Subject: Re: [RFC] kernel/pid.c pid allocation wierdness
Posted by Oleg Nesterov on Wed, 14 Mar 2007 15:33:41 GMT
View Forum Message <> Reply to Message

On 03/14, Eric W. Biederman wrote:
> Pavel Emelianov <xemul@sw.ru> writes:
>
> > Hi.
> >
> > I'm looking at how alloc_pid() works and can't understand
> > one (simple/stupid) thing.
> >
> > It first kmem_cache_alloc()-s a strct pid, then calls
> > alloc_pidmap() and at the end it taks a global pidmap_lock()
> > to add new pid to hash.

We need some global lock. pidmap_lock is already here, and it is only used to protect pidmap->page allocation. Iow, it is almost unused. So it was very natural to re-use it while implementing pidrefs.

> > The question is - why does alloc_pidmap() use at least
> > two atomic ops and potentially loop to find a zero bit
> > in pidmap? Why not call alloc_pidmap() under pidmap_lock
> > and find zero pid in pidmap w/o any loops and atomics?

Currently we search for zero bit lockless, why do you want to do it under spin_lock ?

Oleg.

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers