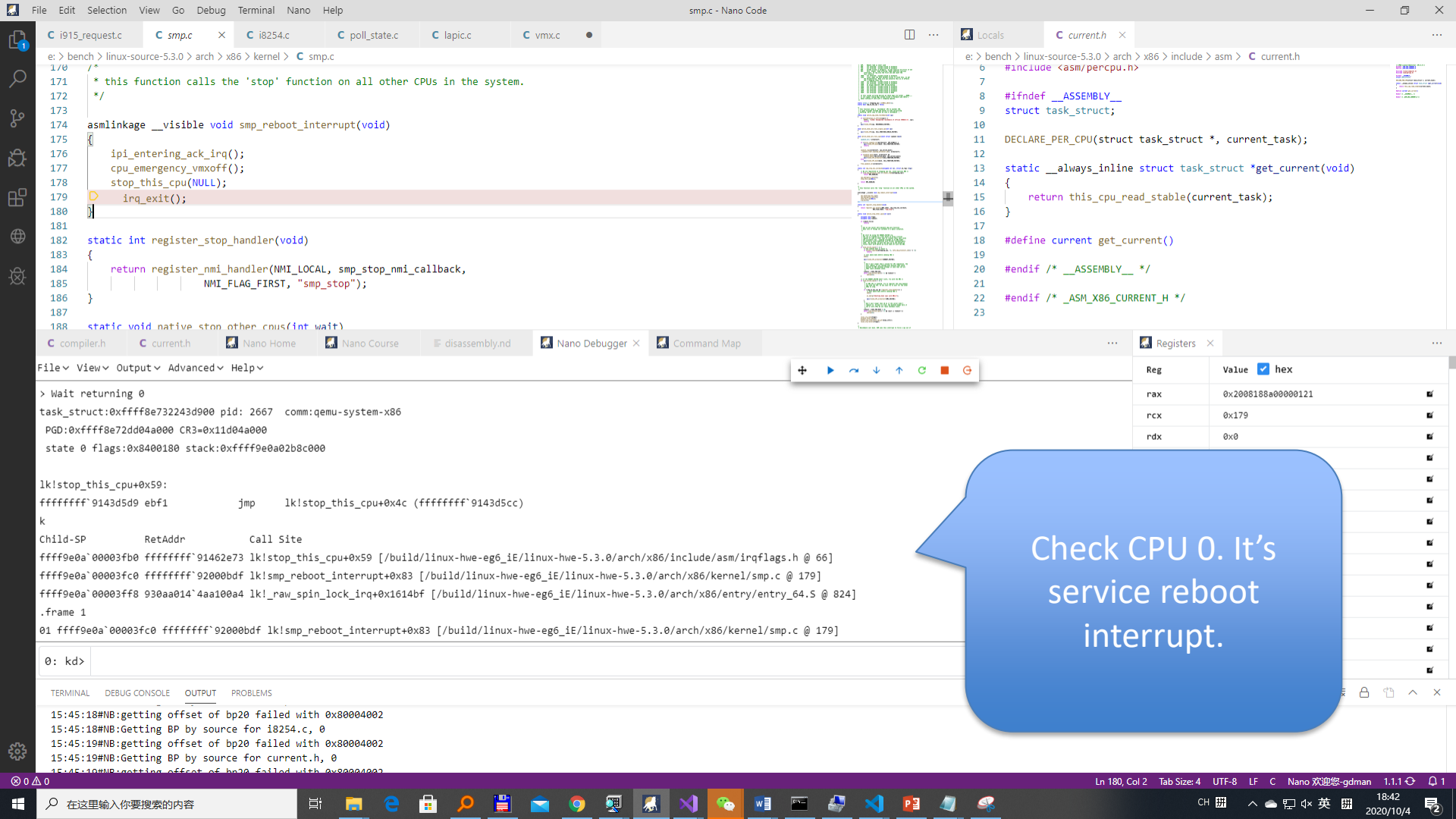
Useful Breakpoints

- bl
- 0 e fffffff`c0ad4060 0001 (0001) kvm!kvm_arch_vcpu_create
- 1 e fffffff`c0b5be50 0001 (0001) kvm_intel!alloc_vmcs_cpu
- 2 e fffffff`c0ad3740 0001 (0001) kvm!kvm_arch_exit
- 3 e fffffff`c0b22400 0001 (0001) kvm!kvm_arch_create_vcpu_
- 4 e fffffff`c0abc550 0001 (0001) kvm!kvm_vfio_ops_exit
- 5 e fffffff`c0ac2370 0001 (0001) kvm!write_exit_mmio
- 6 e fffffff`c0b6e730 0001 (0001) kvm_intel!handle_vmon
- 7 e fffffff`c0b61d70 0001 (0001) kvm_intel!vmx_handle_exit

