

Re: smartcard reader + pcmcia/pccard subsystem problems

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-02/msg06655.html>

- *From:* "Markus Rechberger" <mrechberger@xxxxxxxxx>
 - *Date:* Mon, 19 Feb 2007 11:40:27 +0100
-

Hi,

after doing some code review it seems like I found 2 bugs.

- a.) deadlock within the pccard/pcmcia framework
- b.) slab corruption when unloading the driver

a.) it's easy to reproduce a, just run

```
$ while ;; do pccardctl eject; pccardctl insert; done
```

concurrently on 2 shells.

another way to reproduce it is:

```
$ while ;; do insmod oz...ko; pccardctl insert; pccardctl eject; rmdir oz...ko; done
```

(it's sufficient to run that on one shell)

The bug is sysfs related, there are 2 locks one to protect critical parts of the pccard/pcmcia framework, the other one to lock the sysfs inode. I'll try to commit a patch for this during the next few days I was able to solve one problem there already.

b.) slab corruption; it's a bit difficult to reproduce since a. might lock up the device before anything shows up. So I haven't done much here yet.

Markus

On 2/17/07, Markus Rechberger <mrechberger@xxxxxxxxx> wrote:

Hi Eric,

I committed your code to linuxtv.org to review and modify it there.
<http://linuxtv.org/hg/~mrechberger/chipcardreader>

Re: smartcard reader + pcmcia/pccard subsystem problems

one thing I noticed is the error handling in ozscr_probe.

I'll continue the rest during the next few days, I'd like to see it as soon as possible in the upstream kernel before some kernel api changes again which affects your current driver.

Markus

On 2/17/07, Markus Rechberger <mrechberger@xxxxxxxxxx> wrote:

```
> Hi,
>
> so finally I'm also looking at that driver,
> http://pieleric.free.fr/o2scr/
> the driver compiles fine, though it doesn't seem to work (unless I'm
> doing something wrong here)
>
> dmesg shows up following entries:
>
> pccard: card ejected from slot 1
> PCMCIA: socket c160c364: *** DANGER *** unable to remove socket power
> pccard: PCMCIA card inserted into slot 1
> pcmcia: registering new device pcmcia1.0
> pccard: card ejected from slot 1
> PCMCIA: socket c160c364: *** DANGER *** unable to remove socket power
> pccard: PCMCIA card inserted into slot 1
> pcmcia: registering new device pcmcia1.0
> pccard: card ejected from slot 1
> PCMCIA: socket c160c364: *** DANGER *** unable to remove socket power
> pccard: PCMCIA card inserted into slot 1
> pcmcia: registering new device pcmcia1.0
> OZSCLX O2Micro SmartCardBus Reader (for kernel >= 2.6.17)
>
> The module for any reason has a usecount value of 1
> ozscr1x 21548 1
>
> devicenode /dev/ozscr1x isn't opened anywhere either.
>
> I'll do some further investigations upon it, I'd also like to see it
> directly in the kernel. It would be handy to use for encrypted
> filesystems.
>
> Markus
>
>
> On 12/12/06, Eric Piel <Eric.Piel@xxxxxxxxxxxxxxxxxxxx> wrote:
>> 28.11.2006 12:49, Oliver Neukum wrote/a écrit:
>>> Latest version I've published is there:
>>> http://pieleric.free.fr/o2scr/
>>>
>>> case OZSCR_OPEN: /* Request ICC */
>>> dprintk("OZSCR_OPEN\n");
```

Re: smartcard reader + pcmcia/pccard subsystem problems

```
>>> ATRLength = ATR_SIZE;
>>> pRdrExt->IOBase = (PSCR_REGISTERS *) dev->io_base; //XXX
>> necessary?
>>> pRdrExt->membase = dev->am_base; //XXX necessary?
>>>
>>> pRdrExt->m_SCard.AvailableProtocol = 0;
>>> pRdrExt->m_SCard.RqstProtocol = 0;
>>> dprintk("membase:%p\n", pRdrExt->membase);
>>> dprintk("ioport:0x%03x\n", (unsigned)pRdrExt->IOBase);
>>>
>>> ret = CmdResetReader( pRdrExt, FALSE, ATRBuffer,
&ATRLength
> );
>>> apdu.LengthOut = ATRLength;
>>>
>>> #ifdef PCMCIA_DEBUG
>>> printk(KERN_DEBUG "Open finished, ATR buffer = ");
>>> for( ATRLength = 0; ATRLength < apdu.LengthOut;
ATRLength++
> )
>>> printk(" [%02X] ", ATRBuffer[ATRLength] );
>>> printk("\n");
>>> #endif
>>>
>>> memcpy( apdu.DataOut, ATRBuffer, ATRLength );
>>> ret = copy_to_user((struct ozscr_apdu *)arg, &apdu,
>> sizeof(struct ozscr_apdu));
>>> break;
>>>
>>> 1. This needs locking against concurrent ioctls
>>> 2. The interpretation of copy_to_user()'s return code is incorrect
>>>
>>
>> Hi Oliver,
>>
>> Thanks a lot for reading my code, I didn't even hope that someone
would!
>> I've corrected the copy_to_user (and copy_from_user) code. However I
>> don't know how to do locking for the concurrent ioctls. Indeed, I don't
>> think there is anything preventing two programs to call the driver at
>> the same time. Unfortunately, I've got no idea how to do the locking
and
>> surprisingly couldn't find any ioctl code in the kernel doing locking.
>> Maybe I've just not looked at the right place, could you give a me some
>> hint how to do locking for ioctl's ?
>>
>> See you,
>> Eric
>>
>> -
>> To unsubscribe from this list: send the line "unsubscribe linux-kernel"
```

Re: smartcard reader + pcmcia/pccard subsystem problems

> 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/>
>>
>
>
> --
> Markus Rechberger
>

--
Markus Rechberger

--
Markus Rechberger

-
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/>