

---

Subject: Re: [PATCH] Fix inet\_diag.ko register vs rcv race  
Posted by [Arnaldo Carvalho de M](#) on Thu, 29 Nov 2007 12:47:17 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Em Thu, Nov 29, 2007 at 11:37:34PM +1100, Herbert Xu escreveu:  
> On Tue, Nov 27, 2007 at 04:09:43PM +0300, Pavel Emelyanov wrote:  
> > The following race is possible when one cpu unregisters the handler  
> > while other one is trying to receive a message and call this one:  
>  
> Good catch! But I think we need a bit more to close this fully.  
>  
> Dumps can resume asynchronously which means that they won't be  
> holding inet\_diag\_mutex. We can fix that pretty easily by  
> giving that as our cb\_mutex.  
>  
> So could you add that to your patch and resubmit?  
>  
> Arnaldo, synchronize\_rcu() doesn't work on its own. Whoever accesses  
> the object that it's supposed to protect has to use the correct RCU  
> primitives for this to work.  
>  
> Synchronisation is like tango, it always takes two to make it work :)

Agreed, I didn't checked that when refactoring inet\_diag, leaving this  
as it was before I put my hands on it :-)

- Arnaldo

---