Subject: Re: [PATCH] proc: use BUG_ON() in de_put()
Posted by Alexey Dobriyan on Thu, 22 Nov 2007 08:49:23 GMT
View Forum Message <> Reply to Message

On Fri, Nov 16, 2007 at 01:46:42PM -0800, Andrew Morton wrote:
> On Thu, 15 Nov 2007 19:12:49 +0300
> Alexey Dobriyan <adobriyan@sw.ru> wrote:
>
> > It's much more visible that some printk. I still has an unexplained oops
> > in proc, so let's leave it for a while.

```
> > --- a/fs/proc/inode.c
> > +++ b/fs/proc/inode.c
> > @@ -37,12 +37,7 @@ void de_put(struct proc_dir_entry *de)
> >  {
> >   if (de) {
> >    lock_kernel();
> > -  if (!atomic_read(&de->count)) {
> > -   printk("de_put: entry %s already free!\n", de->name);
> > -   unlock_kernel();
> > -   return;
> > -  }
> > -
> > +  BUG_ON(atomic_read(&de->count) == 0);
> >    if (atomic_dec_and_test(&de->count)) {
> >     if (de->deleted) {
> >      printk("de_put: deferred delete of %s\n",
```

>
> I don't see that an error in here _requires_ that we nuke the machine.
> Surely we can emit a warning and then recover in some fashion?

0 => -1 transition, google says this check triggered only once. I think
we should just drop it.