
Subject: [PATCH 3/4] proc: simplify remove_proc_entry() wrt locking

Posted by [Alexey Dobriyan](#) on Fri, 16 Nov 2007 15:10:15 GMT

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

We can take proc_subdir_lock for duration of list searching and removing from lists only. It can't hurt -- we can gather any amount of looked up PDEs right after proc_subdir_lock droppage in proc_lookup() anyway. Current code should already deal with this correctly.

Also this should make code more understandable:

- * original looks like a loop, however, it's a loop with unconditional trailing "break;" -- not loop at all.

- * more explicit statement that proc_subdir_lock protects only ->subdir lists.

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

```
fs/proc/generic.c | 67 ++++++-----
1 file changed, 32 insertions(+), 35 deletions(-)
```

--- a/fs/proc/generic.c

+++ b/fs/proc/generic.c

@@ -686,12 +686,12 @@ void free_proc_entry(struct proc_dir_entry *de)

void remove_proc_entry(const char *name, struct proc_dir_entry *parent)

```
{
    struct proc_dir_entry **p;
- struct proc_dir_entry *de;
+ struct proc_dir_entry *de = NULL;
    const char *fn = name;
    int len;
```

```
    if (!parent && xlate_proc_name(name, &parent, &fn) != 0)
```

```
- goto out;
```

```
+ return;
```

```
    len = strlen(fn);
```

```
    spin_lock(&proc_subdir_lock);
```

@@ -701,45 +701,42 @@ void remove_proc_entry(const char *name, struct proc_dir_entry *parent)

```
    de = *p;
```

```
    *p = de->next;
```

```
    de->next = NULL;
```

```
+ }
```

```
+ spin_unlock(&proc_subdir_lock);
```

```
+ if (!de)
```

```
+ return;
```

```
- spin_lock(&de->pde_unload_lock);
```

```

- /*
-  * Stop accepting new callers into module. If you're
-  * dynamically allocating ->proc_fops, save a pointer somewhere.
-  */
- de->proc_fops = NULL;
- /* Wait until all existing callers into module are done. */
- if (de->pde_users > 0) {
-   DECLARE_COMPLETION_ONSTACK(c);
+ spin_lock(&de->pde_unload_lock);
+ /*
+  * Stop accepting new callers into module. If you're
+  * dynamically allocating ->proc_fops, save a pointer somewhere.
+  */
+ de->proc_fops = NULL;
+ /* Wait until all existing callers into module are done. */
+ if (de->pde_users > 0) {
+   DECLARE_COMPLETION_ONSTACK(c);

-   if (!de->pde_unload_completion)
-     de->pde_unload_completion = &c;
+   if (!de->pde_unload_completion)
+     de->pde_unload_completion = &c;

-   spin_unlock(&de->pde_unload_lock);
-   spin_unlock(&proc_subdir_lock);
+   spin_unlock(&de->pde_unload_lock);

-   wait_for_completion(de->pde_unload_completion);
+   wait_for_completion(de->pde_unload_completion);

-   spin_lock(&proc_subdir_lock);
-   goto continue_removing;
- }
- spin_unlock(&de->pde_unload_lock);
+   goto continue_removing;
+ }
+ spin_unlock(&de->pde_unload_lock);

continue_removing:
- if (S_ISDIR(de->mode))
-   parent->nlink--;
- de->nlink = 0;
- WARN_ON(de->subdir);
- if (!atomic_read(&de->count))
-   free_proc_entry(de);
- else {
-   de->deleted = 1;
-   printk("remove_proc_entry: %s/%s busy, count=%d\n",

```

```

- parent->name, de->name, atomic_read(&de->count));
- }
- break;
+ if (S_ISDIR(de->mode))
+ parent->nlink--;
+ de->nlink = 0;
+ WARN_ON(de->subdir);
+ if (!atomic_read(&de->count))
+ free_proc_entry(de);
+ else {
+ de->deleted = 1;
+ printk("remove_proc_entry: %s/%s busy, count=%d\n",
+ parent->name, de->name, atomic_read(&de->count));
+ }
- spin_unlock(&proc_subdir_lock);
-out:
- return;
}

