

Re: poll() in 2.6 and beyond

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-03/0683.html>

From: Richard B. Johnson (root_at_chaos.analogic.com)

Date: 03/03/04

Date: Wed, 3 Mar 2004 13:23:26 -0500 (EST)

To: David Dillow <dave@thedillows.org>

On Tue, 2 Mar 2004, David Dillow wrote:

> On Tue, 2004-03-02 at 18:32, Richard B. Johnson wrote:
> > Yes. The code I attached earlier shows that the poll() in a driver
> > gets called (correctly), then it calls poll_wait(). Unfortunately
> > the call to poll_wait() returns immediately so that the return
> > value from the driver's poll() is whatever it was before some
> > event occurred that the driver was going to signal with
> > wake_up_interruptible().
>
> You've been handed a clue enough times now that you should understand
> that poll_wait() does not, and has never, put the process to sleep.
>
> If you can show a case where do_poll() returns stale data, then by all
> means do so. We will be happy to fix any such error in the kernel.
>
> You say do_poll() loses the status returned from your driver's poll
> method. If your driver is truly returning a nonzero status from the
> poll() method call, then a simple read of the code in do_pollfd() will
> show that the only way it loses information from that event mask is if
> your user space is not setting that event type in pollfd.events.
>
> If I were you, I'd check two things:
> 1) that your poll method is really returning a non-zero status when you
> think it is
> 2) that your user space program is really asking for all events you
> think it is
>
> I think you'll find your problem is not this well-used mechanism in the
> kernel.
>
> Dave

The very great problems that exist with poll on linux-2.6.0 are being quashed by those who just like to argue. Therefore, I wrote some code that emulates the environment in which I

Linux–Kernel: Re: poll() in 2.6 and beyond

discovered the poll failure. Experts can