Subject: [PATCH -mm] proc: introduce and use pde_users_dec()
Posted by Alexey Dobriyan on Mon, 14 May 2007 11:33:12 GMT
View Forum Message <> Reply to Message

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

 Please, fold into fix-rmmod-read-write-races-in-proc-entries.patch
 This should address last review comment.

 fs/proc/inode.c | 135 ++++++++++++++++++++++++++++++++----------------------
 1 file changed, 54 insertions(+), 81 deletions(-)

--- a/fs/proc/inode.c
+++ b/fs/proc/inode.c
@@ -141,6 +141,15 @@ static const struct super_operations pro
  .remount_fs = proc_remount,
 };

+static void pde_users_dec(struct proc_dir_entry *pde)
+{
+ spin_lock(&pde->pde_unload_lock);
+ pde->pde_users--;
+ if (pde->pde_unload_completion && pde->pde_users == 0)
+ complete(pde->pde_unload_completion);
+ spin_unlock(&pde->pde_unload_lock);
+}
+
 static loff_t proc_reg_llseek(struct file *file, loff_t offset, int whence)
 {
  struct proc_dir_entry *pde = PDE(file->f_path.dentry->d_inode);
@@ -152,8 +161,10 @@ static loff_t proc_reg_llseek(struct fil
   * remove_proc_entry() is going to delete PDE (as part of module
   * cleanup sequence). No new callers into module allowed.
   */
- if (!pde->proc_fops)
- goto out_unlock;
+ if (!pde->proc_fops) {
+ spin_unlock(&pde->pde_unload_lock);
+ return rv;
+ }
  /*
   * Bump refcount so that remove_proc_entry will wail for ->llseek to
   * complete.
@@ -170,13 +181,7 @@ static loff_t proc_reg_llseek(struct fil
  llseek = default_llseek;
 rv = llseek(file, offset, whence);

```
- spin_lock(&pde->pde_unload_lock);
- pde->pde_users--;
- if (pde->pde_unload_completion && pde->pde_users == 0)
-  complete(pde->pde_unload_completion);
-out_unlock:
- spin_unlock(&pde->pde_unload_lock);
-
+ pde_users_dec(pde);
  return rv;
 }

@@ -187,8 +192,10 @@ static ssize_t proc_reg_read(struct file
  ssize_t (*read)(struct file *, char __user *, size_t, loff_t *);

  spin_lock(&pde->pde_unload_lock);
- if (!pde->proc_fops)
-  goto out_unlock;
+ if (!pde->proc_fops) {
+  spin_unlock(&pde->pde_unload_lock);
+  return rv;
+ }
  pde->pde_users++;
  read = pde->proc_fops->read;
  spin_unlock(&pde->pde_unload_lock);
@@ -196,13 +203,7 @@ static ssize_t proc_reg_read(struct file
  if (read)
   rv = read(file, buf, count, ppos);

- spin_lock(&pde->pde_unload_lock);
- pde->pde_users--;
- if (pde->pde_unload_completion && pde->pde_users == 0)
-  complete(pde->pde_unload_completion);
-out_unlock:
- spin_unlock(&pde->pde_unload_lock);
-
+ pde_users_dec(pde);
  return rv;
 }

@@ -213,8 +214,10 @@ static ssize_t proc_reg_write(struct fil
  ssize_t (*write)(struct file *, const char __user *, size_t, loff_t *);

  spin_lock(&pde->pde_unload_lock);
- if (!pde->proc_fops)
-  goto out_unlock;
+ if (!pde->proc_fops) {
+  spin_unlock(&pde->pde_unload_lock);
+  return rv;
```

```
+ }
  pde->pde_users++;
  write = pde->proc_fops->write;
  spin_unlock(&pde->pde_unload_lock);
@@ -222,13 +225,7 @@ static ssize_t proc_reg_write(struct fil
  if (write)
   rv = write(file, buf, count, ppos);

- spin_lock(&pde->pde_unload_lock);
- pde->pde_users--;
- if (pde->pde_unload_completion && pde->pde_users == 0)
-  complete(pde->pde_unload_completion);
-out_unlock:
- spin_unlock(&pde->pde_unload_lock);
-
+ pde_users_dec(pde);
  return rv;
 }

@@ -239,8 +236,10 @@ static unsigned int proc_reg_poll(struct
  unsigned int (*poll)(struct file *, struct poll_table_struct *);

  spin_lock(&pde->pde_unload_lock);
- if (!pde->proc_fops)
-  goto out_unlock;
+ if (!pde->proc_fops) {
+  spin_unlock(&pde->pde_unload_lock);
+  return rv;
+ }
  pde->pde_users++;
  poll = pde->proc_fops->poll;
  spin_unlock(&pde->pde_unload_lock);
@@ -248,13 +247,7 @@ static unsigned int proc_reg_poll(struct
  if (poll)
   rv = poll(file, pts);

- spin_lock(&pde->pde_unload_lock);
- pde->pde_users--;