```

Subject: Re: [PATCH 3/4] proc: simplify remove_proc_entry() wrt locking

Posted by [akpm](#) on Wed, 21 Nov 2007 04:08:42 GMT

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

On Fri, 16 Nov 2007 18:10:15 +0300 Alexey Dobriyan <adobriyan@sw.ru> wrote:

```

> We can take proc_subdir_lock for duration of list searching and removing
> from lists only. It can't hurt -- we can gather any amount of looked up
> PDEs right after proc_subdir_lock droppage in proc_lookup() anyway.
> Current code should already deal with this correctly.
>
> Also this should make code more understandable:
> * original looks like a loop, however, it's a loop with unconditional
>   trailing "break;" -- not loop at all.
> * more explicit statement that proc_subdir_lock protects only ->subdir lists.

```

oopses the Vaio.

```

[ 12.595145] BUG: unable to handle kernel NULL pointer dereference at virtual address
00000030
[ 12.598487] printing eip: c01a607f *pde = 00000000
[ 12.601795] Oops: 0000 [#1] PREEMPT
[ 12.605101] last sysfs file:
[ 12.608432] Modules linked in:
[ 12.611727]
[ 12.615000] Pid: 1, comm: swapper Not tainted (2.6.24-rc3-mm1 #4)
[ 12.618345] EIP: 0060:[<c01a607f>] EFLAGS: 00010206 CPU: 0

```

```

[ 12.621713] EIP is at remove_proc_entry+0x69/0x16c
[ 12.625071] EAX: 00000000 EBX: f726d940 ECX: f726d9bd EDX: 00000000
[ 12.628445] ESI: 00000030 EDI: f726d940 EBP: f7841e3c ESP: f7841dcc
[ 12.631747] DS: 007b ES: 007b FS: 0000 GS: 0000 SS: 0068
[ 12.635052] Process swapper (pid: 1, ti=F7840000 task=F783ED30 task.ti=F7840000)
[ 12.635181] Stack: 00000005 f726d9c0 c042747c f7841df0 c0131992 00000282 f7841e1c
00000005
[ 12.638669]      00000000 00000046 00000174 00000000 c0138012 00000000 00000000
00000000
[ 12.642173]      f783f2e0 f783ed30 00000000 f783ed30 c0320d27 00000010 f7841e34
c013a1e4
[ 12.645623] Call Trace:
[ 12.652432] [<c0104e0c>] show_trace_log_lvl+0x12/0x25
[ 12.655938] [<c0104eab>] show_stack_log_lvl+0x8c/0x9e
[ 12.659333] [<c0104f47>] show_registers+0x8a/0x1c0
[ 12.662755] [<c010516b>] die+0xee/0x1c4
[ 12.666101] [<c0117a18>] do_page_fault+0x405/0x4e1
[ 12.669427] [<c0320fda>] error_code+0x6a/0x70
[ 12.672700] [<c0151d13>] unregister_handler_proc+0x1b/0x1d
[ 12.675974] [<c0150bb2>] free_irq+0xb3/0xdc
[ 12.679227] [<c028a51b>] yenta_probe_cb_irq+0xc9/0xd6
[ 12.682482] [<c028a829>] ti12xx_override+0x12b/0x4c5
[ 12.685782] [<c028b270>] yenta_probe+0x2b1/0x55d
[ 12.689042] [<c01fed0d>] pci_device_probe+0x39/0x5b
[ 12.692276] [<c025a9ad>] driver_probe_device+0xd1/0x147
[ 12.695492] [<c025ab37>] __driver_attach+0x6a/0xa1
[ 12.698666] [<c0259ee9>] bus_for_each_dev+0x37/0x5c
[ 12.701783] [<c025a816>] driver_attach+0x14/0x16
[ 12.704891] [<c025a220>] bus_add_driver+0x7a/0x191
[ 12.708015] [<c025ad08>] driver_register+0x57/0x5c
[ 12.711095] [<c01fee6f>] __pci_register_driver+0x56/0x83
[ 12.714170] [<c047203c>] yenta_socket_init+0x14/0x16
[ 12.717195] [<c0458650>] kernel_init+0xc5/0x20f
[ 12.720120] [<c0104aaf>] kernel_thread_helper+0x7/0x10
[ 12.723041] =====
[ 12.725918] INFO: lockdep is turned off.
[ 12.728813] Code: 75 94 83 c6 38 eb 24 8b 55 f0 89 d9 8b 45 90 e8 ab fe ff ff 85 c0 74 0e 8b 43
30 89 df 89 06 c7 43 30 00 00 00 00 8b 36 83 c6 30 <8b> 1e 85 db 75 d6 b8 e0 2b 43 c0 e8 ab
ab 17 00 85 ff 0f 84 e3
[ 12.735473] EIP: [<c01a607f>] remove_proc_entry+0x69/0x16c SS:ESP 0068:f7841dcc

