Subject: [PATCH] Switch nfs/callback.c to using struct pid, not pid_t
Posted by Pavel Emelianov on Wed, 29 Aug 2007 13:36:24 GMT
View Forum Message <> Reply to Message

Pid namespaces make it dangerous to use pid and tgid values
when run in some namespace. The struct pid itself is going
to be the only way for working with task pids, so make the
nfs callback thread use it.

Since nfs_callback_info.pid is set to current's one and reset
on the thread exit, it is safe not to get the struct pid.

Since this pid is used later under lock_kernel() w/o sleeping
operations, checking for i to be not NULL and killing the
thread with kill_pid() is safe.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

```
diff --git a/fs/nfs/callback.c b/fs/nfs/callback.c
index a796be5..5b8e5fc 100644
--- a/fs/nfs/callback.c
+++ b/fs/nfs/callback.c
@@ -27,7 +27,7 @@
 struct nfs_callback_data {
  unsigned int users;
  struct svc_serv *serv;
- pid_t pid;
+ struct pid *pid;
  struct completion started;
  struct completion stopped;
 };
@@ -64,7 +64,7 @@ static void nfs_callback_svc(struct svc_
  __module_get(THIS_MODULE);
  lock_kernel();

- nfs_callback_info.pid = current->pid;
+ nfs_callback_info.pid = task_pid(current);
  daemonize("nfsv4-svc");
  /* Process request with signals blocked, but allow SIGKILL.  */
  allow_signal(SIGKILL);
@@ -98,7 +98,7 @@ static void nfs_callback_svc(struct svc_
  }

  svc_exit_thread(rqstp);
- nfs_callback_info.pid = 0;
+ nfs_callback_info.pid = NULL;
```

```
  complete(&nfs_callback_info.stopped);
  unlock_kernel();
  module_put_and_exit(0);
@@ -114,7 +114,7 @@ int nfs_callback_up(void)

  lock_kernel();
  mutex_lock(&nfs_callback_mutex);
- if (nfs_callback_info.users++ || nfs_callback_info.pid != 0)
+ if (nfs_callback_info.users++ || nfs_callback_info.pid != NULL)
    goto out;
  init_completion(&nfs_callback_info.started);
  init_completion(&nfs_callback_info.stopped);
@@ -157,9 +157,9 @@ void nfs_callback_down(void)
  mutex_lock(&nfs_callback_mutex);
  nfs_callback_info.users--;
  do {
-  if (nfs_callback_info.users != 0 || nfs_callback_info.pid == 0)
+  if (nfs_callback_info.users != 0 || nfs_callback_info.pid == NULL)
     break;
-  if (kill_proc(nfs_callback_info.pid, SIGKILL, 1) < 0)
+  if (kill_pid(nfs_callback_info.pid, SIGKILL, 1) < 0)
     break;
  } while (wait_for_completion_timeout(&nfs_callback_info.stopped, 5*HZ) == 0);
  mutex_unlock(&nfs_callback_mutex);
```

_____

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers