
Subject: Re: [PATCH] proc: use BUG_ON() in de_put()

Posted by [akpm](#) on Fri, 16 Nov 2007 21:46:42 GMT

[View Forum Message](#) <> [Reply to Message](#)

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.

>

> Signed-off-by: Alexey Dobriyan <adobriyan@sw.ru>

> ---

>

> fs/proc/inode.c | 7 +-----
> 1 file changed, 1 insertion(+), 6 deletions(-)

>

> --- 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?