```

(gdb) l *0xc01a4fdf

0xc01a4fdf is in remove_proc_entry (fs/proc/generic.c:698).

warning: Source file is more recent than executable.

```

693      if (!parent && xlate_proc_name(name, &parent, &fn) != 0)
694          return;
695      len = strlen(fn);

```

```

696
697     spin_lock(&proc_subdir_lock);
698     for (p = &parent->subdir; *p; p=&(*p)->next ) {
699         if (!proc_match(len, fn, *p))
700             continue;
701         de = *p;
702         *p = de->next;

```

iirc this is what Andy was hitting.

Subject: Re: [PATCH 3/4] proc: simplify remove_proc_entry() wrt locking

Posted by [Alexey Dobriyan](#) on Fri, 23 Nov 2007 09:18:29 GMT

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

On Tue, Nov 20, 2007 at 08:08:42PM -0800, Andrew Morton wrote:

> On Fri, 16 Nov 2007 18:10:15 +0300 Alexey Dobriyan <adobriyan@sw.ru> wrote:

>

> > We can take proc_subdir_lock for duration of list searching and removing

> > from lists only. It can't hurt -- we can gather any amount of looked up

> > PDEs right after proc_subdir_lock droppage in proc_lookup() anyway.

> > Current code should already deal with this correctly.

> >

> > Also this should make code more undestandable:

> > * original looks like a loop, however, it's a loop with unconditional

> > trailing "break;" -- not loop at all.

> > * more explicit statement that proc_subdir_lock protects only ->subdir lists.

>

> oopses the Vaio.

>

>

> [12.595145] BUG: unable to handle kernel NULL pointer dereference at virtual address 00000030

> [12.598487] printing eip: c01a607f *pde = 00000000

> [12.601795] Oops: 0000 [#1] PREEMPT

> [12.605101] last sysfs file:

> [12.608432] Modules linked in:

> [12.611727]

> [12.615000] Pid: 1, comm: swapper Not tainted (2.6.24-rc3-mm1 #4)

> [12.618345] EIP: 0060:[<c01a607f>] EFLAGS: 00010206 CPU: 0

> [12.621713] EIP is at remove_proc_entry+0x69/0x16c

> [12.625071] EAX: 00000000 EBX: f726d940 ECX: f726d9bd EDX: 00000000

> [12.628445] ESI: 00000030 EDI: f726d940 EBP: f7841e3c ESP: f7841dcc

> [12.631747] DS: 007b ES: 007b FS: 0000 GS: 0000 SS: 0068

> [12.635052] Process swapper (pid: 1, ti=F7840000 task=F783ED30 task.ti=F7840000)

