Subject: Re: [PATCH 1/7] introduce atomic_dec_and_lock_irqsave()
Posted by Oleg Nesterov on Wed, 30 Aug 2006 12:01:38 GMT
View Forum Message <> Reply to Message

On 08/30, Roman Zippel wrote:
> Hi,
>
> On Tue, 29 Aug 2006, Kirill Korotaev wrote:
>
> > --- ./kernel/user.c.dlirq 2006-07-10 12:39:20.000000000 +0400
> > +++ ./kernel/user.c 2006-08-28 11:08:56.000000000 +0400
> > @@ -108,15 +108,12 @@ void free_uid(struct user_struct *up)
> > if (!up)
> > return;
> >
> > - local_irq_save(flags);
> > - if (atomic_dec_and_lock(&up->__count, &uidhash_lock)) {
> > + if (atomic_dec_and_lock_irqsave(&up->__count, &uidhash_lock, flags)) {
> > uid_hash_remove(up);
> > spin_unlock_irqrestore(&uidhash_lock, flags);
> > key_put(up->uid_keyring);
> > key_put(up->session_keyring);
> > kmem_cache_free(uid_cachep, up);
> > - } else {
> > - local_irq_restore(flags);
> > }
> > }
>
> Why does this need protection against interrupts?

uidhash_lock can be taken from irq context. For example, delayed_put_task_struct()
does __put_task_struct()->free_uid().

Oleg.