

## Re: [linux-usb-devel] USB deadlock after resume

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-11/msg07566.html>

---

- *From:* "Markus Rechberger" <mrechberger@xxxxxxxx>
  - *Date:* Wed, 21 Nov 2007 23:22:23 +0100
- 

On 11/21/07, Laurent Pinchart <laurent.pinchart@xxxxxxxx> wrote:

On Wednesday 21 November 2007, Markus Rechberger wrote:

On 11/21/07, Alan Stern <stern@xxxxxxxxxxxxxxxxxxxx> wrote:

On Wed, 21 Nov 2007, Markus Rechberger wrote:

it's not just  
usb\_set\_interface  
that hangs  
actually.  
It seems to  
hang at

```
wait_event(usb_kill_urb_queue,  
atomic_read(&urb->use_count)  
== 0);
```

in  
drivers/usb/core/urb.c  
after  
resuming. I  
disabled  
access to  
the  
usb  
subsystem  
in the uvc  
driver,  
although  
connecting  
any other  
usb  
storage fails  
too, just at  
the same  
point.

Re: [linux-usb-devel] USB deadlock after resume

Which URB is  
usb\_kill\_urb() called for?

it's the usb\_control\_message which calls  
usb\_kill\_urb if I haven't got  
it wrong. (if you're looking for some other  
information please let me  
know)

Although, I got a bit further with it. The  
error seems to happen  
earlier already.

If I load the driver, and do not access the  
device after suspending  
all usb\_control commands fail with -71  
eproto.

That's very strange. Getting -71 errors is understandable; it  
indicates that the device can't handle being suspended. But  
the  
wait\_event() line still shouldn't hang. If it does, it indicates  
that  
there's something wrong with the USB host controller, not  
just the  
device.

Can you try testing this on a different sort of computer?

Not really, suspending doesn't work at all on my other notebook it  
just freezes..

I'm basically trying to get that driver work on my eee PC [1], it's  
cheap and tiny so I don't expect anything special in there..

The system is preloaded with Xandros (it's debian etch with a few  
custom applications) and linux 2.6.21.4.

If I'm not mistaken, the EeePC ACPI bios plays tricks with the USB ports  
during suspend/resume. You should really test suspend/resume with the same  
camera chipset on a proper computer. If the camera still crashes, we have a  
buggy chipset which needs a reset quirk. If it doesn't, the EeePC ACPI bios  
is probably at fault. Adding quirks and hacks to the Linux kernel (either in  
the USB stack or the uvcvideo driver) is pretty pointless if the bios tries  
to make the system crash. The ACPI code should be fixed in that case.

With ACPI it seems to be possible to disconnect the uvc device.  
I tested the suspend/resume functions by adding a proc interface to

Re: [linux-usb-devel] USB deadlock after resume

Re: [linux-usb-devel] USB deadlock after resume

it, and it worked properly.

Although the eee PC also suspends the underlying bus where the usb controller is connected to (which is PCI or PCIe)

The system still locks up, although only if I leave the video application running during suspending. I don't have to reload the driver anymore after resuming if the video node doesn't get accessed (I'm looking for races in the uvc driver at the moment).

The current state I revealed is that after suspend if the video node isn't used it's not necessary to reconnect the device nor to reload the driver again if that reset is implemented.

That eee PC comes with 2.6.21.3 which has no such reset quirk feature in the usbcore (that's what I initially meant actually).

If a videoapplication accesses the nodes during suspend the notebook won't come back again.

I also think it's faulty hardware in that case but I'm moreover looking for a solution for it. My other intel notebook doesn't even awake from suspend to ram, and for some reason suspend to disk just didn't work as expected either (Acer Travelmate 660).

thanks for the feedback,  
Markus

—

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

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

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