> [12.635181] Stack: 00000005 f726d9c0 c042747c f7841df0 c0131992 00000282 f7841e1c 00000005

> [12.638669] 00000000 00000046 00000174 00000000 c0138012 00000000 00000000

00000000

```
> [ 12.642173] f783f2e0 f783ed30 00000000 f783ed30 c0320d27 00000010 f7841e34
c013a1e4
> [ 12.645623] Call Trace:
> [ 12.652432] [<c0104e0c>] show_trace_log_lvl+0x12/0x25
> [ 12.655938] [<c0104eab>] show_stack_log_lvl+0x8c/0x9e
> [ 12.659333] [<c0104f47>] show_registers+0x8a/0x1c0
> [ 12.662755] [<c010516b>] die+0xee/0x1c4
> [ 12.666101] [<c0117a18>] do_page_fault+0x405/0x4e1
> [ 12.669427] [<c0320fda>] error_code+0x6a/0x70
> [ 12.672700] [<c0151d13>] unregister_handler_proc+0x1b/0x1d
> [ 12.675974] [<c0150bb2>] free_irq+0xb3/0xdc
> [ 12.679227] [<c028a51b>] yenta_probe_cb_irq+0xc9/0xd6
> [ 12.682482] [<c028a829>] ti12xx_override+0x12b/0x4c5
> [ 12.685782] [<c028b270>] yenta_probe+0x2b1/0x55d
> [ 12.689042] [<c01fed0d>] pci_device_probe+0x39/0x5b
> [ 12.692276] [<c025a9ad>] driver_probe_device+0xd1/0x147
> [ 12.695492] [<c025ab37>] __driver_attach+0x6a/0xa1
> [ 12.698666] [<c0259ee9>] bus_for_each_dev+0x37/0x5c
> [ 12.701783] [<c025a816>] driver_attach+0x14/0x16
> [ 12.704891] [<c025a220>] bus_add_driver+0x7a/0x191
> [ 12.708015] [<c025ad08>] driver_register+0x57/0x5c
> [ 12.711095] [<c01fee6f>] __pci_register_driver+0x56/0x83
> [ 12.714170] [<c047203c>] yenta_socket_init+0x14/0x16
> [ 12.717195] [<c0458650>] kernel_init+0xc5/0x20f
> [ 12.720120] [<c0104aaf>] kernel_thread_helper+0x7/0x10
> [ 12.723041] =====
> [ 12.725918] INFO: lockdep is turned off.
> [ 12.728813] Code: 75 94 83 c6 38 eb 24 8b 55 f0 89 d9 8b 45 90 e8 ab fe ff ff 85 c0 74 0e 8b
43 30 89 df 89 06 c7 43 30 00 00 00 00 8b 36 83 c6 30 <8b> 1e 85 db 75 d6 b8 e0 2b 43 c0 e8
ab ab 17 00 85 ff 0f 84 e3
> [ 12.735473] EIP: [<c01a607f>] remove_proc_entry+0x69/0x16c SS:ESP 0068:f7841dcc
>
> (gdb) l *0xc01a4fdf
> 0xc01a4fdf is in remove_proc_entry (fs/proc/generic.c:698).
> warning: Source file is more recent than executable.
>
> 693         if (!parent && xlate_proc_name(name, &parent, &fn) != 0)
> 694             return;
> 695         len = strlen(fn);
> 696
> 697         spin_lock(&proc_subdir_lock);
> 698         for (p = &parent->subdir; *p; p=&(*p)->next) {
> 699             if (!proc_match(len, fn, *p))
> 700                 continue;
> 701             de = *p;
> 702             *p = de->next;
> 703             de->next = NULL;
```

```
break; <===
```

```
704 }
```

```
>
```

```
> iirc this is what Andy was hitting.
```

Doh! this was missing "break;". I'll resend the patch and refcounting fixes soon.

P.S.: Do you make notches on Vaio-of-death?