A Real Case



- Ubuntu shutdown takes long time
- It seems hang somewhere
- Hard to debug


File Edit Selection View Go Debug Terminal Nano Help
smp.c - Nano Code
i915_request.c
smp.c
i8254.c
poll_state.c
lapic.c
vmx.c
e: > bench > linux-source-5.3.0 > arch > x86 > kernel > smp.c
/*
* this function calls the 'stop' function on all other CPUs in the system.
*/
asmlinkage __visible void smp_reboot_interrupt(void)
{
ipi_entering_ack_irq();
cpu_emergency_vmxoff();
stop_this_cpu(NULL);
irq_exit();
}
static int register_stop_handler(void)
{
return register_nmi_handler(NMI_LOCAL, smp_stop_nmi_callback,
NMI_FLAG_FIRST, "smp_stop");
}
static void native_stop_other_cpus(int wait)
Locals
current.h
e: > bench > linux-source-5.3.0 > arch > x86 > include > asm > current.h
#include <asm/percpu.h>
#ifndef __ASSEMBLY__
struct task_struct;
DECLARE_PER_CPU(struct task_struct *, current_task);
static __always_inline struct task_struct *get_current(void)
{
return this_cpu_read_stable(current_task);
}
#define current get_current()
#endif /* __ASSEMBLY__ */
#endif /* _ASM_X86_CURRENT_H */
compiler.h
current.h
Nano Home
Nano Course
disassembly.nd
Nano Debugger
Command Map
File View Output Advanced Help
> Wait returning 0
task_struct:0xffff8e732243d900 pid: 2667 comm:qemu-system-x86
PGD:0xffff8e72dd04a000 CR3=0x11d04a000
state 0 flags:0x8400180 stack:0xffff9e0a02b8c000
lk!stop_this_cpu+0x59:
ffffffff`9143d5d9 ebf1 jmp lk!stop_this_cpu+0x4c (ffffffff`9143d5cc)
k
Child-SP RetAddr Call Site
ffff9e0a`00003fb0 ffffffff`91462e73 lk!stop_this_cpu+0x59 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/include/asm/irqflags.h @ 66]
ffff9e0a`00003fc0 ffffffff`92000bdf lk!smp_reboot_interrupt+0x83 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kernel/smp.c @ 179]
ffff9e0a`00003ff8 930aa014`4aa100a4 lk!_raw_spin_lock_irq+0x1614bf [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/entry/entry_64.S @ 824]
.frame 1
01 ffff9e0a`00003fc0 ffffffff`92000bdf lk!smp_reboot_interrupt+0x83 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kernel/smp.c @ 179]
0: kd>
Registers
Reg
Value
hex
rax
0x2008188a00000121
rcx
0x179
rdx
0x0
Check CPU 0. It's service reboot interrupt.
TERMINAL
DEBUG CONSOLE
OUTPUT
PROBLEMS
15:45:18#NB:getting offset of bp20 failed with 0x80004002
15:45:18#NB:Getting BP by source for i8254.c, 0
15:45:19#NB:getting offset of bp20 failed with 0x80004002
15:45:19#NB:Getting BP by source for current.h, 0
Ln 180, Col 2
Tab Size: 4
UTF-8
LF
C
Nano 欢迎您-gdman
1.1.1
在这里输入你要搜索的内容
18:42
2020/10/4

```
170  /*
171   * this function calls the 'stop' function on all other CPUs in the system.
172   */
173
174  asmlinkage __visible void smp_reboot_interrupt(void)
175  {
176      ipi_entering_ack_irq();
177      cpu_emergency_vmxoff();
178      stop_this_cpu(NULL);
179      irq_exit();
180  }
```

