Subject: Re: [PATCH] Wake up mandatory locks waiter on chmod
Posted by Pavel Emelianov on Mon, 17 Sep 2007 06:37:56 GMT

View Forum Message <> Reply to Message

J. Bruce Fields wrote:
> On Thu, Sep 13, 2007 at 06:30:43PM +0400, Pavel Emelyanov wrote:
>> When the process is blocked on mandatory lock and someone changes
>> the inode's permissions, so that the lock is no longer mandatory,
>> nobody wakes up the blocked process, but probably should.
>
> I suppose so. Does anyone actually use mandatory locking?

:) Good question.

> Would it be worth adding a
>
> if (MANDATORY_LOCK(inode))
>   return;
>
> to the beginning of locks_wakeup_mandatory() to avoid walking the list
> of locks in that case? Perhaps setattr is rare enough that this just
> isn't worth caring about.
>
> 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:

> 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...

> (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.)
>
> --b.

Thanks,
Pavel