Linux Interrupt Handling in User Space

In Linux, what are the options for handling device interrupts in user space code rather than in kernel space?

linux interrupt-handling

asked Nov 2 '11 at 19:31

Brandon E Taylor
11.5k 2 23 44

3 Answers

Experience tells it is possible to write good and stable user-space drivers for almost any PCI adapter. It just requires some sophistication and a small proxying layer in the kernel. UIO is a step in that direction, but If you want to correctly handle interrupts in user-space then UIO might not be enough, for example if the device doesn't support the PCI-spec's interrupt disable bit which UIO relies on.

Notice that process wakeup latencies are a few microsecs so if your implementation requires very low latency then user-space might be a drag on it.

If I were to implement a user-space driver, I would reduce the kernel ISR to just a "disable & ack & wakeup-userpace" operation, handle the interrupt inside the waked-up process, and then re-enable the interrupt (of course, by writing to mapped PCI memory from the userspace process).

answered Nov 2 '11 at 22:23

Dan Aloni
2,252 6 18

There is UIO, but *handling* should still be done in kernelspace. OTOH, if you just need to notice the interrupt, you don't need the kernel part.

answered Nov 2 '11 at 20:05

ninjalj
23.5k 2 41 75

You may like to take a look at CHAPTER 10: Interrupt Handling from Linux Device Drivers, Third Edition book.

answered Nov 2 '11 at 20:21

Maxim Egorushkin
35.1k 4 40 74