Subject: Re: [PATCH] Memory shortage can result in inconsistent flocks state
Posted by Chuck Ebbert on Thu, 13 Sep 2007 19:45:01 GMT
View Forum Message <> Reply to Message

On 09/13/2007 03:34 PM, J. Bruce Fields wrote:
>> Doesn't that create a leak in some cases?
>>
>>>   for_each_lock(inode, before) {
>>>   struct file_lock *fl = *before;
>>>   if (IS_POSIX(fl))
>>>   break;
>>>   if (IS_LEASE(fl))
>>>   continue;
>>>   if (filp != fl->fl_file)
>>>   continue;
>>>   if (request->fl_type == fl->fl_type)
>>>   goto out; <<<<<<<<<<<<<<< LEAK?
>
> You mean, a leak of the memory allocated for new_fl? That's freed at
> the exit labeled with "out". It's the only exit:
>
> out:
>   unlock_kernel();
>   if (new_fl)
>   locks_free_lock(new_fl);
>   return error;
>
> And new_fl is initially NULL, assigned only once by the allocation, then
> assigned to NULL only at the very end when we know we've succeeded.
>
> Am I missing something else?
>

Never mind, I didn't look closely enough. Looks good to me.