How are interrupts handled by dual processor machines?

I have an idea of how interrupts are handled by a dual core CPU. I was wondering about how interrupt handling is implemented on a board with more than one physical processor.

Is any of the interrupt responsibility determined by the physical board's configuration? Each processor must be able to handle some types of interrupts, like disk I/O. Unless there is some circuitry to manage and dispatch interrupts to the appropriate processor? My guess is that the scheme must be processor neutral, so that any processor and core can run the interrupt handler.

If a core is waiting on a disk read, will that core be the one to run the interrupt handler when the disk is ready?

hardware cpu multicore interrupt cpu-architecture

edited May 10 '10 at 8:30
Jonas
24.6k 58 171 270

asked Mar 1 '09 at 4:05
jeffD
594 2 9 18

1 I see now that a core never waits on a read, a thread waits. When the read is finished though, it seems it would likely be better to run the newly woken thread on the same processor as before due to the chance for data still being cached on that processor. – jeffD Mar 1 '09 at 5:20

4 Answers

On x86 systems each CPU gets its own local APIC (Advanced Programmable Interrupt Controller) which are also wired to each other and to an I/O APIC that handles routing device interrupts to the local APICs.

The OS can program the APICs to determine which interrupts get routed to which CPUs (or to let the APICs make that decision).

I imagine that a multi-core CPU would have a local APIC for each core, but I'm honestly not certain about that.

See these links for more details:

- http://osdev.berlios.de/pic.html
- http://www.microsoft.com/whdc/archive/io-apic.mspx
- http://en.wikipedia.org/wiki/Intel_APIC_Architecture

answered Mar 1 '09 at 4:25
Michael Burr
194k 24 264 492

What you're interested in is SMP Processor Affinity. Here is an excellent article about how it is handled in Linux. The Advanced Programmable Interrupt Controller (APIC) is how you manage this in a modern system. Basically, the default would be to all go to processor 0 unless you had an OS that utilized this interface to set things up properly. Also, you don't necessarily want the core that issued a command to wait on a particular interrupt. You want the less loaded cores to receive it.

edited Mar 1 '09 at 5:17

answered Mar 1 '09 at 4:28
John Ellinwood
6,987 5 21 40

Thanks, the link on SMP in Linux was interesting. It's good to know that it's possible to modify servers in this way. – jeffD Mar 1 '09 at 5:05

I already asked this question a while back. Maybe it can offer you some insight :)

how do interrupts in multicore/multicpu machines work

answered Mar 1 '09 at 4:30
Giovanni Galbo
8,467 8 37 66

Useful info on how programmatic disk I/O is broken down into sector I/O operations and associated interrupt. – jeffD Mar 1 '09 at 5:23

I would say that it would depend on the hardware manufacturer...

However this link makes me believe most are probably handled by the primary processor and/or first core.

Another link

answered Mar 1 '09 at 4:17
uzbones
1,073 8 15