**Xen Interrupt分析**

[I/O Port Bitmap](http://zellux-notes.googlecode.com/hg/html/code/xen-interrupt.html?r=4e96fb27ec25c0a4863fa9c275157c79aac17e83" \l "sec1)

[[xen/include/xen/sched.h]](http://zellux-notes.googlecode.com/hg/html/code/xen-interrupt.html?r=4e96fb27ec25c0a4863fa9c275157c79aac17e83#sec2)

[[domain\_build.c]](http://zellux-notes.googlecode.com/hg/html/code/xen-interrupt.html?r=4e96fb27ec25c0a4863fa9c275157c79aac17e83#sec3)

[[linux-2.6.16-xen/drivers/xen/core/evtchn.c]](http://zellux-notes.googlecode.com/hg/html/code/xen-interrupt.html?r=4e96fb27ec25c0a4863fa9c275157c79aac17e83#sec4)

[[xen/common/event\_channel.c]](http://zellux-notes.googlecode.com/hg/html/code/xen-interrupt.html?r=4e96fb27ec25c0a4863fa9c275157c79aac17e83#sec5)

[Xen IDT initialization](http://zellux-notes.googlecode.com/hg/html/code/xen-interrupt.html?r=4e96fb27ec25c0a4863fa9c275157c79aac17e83#sec6)

[[xen/arch/x86/i8259.c]](http://zellux-notes.googlecode.com/hg/html/code/xen-interrupt.html?r=4e96fb27ec25c0a4863fa9c275157c79aac17e83#sec7)

[[xen/include/asm-x86/x86\_32/asm\_defns.h]](http://zellux-notes.googlecode.com/hg/html/code/xen-interrupt.html?r=4e96fb27ec25c0a4863fa9c275157c79aac17e83#sec8)

[[xen/arch/x86/x86\_32/entry.S]](http://zellux-notes.googlecode.com/hg/html/code/xen-interrupt.html?r=4e96fb27ec25c0a4863fa9c275157c79aac17e83#sec9)

**I/O Port Bitmap**

[xen/include/xen/sched.h]

**struct** **domain** { */\* I/O capabilities (access to IRQs and memory-mapped I/O). \*/* **struct** **rangeset** \*iomem\_caps; **struct** **rangeset** \*irq\_caps; **struct** **arch\_domain** arch { **struct** **rangeset** \*ioport\_caps; } }

[domain\_build.c]

ioports\_deny\_access(): remove caps from domain->arch.ioport\_caps

*/\* DOM0 is permitted full I/O capabilities. \*/* rc |= ioports\_permit\_access(dom0, 0, 0xFFFF); rc |= iomem\_permit\_access(dom0, 0UL, ~0UL); rc |= irqs\_permit\_access(dom0, 0, NR\_PIRQS-1); */\* \* Modify I/O port access permissions. \*/* */\* Master Interrupt Controller (PIC). \*/* rc |= ioports\_deny\_access(dom0, 0x20, 0x21); */\* Slave Interrupt Controller (PIC). \*/* rc |= ioports\_deny\_access(dom0, 0xA0, 0xA1); */\* Interval Timer (PIT). \*/* rc |= ioports\_deny\_access(dom0, 0x40, 0x43); */\* PIT Channel 2 / PC Speaker Control. \*/* rc |= ioports\_deny\_access(dom0, 0x61, 0x61); */\* Command-line I/O ranges. \*/* process\_dom0\_ioports\_disable();

[linux-2.6.16-xen/drivers/xen/core/evtchn.c]

startup\_pirq() calls HYPERVISOR\_event\_channel\_op() to register the IRQ

[xen/common/event\_channel.c]

Handling EVTCHNOP\_bind\_pirq hypercall

**Xen IDT initialization**

[xen/arch/x86/i8259.c]

**static** **void** (\***interrupt**[])(**void**) = { IRQLIST\_16(0x0), IRQLIST\_16(0x1), IRQLIST\_16(0x2), IRQLIST\_16(0x3), IRQLIST\_16(0x4), IRQLIST\_16(0x5), IRQLIST\_16(0x6), IRQLIST\_16(0x7), IRQLIST\_16(0x8), IRQLIST\_16(0x9), IRQLIST\_16(0xa), IRQLIST\_16(0xb), IRQLIST\_16(0xc), IRQLIST\_16(0xd), IRQLIST\_16(0xe), IRQLIST\_16(0xf) }; *// ...* **for** ( i = 0; i < NR\_IRQS; i++ ) { irq\_desc[i].status = IRQ\_DISABLED; irq\_desc[i].handler = &no\_irq\_type; irq\_desc[i].action = **NULL**; irq\_desc[i].depth = 1; spin\_lock\_init(&irq\_desc[i].lock); set\_intr\_gate(i, interrupt[i]); }

As to IRQLIST\_16 macro   
IRQLIST\_16(0x1) =>   
IRQ(0x1,0), IRQ(0x1,1), …, IRQ(0x1,f) =>   
IRQ0x10\_interrupt, IRQ0x11\_interrupt, …, IRQ0x1f\_interrupt

BUILD\_16\_IRQS(0x1) =>   
BI(0x1,0), …, BI(0x1, f) =>   
BUILD\_IRQ(0x10), …, BUILD\_IRQ(0x1f)

[xen/include/asm-x86/x86\_32/asm\_defns.h]

BUILD\_IRQ(0x1f) Macro:

*"\n"*\_\_ALIGN\_STR*"\n"* **IRQ0x1f\_interrupt**: **pushl** **$0x1f**<<16 **jmp** **common**\_interrupt **common\_interrupt**: **STR**(FIXUP\_RING0\_GUEST\_STACK) **STR**(SAVE\_ALL(a)) **movl** **%esp**,%eax **pushl** **%eax** **call** **do**\_IRQ **addl** **$4**,%esp **jmp** **ret**\_from\_intr

[xen/arch/x86/x86\_32/entry.S]

**ENTRY**(ret\_from\_intr) **GET\_CURRENT**(%ebx) **movl** **UREGS**\_eflags(%esp),%eax **movb** **UREGS**\_cs(%esp),%al **testl** **$**(3|X86\_EFLAGS\_VM),%eax **jnz** **test**\_all\_events **jmp** **restore**\_all\_xen