Subject: [RFC][PATCH 13/16] Remove proc_mnt's use for killing inodes
Posted by Sukadev Bhattiprolu on Thu, 24 May 2007 01:14:17 GMT
View Forum Message <> Reply to Message

Subject: Remove proc_mnt's use for killing inodes

From: Dave Hansen <hansendc@us.ibm.com>

We use proc_mnt as a shortcut to find a superblock on which
to go killing /proc inodes. This will break if we ever have
more than one /proc superblock. So, use the superblock list
to go find each of the /proc sb's and kill inodes on each
superblock.

This does introduce an extra lock grab from what was there
before, but the list should be only 1 long 99% of the time,
and we don't exactly remove proc entries in hot paths. Note
that this *isn't* the path that we use to get rid of the
actual /proc pid entries. Those are a different beast.

Signed-off-by: Dave Hansen <haveblue@us.ibm.com>
---
fs/proc/generic.c | 28 ++++++++++++++++++++++-----
1 file changed, 23 insertions(+), 5 deletions(-)

Index: lx26-21-mm2/fs/proc/generic.c
===
--- lx26-21-mm2.orig/fs/proc/generic.c 2007-05-22 16:51:19.000000000 -0700
+++ lx26-21-mm2/fs/proc/generic.c 2007-05-22 16:59:55.000000000 -0700
@@ -554,13 +554,10 @@ static int proc_register(struct proc_dir
 return 0;
}

-/*
- * Kill an inode that got unregistered..
- */
-static void proc_kill_inodes(struct proc_dir_entry *de)
+static void proc_kill_inodes_sb(struct proc_dir_entry *de,
+ struct super_block *sb)
{
 struct list_head *p;
- struct super_block *sb = proc_mnt->mnt_sb;

 /*
  * Actually it's a partial revoke().
@@ -584,6 +581,27 @@ static void proc_kill_inodes(struct proc
 file_list_unlock();

```
 }

+/*
+ * Kill an inode that got unregistered..
+ */
+static void proc_kill_inodes(struct proc_dir_entry *de)
+{
+ struct list_head *l;
+ struct file_system_type *procfs;
+
+ procfs = get_fs_type("proc");
+ if (!procfs)
+  return;
+
+ spin_lock(&sb_lock);
+ list_for_each(l, &procfs->fs_supers) {
+  struct super_block *sb;
+  sb = list_entry(l, struct super_block, s_instances);
+  proc_kill_inodes_sb(de, sb);
+ }
+ spin_unlock(&sb_lock);
+}
+
 static struct proc_dir_entry *proc_create(struct proc_dir_entry **parent,
        const char *name,
        mode_t mode,
```