

## Re: USB performance bug since kernel 2.6.13 (CRITICAL???)

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-10/msg04905.html>

---

- *From:* Open Source <[opensource3141@xxxxxxxxxx](mailto:opensource3141@xxxxxxxxxx)>
  - *Date:* Fri, 13 Oct 2006 12:31:07 -0700 (PDT)
- 

Hi Wolfgang (and all),

Thanks for the input. However, I am not understanding exactly why kernel mode is treated any differently than user mode for this sort of thing. I am looking at the code in ehci-q.c and ehci-hcd.c.

It seems like the unlinking of completed URBs happens asynchronously on a timer. This is a surprise to me since I thought this was happening on an IRQ from the host controller. But if what I'm surmising is correct it would explain everything I am seeing. I'm not able to ascertain how user mode drivers are treated differently than kernel mode drivers in this regard. From what I can tell, all drivers would be broken equally! Can anyone who has more experience with this code confirm this for me?

Besides, we count on sub-10 ms response times all the time in user mode. Take for example, the access of a file. If opening a file had a fixed latency of 4 ms, people would be up in arms. So that's not entirely a valid excuse. A USB operation that used to take 1 ms now takes 4 ms. That's a pretty big change.

The ability to write user-mode drivers for USB devices is very powerful for deployment. If one writes a kernel driver, there are severe deployment hassles. As such, my company has chosen to write user-mode drivers on both Windows to avoid driver deployment nightmares. This has been extremely successful so far.. Ironically, Windows (using libusb-win32) has had no such performance glitches. As a matter of principle, Linux should at least be as good as Windows, right?

Hopefully we can get this sorted out.

Re: USB performance bug since kernel 2.6.13 (CRITICAL???)

Cheers.

----- Original Message -----

From: WolfgangMües <wolfgang@xxxxxxxxxxxxxx>

To: linux-usb-devel@xxxxxxxxxxxxxxxxxxxxxxxx

Sent: Friday, October 13, 2006 12:11:08 PM

Subject: Re: [linux-usb-devel] USB performance bug since kernel 2.6.13 (CRITICAL???)

On Friday 13 October 2006 19:20, Open Source wrote:

Alan — yes, I understand the ability to increase throughput by transferring more bytes and I am definitely able to see better overall throughput when increasing the number of bytes per transaction. However, I needs to still have good transaction-level timing because I cannot always queue the transactions up. Recall that each transaction is a WRITE followed by a READ. The results of the READ determine the outgoing bytes for the following transaction's WRITE.

Relying on sub-10ms response times i