- if (pde->pde_unload_completion && pde->pde_users == 0)
-  complete(pde->pde_unload_completion);
-out_unlock:
- spin_unlock(&pde->pde_unload_lock);
-
+ pde_users_dec(pde);
  return rv;
 }

@@ -266,8 +259,10 @@ static long proc_reg_unlocked_ioctl(stru
```

```
  int (*ioctl)(struct inode *, struct file *, unsigned int, unsigned long);

  spin_lock(&pde->pde_unload_lock);
- if (!pde->proc_fops)
-  goto out_unlock;
+ if (!pde->proc_fops) {
+  spin_unlock(&pde->pde_unload_lock);
+  return rv;
+ }
  pde->pde_users++;
  unlocked_ioctl = pde->proc_fops->unlocked_ioctl;
  ioctl = pde->proc_fops->ioctl;
@@ -283,13 +278,7 @@ static long proc_reg_unlocked_ioctl(stru
   unlock_kernel();
  }

- spin_lock(&pde->pde_unload_lock);
- pde->pde_users--;
- if (pde->pde_unload_completion && pde->pde_users == 0)
-  complete(pde->pde_unload_completion);
-out_unlock:
- spin_unlock(&pde->pde_unload_lock);
-
+ pde_users_dec(pde);
  return rv;
 }

@@ -301,8 +290,10 @@ static long proc_reg_compat_ioctl(struct
  long (*compat_ioctl)(struct file *, unsigned int, unsigned long);

  spin_lock(&pde->pde_unload_lock);
- if (!pde->proc_fops)
-  goto out_unlock;
+ if (!pde->proc_fops) {
+  spin_unlock(&pde->pde_unload_lock);
+  return rv;
+ }
  pde->pde_users++;
  compat_ioctl = pde->proc_fops->compat_ioctl;
  spin_unlock(&pde->pde_unload_lock);
@@ -310,13 +301,7 @@ static long proc_reg_compat_ioctl(struct
  if (compat_ioctl)
   rv = compat_ioctl(file, cmd, arg);

- spin_lock(&pde->pde_unload_lock);
- pde->pde_users--;
- if (pde->pde_unload_completion && pde->pde_users == 0)
-  complete(pde->pde_unload_completion);
```

```
-out_unlock:
- spin_unlock(&pde->pde_unload_lock);
-
+ pde_users_dec(pde);
  return rv;
 }
 #endif
@@ -328,8 +313,10 @@ static int proc_reg_mmap(struct file *fi
  int (*mmap)(struct file *, struct vm_area_struct *);

  spin_lock(&pde->pde_unload_lock);
- if (!pde->proc_fops)
-  goto out_unlock;
+ if (!pde->proc_fops) {
+  spin_unlock(&pde->pde_unload_lock);
+  return rv;
+ }
  pde->pde_users++;
  mmap = pde->proc_fops->mmap;
  spin_unlock(&pde->pde_unload_lock);
@@ -337,13 +324,7 @@ static int proc_reg_mmap(struct file *fi
  if (mmap)
   rv = mmap(file, vma);

- spin_lock(&pde->pde_unload_lock);
- pde->pde_users--;
- if (pde->pde_unload_completion && pde->pde_users == 0)
-  complete(pde->pde_unload_completion);
-out_unlock:
- spin_unlock(&pde->pde_unload_lock);
-
+ pde_users_dec(pde);
  return rv;
 }

@@ -354,8 +335,10 @@ static int proc_reg_open(struct inode *i
  int (*open)(struct inode *, struct file *);

  spin_lock(&pde->pde_unload_lock);
- if (!pde->proc_fops)
-  goto out_unlock;
+ if (!pde->proc_fops) {
+  spin_unlock(&pde->pde_unload_lock);
+  return rv;
+ }
  pde->pde_users++;
  open = pde->proc_fops->open;
  spin_unlock(&pde->pde_unload_lock);
```

```
@@ -363,13 +346,7 @@ static int proc_reg_open(struct inode *i
  if (open)
   rv = open(inode, file);

- spin_lock(&pde->pde_unload_lock);
- pde->pde_users--;
- if (pde->pde_unload_completion && pde->pde_users == 0)
-  complete(pde->pde_unload_completion);
-out_unlock:
- spin_unlock(&pde->pde_unload_lock);
-
+ pde_users_dec(pde);
  return rv;
 }

@@ -380,8 +357,10 @@ static int proc_reg_release(struct inode
  int (*release)(struct inode *, struct file *);

  spin_lock(&pde->pde_unload_lock);
- if (!pde->proc_fops)
-  goto out_unlock;
+ if (!pde->proc_fops) {
+  spin_unlock(&pde->pde_unload_lock);
+  return rv;
+ }
  pde->pde_users++;
  release = pde->proc_fops->release;
  spin_unlock(&pde->pde_unload_lock);
@@ -389,13 +368,7 @@ static int proc_reg_release(struct inode
  if (release)
   rv = release(inode, file);

- spin_lock(&pde->pde_unload_lock);
- pde->pde_users--;
- if (pde->pde_unload_completion && pde->pde_users == 0)
-  complete(pde->pde_unload_completion);
-out_unlock:
- spin_unlock(&pde->pde_unload_lock);
-
+ pde_users_dec(pde);
  return rv;
 }
```