Question regarding the interrupt and interrupt handling

Could you please let me know how the hardware/driver raises an interrupt? and in an SMP, which CPU gets interrupted? If IRQ is shared by multiple devices, how the kernel identifies the which device caused the interrupt.

/Ganesh

linux linux-kernel interrupt interrupt-handling

edited Dec 10 '10 at 11:06

asked Dec 10 '10 at 7:52
Ganesh Kundapur
210 2 11

I'm interested in your question, but I'll need to warn you: you're being a little vague on "how the hardware/driver raises an interrupt." Simply put, the HW executes the code at a given interrupt vector. That's how it's done. I'm guessing you want something more specific. – San Jacinto Dec 10 '10 at 13:16

I mean whenever peripheral hardware needs CPU attention, it raises an interrupt, isn't it? CPU executes the interrupt handler for that interrupt. Now my question how the hardware raises an interrupt(By executing some instruction or ...) – Ganesh Kundapur Dec 10 '10 at 19:28

2 Answers

Traditionally there is an actual interrupt wire that runs from the device to the interrupt controller, when it is high (or low, or on an edge) an interrupt is generated and the CPU starts executing the interrupt handler.

On modern systems interrupts tend to be messages on a bus which are sent to the interrupt controller (or there may be several).

In terms of more detail you'll need to be more specific, the details vary depending on what sort of hardware you're talking about.

answered Dec 11 '10 at 13:26
mpe
1,647 8 13

Could you please let me know how the device generates interrupt on an x86 machine? – Ganesh Kundapur Dec 13 '10 at 7:40

It's not a question of x86 or not, it's PCI vs Hypertransport, vs on-chip devices etc. – mpe Dec 15 '10 at 3:37

How the device makes the interrupt line connected PCI high? – Ganesh Kundapur Dec 16 '10 at 8:05

Which CPU handles that interrupt in an SMP systems? – Ganesh Kundapur Dec 16 '10 at 8:05

The interrupt handler also called ISR is a part of device driver in OS. In OS, each irq number represents a interrupt line from the interrupt controller.

The devices are hard wired to the interrupt controller and the interrupt controller will signal the

corresponding CPU if there are interrupts generated in devices. The interrupt target CPUs are programmable in interrupt controller for each interrupt lines.

For sharing interrupt numbers in hardware, like GPIO in ARM, the interrupt controller or device should provide an additional register for the real interrupt number. In addition, the ISR should consult that register for the real IRQ number.

answered Jun 27 '12 at 8:21

jumper
166 5