Subject: Re: [PATCH] Wake up mandatory locks waiter on chmod
Posted by bfields on Mon, 17 Sep 2007 14:59:34 GMT
View Forum Message <> Reply to Message

On Mon, Sep 17, 2007 at 10:37:56AM +0400, Pavel Emelyanov wrote:
> J. Bruce Fields wrote:
> > Is there a small chance that a lock may be applied after this check:
> >
> >> + mandatory = (inode->i_flock && MANDATORY_LOCK(inode));
> >> +
> >
> > but early enough that someone can still block on the lock while the file
> > is still marked for mandatory locking? (And is the inode->i_flock check
> > there really necessary?)
>
> There is, but as you have noticed:

OK, but why not just remove the inode->i_flock check there? I can't see
how it helps anyway.

> > Well, there are probably worse races in the mandatory locking code.
>
> ...there are. The inode->i_lock is protected with lock_kernel() only
> and is not in sync with any other checks for inodes. This is sad :(
> but a good locking for locks is to be done...

I would also prefer a locking scheme that didn't rely on the BKL. That
said, except for this race:

> > (For example, my impression is that a mandatory lock can be applied just
> > after the locks_mandatory_area() checks but before the io actually
> > completes.)

... I'm not aware of other races in the existing file-locking code. It
sounds like you might be. Could you give specific examples?

--b.