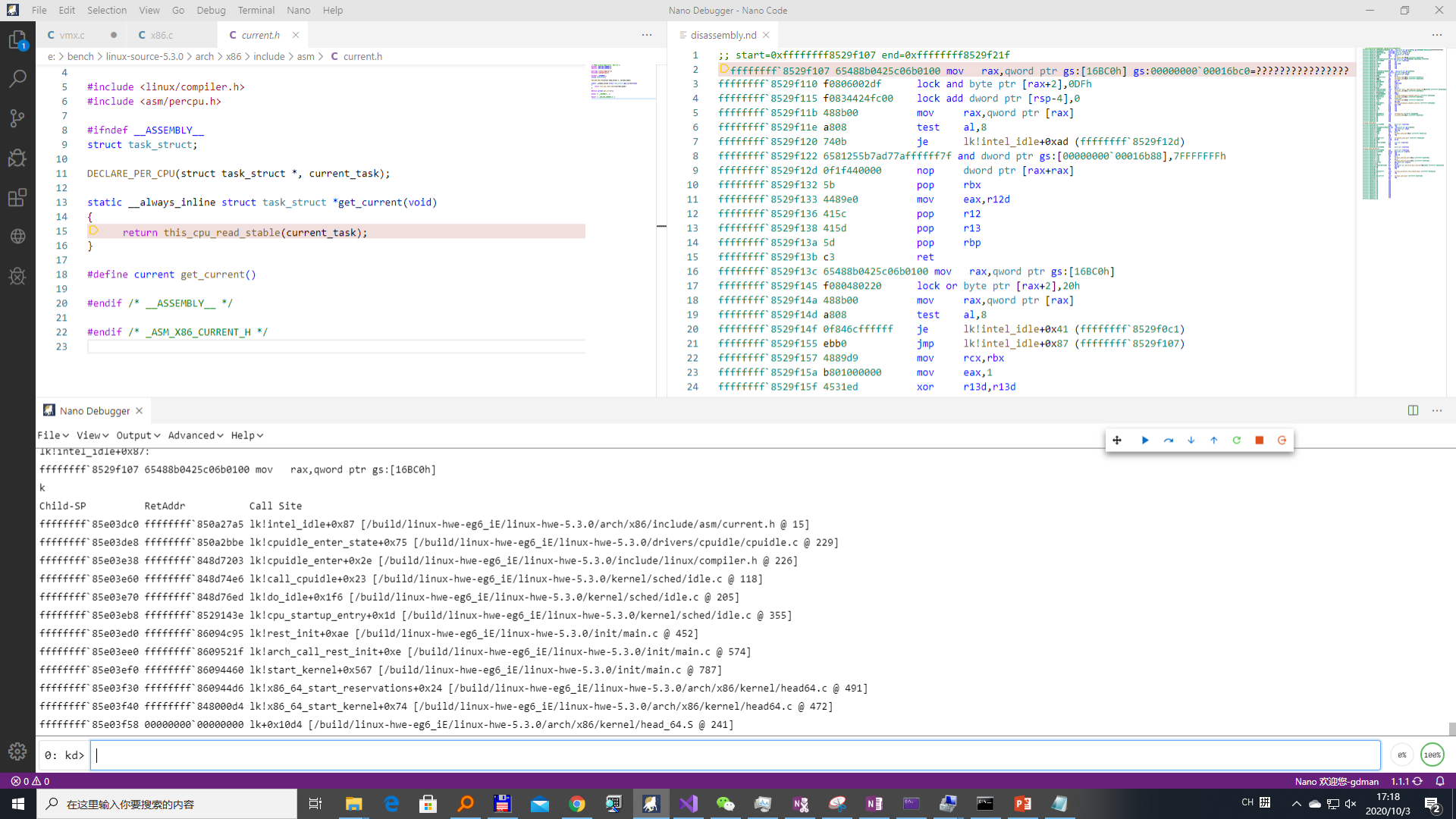
- Call stop on all other CPUs

CPU2 is in Panic

```
Child-SP          RetAddr           Call Site
ffff9e0a`00c4be48 ffffffff`91e8b426 lk!delay_tsc(void)+0x24
ffff9e0a`00c4be58 ffffffff`9149bb88 lk!__const_udelay(void)+0x46
ffff9e0a`00c4be68 ffffffff`9149b559 lk!panic(void)+0x2cc
ffff9e0a`00c4bef0 ffffffff`91539a89 lk!__stack_chk_fail(void)+0x19
ffff9e0a`00c4bf00 00000000`25b30f83 lk!__x64_sys_clock_gettime(void)+0xa9
```

- It’s in const delay.
- It might have cleared interrupt.


