

Re: [PATCH/RFC] SPI: add DMAUNSAFE analog to David Brownell's core

# Re: [PATCH/RFC] SPI: add DMAUNSAFE analog to David Brownell's core

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-12/msg05174.html>

---

- *From:* Vitaly Wool <[vwool@xxxxxxxxxxxxxx](mailto:vwool@xxxxxxxxxxxxxx)>
  - *Date:* Fri, 16 Dec 2005 11:37:46 +0300
- 

Greg KH wrote:

On Fri, Dec 16, 2005 at 01:23:32AM +0300, Vitaly Wool wrote:

Greg KH wrote:

On Thu, Dec 15, 2005 at 09:47:42AM +0300, Vitaly Wool wrote:

David Brownell wrote:

No, "stupid drivers will suffer"; nothing new. Just observe how the ads7846 touchscreen driver does small async transfers.

Re: [PATCH/RFC] SPI: add DMAUNSAFE analog to David Brownell's core

One cannot allocate memory in interrupt context, so the way to go is allocating it on stack, thus the buffer is not DMA-safe.

Making it DMA-safe in thread that does the very message processing is a good way of overcoming this.

Using preallocated buffer is not a good way, since it may well be already used by another interrupt or not yet processed by the worker thread (or tasklet, or whatever).

Yes it is a good way. That's the way USB currently works in the kernel, and it works just fine. It keeps the rules simple and everyone knows what needs to be done.

Looking at my usbnet stuff, I can't share that opinion :-/  
Are you really ready to lower the performance and quality of service just for approach uniformity?

What performance issues? As an example, USB has this rule, and we can saturate a 480Mbit line with a `_userspace_` driver (loads of memcopy calles involved there.)

What CPU is used there? I guess it's not 144 MHz ARM ;-)

Re: [PATCH/RFC] SPI: add DMAUNSAFE analog to David Brownell's core

And, can you please point me out the examples of devices behind USB bus that need to write registers from an interrupt context?

usb to serial drivers need to allocate buffers for their write functions as they can be called in irq context from a tty line discipline, which causes a USB packet to be dynamically created and sent to the USB core. I also think the USB network and ATM drivers have these requirements too, just search for GFP\_ATOMIC in the drivers/usb/ directory to find these instances.

Oh BTW... I'm experiencing constant problems with root filesystem over NFS over usbnet on my target I'm getting "server not responding, still trying" error whenever the system (208-MHz ARM926 board) is under heavy load. I think it may well be related to the thing we discuss.

Vitaly

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>