Subject: Re: [PATCH] mark read_crX() asm code as volatile
Posted by Nick Piggin on Tue, 02 Oct 2007 12:14:54 GMT
View Forum Message <> Reply to Message

On Wednesday 03 October 2007 04:27, Chuck Ebbert wrote:
> On 10/02/2007 11:28 AM, Arjan van de Ven wrote:
> > On Tue, 02 Oct 2007 18:08:32 +0400
> >
> > Kirill Korotaev <dev@openvz.org> wrote:
> >> Some gcc versions (I checked at least 4.1.1 from RHEL5 & 4.1.2 from
> >> gentoo) can generate incorrect code with read_crX()/write_crX()
> >> functions mix up, due to cached results of read_crX().
> >
> > I'm not so sure volatile is the right answer, as compared to giving the
> > asm more strict contraints....
> >
> > asm volatile tends to mean something else than "the result has
> > changed"....
>
> It means "don't eliminate this code if it's reachable" which should be
> just enough for this case. But it could still be reordered in some cases
> that could break, I think.
>
> This should work because the result gets used before reading again:
>
> read_cr3(a);
> write_cr3(a | 1);
> read_cr3(a);
>
> But this might be reordered so that b gets read before the write:
>
> read_cr3(a);
> write_cr3(a | 1);
> read_cr3(b);
>
> ?

I don't see how, as write_cr3 clobbers memory.