KVM
FORUM


File Edit Selection View Go Debug Terminal Nano Help
Nano Debugger - Nano Code
vmx.c
x86.c
current.h
disassembly.nd
e: > bench > linux-source-5.3.0 > arch > x86 > include > asm > current.h
#include <linux/compiler.h>
#include <asm/percpu.h>
#ifndef __ASSEMBLY__
struct task_struct;
DECLARE_PER_CPU(struct task_struct *, current_task);
static __always_inline struct task_struct *get_current(void)
{
return this_cpu_read_stable(current_task);
}
#define current get_current()
#endif /* __ASSEMBLY__ */
#endif /* _ASM_X86_CURRENT_H */
;; start=0xffffffff8529f107 end=0xffffffff8529f21f
ffffffff`8529f107 65488b0425c06b0100 mov rax,qword ptr gs:[16BC0h] gs:00000000`00016bc0=????????????????
ffffffff`8529f110 f0806002df lock and byte ptr [rax+2],0DFh
ffffffff`8529f115 f0834424fc00 lock add dword ptr [rsp-4],0
ffffffff`8529f11b 488b00 mov rax,qword ptr [rax]
ffffffff`8529f11e a808 test al,8
ffffffff`8529f120 740b je lk!intel_idle+0xad (ffffffff`8529f12d)
ffffffff`8529f122 6581255b7ad77affffff7f and dword ptr gs:[00000000`00016b88],7FFFFFFFh
ffffffff`8529f12d 0f1f440000 nop dword ptr [rax+rax]
ffffffff`8529f132 5b pop rbx
ffffffff`8529f133 4489e0 mov eax,r12d
ffffffff`8529f136 415c pop r12
ffffffff`8529f138 415d pop r13
ffffffff`8529f13a 5d pop rbp
ffffffff`8529f13b c3 ret
ffffffff`8529f13c 65488b0425c06b0100 mov rax,qword ptr gs:[16BC0h]
ffffffff`8529f145 f080480220 lock or byte ptr [rax+2],20h
ffffffff`8529f14a 488b00 mov rax,qword ptr [rax]
ffffffff`8529f14d a808 test al,8
ffffffff`8529f14f 0f846cffffff je lk!intel_idle+0x41 (ffffffff`8529f0c1)
ffffffff`8529f155 ebb0 jmp lk!intel_idle+0x87 (ffffffff`8529f107)
ffffffff`8529f157 4889d9 mov rcx,rbx
ffffffff`8529f15a b801000000 mov eax,1
ffffffff`8529f15f 4531ed xor r13d,r13d
Nano Debugger
File View Output Advanced Help
lk!intel_idle+0x87:
ffffffff`8529f107 65488b0425c06b0100 mov rax,qword ptr gs:[16BC0h]
k
Child-SP RetAddr Call Site
ffffffff`85e03dc0 ffffffff`850a27a5 lk!intel_idle+0x87 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/include/asm/current.h @ 15]
ffffffff`85e03de8 ffffffff`850a2bbe lk!cpuidle_enter_state+0x75 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/drivers/cpuidle/cpuidle.c @ 229]
ffffffff`85e03e38 ffffffff`848d7203 lk!cpuidle_enter+0x2e [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/include/linux/compiler.h @ 226]
ffffffff`85e03e60 ffffffff`848d74e6 lk!call_cpuidle+0x23 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/kernel/sched/idle.c @ 118]
ffffffff`85e03e70 ffffffff`848d76ed lk!do_idle+0x1f6 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/kernel/sched/idle.c @ 205]
ffffffff`85e03eb8 ffffffff`8529143e lk!cpu_startup_entry+0x1d [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/kernel/sched/idle.c @ 355]
ffffffff`85e03ed0 ffffffff`86094c95 lk!rest_init+0xae [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/init/main.c @ 452]
ffffffff`85e03ee0 ffffffff`8609521f lk!arch_call_rest_init+0xe [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/init/main.c @ 574]
ffffffff`85e03ef0 ffffffff`86094460 lk!start_kernel+0x567 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/init/main.c @ 787]
ffffffff`85e03f30 ffffffff`860944d6 lk!x86_64_start_reservations+0x24 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kernel/head64.c @ 491]
ffffffff`85e03f40 ffffffff`848000d4 lk!x86_64_start_kernel+0x74 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kernel/head64.c @ 472]
ffffffff`85e03f58 00000000`00000000 lk+0x10d4 [/build/linux-hwe-eg6_iE/linux-hwe-5.3.0/arch/x86/kernel/head_64.S @ 241]
0: kd>
0%
100%
Nano 欢迎您-gdman
1.1.1
在这里输入你要搜索的内容
CH
17:18
2020/10/3

Call Site
kvm!ioeventfd_write
kvm!__kvm_io_bus_write
kvm!kvm_io_bus_write
kvm_intel!handle_ept_misconfig
kvm_intel!vmx_handle_exit
kvm!vcpu_enter_guest

- Call Site
- vhost!translate_desc
  vhost!vhost_get_vq_desc
  vhost_net!handle_rx
  vhost_net!handle_rx_net
  vhost!vhost_worker lk!kthread