Subject: [PATCH] Fix ipc entries removal
Posted by Pavel Emelianov on Mon, 30 Oct 2006 12:07:21 GMT
View Forum Message <> Reply to Message

This patch fixes two issuses related to ipc_ids->entries freeing.

1. When freeing ipc namespace we need to free entries allocated
   with ipc_init_ids().

2. When removing old entries in grow_ary() ipc_rcu_putref()
   may be called on entries set to &ids->nullentry earlier in
   ipc_init_ids().
   This is almost impossible without namespaces, but with
   them this situation becomes possible.

Found during OpenVZ testing after obvious leaks in beancounters.

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

```
--- ./ipc/msg.c.ipcfx 2006-10-30 14:53:07.000000000 +0300
+++ ./ipc/msg.c 2006-10-30 15:00:08.000000000 +0300
@@ -124,6 +124,7 @@ void msg_exit_ns(struct ipc_namespace *n
  }
  mutex_unlock(&msg_ids(ns).mutex);

+ ipc_fini_ids(ns->ids[IPC_MSG_IDS]);
  kfree(ns->ids[IPC_MSG_IDS]);
  ns->ids[IPC_MSG_IDS] = NULL;
 }
--- ./ipc/sem.c.ipcfx 2006-10-30 14:53:07.000000000 +0300
+++ ./ipc/sem.c 2006-10-30 14:59:44.000000000 +0300
@@ -161,6 +161,7 @@ void sem_exit_ns(struct ipc_namespace *n
  }
  mutex_unlock(&sem_ids(ns).mutex);

+ ipc_fini_ids(ns->ids[IPC_SEM_IDS]);
  kfree(ns->ids[IPC_SEM_IDS]);
  ns->ids[IPC_SEM_IDS] = NULL;
 }
--- ./ipc/shm.c.ipcfx 2006-10-30 14:53:07.000000000 +0300
+++ ./ipc/shm.c 2006-10-30 14:59:57.000000000 +0300
@@ -116,6 +116,7 @@ void shm_exit_ns(struct ipc_namespace *n
  }
  mutex_unlock(&shm_ids(ns).mutex);

+ ipc_fini_ids(ns->ids[IPC_SHM_IDS]);
  kfree(ns->ids[IPC_SHM_IDS]);
  ns->ids[IPC_SHM_IDS] = NULL;
```

```
 }
--- ./ipc/util.c.ipcfx 2006-10-26 17:51:58.000000000 +0400
+++ ./ipc/util.c 2006-10-30 14:59:23.000000000 +0300
@@ -301,7 +301,7 @@ static int grow_ary(struct ipc_ids* ids,
   */
  rcu_assign_pointer(ids->entries, new);

- ipc_rcu_putref(old);
+ __ipc_fini_ids(ids, old);
  return newsize;
 }

--- ./ipc/util.h.ipcfx 2006-10-26 17:51:58.000000000 +0400
+++ ./ipc/util.h 2006-10-30 14:59:09.000000000 +0300
@@ -83,6 +83,18 @@ void* ipc_rcu_alloc(int size);
 void ipc_rcu_getref(void *ptr);
 void ipc_rcu_putref(void *ptr);

+static inline void __ipc_fini_ids(struct ipc_ids *ids,
+  struct ipc_id_ary *entries)
+{
+ if (entries != &ids->nullentry)
+  ipc_rcu_putref(entries);
+}
+
+static inline void ipc_fini_ids(struct ipc_ids *ids)
+{
+ __ipc_fini_ids(ids, ids->entries);
+}
+
 struct kern_ipc_perm* ipc_get(struct ipc_ids* ids, int id);
 struct kern_ipc_perm* ipc_lock(struct ipc_ids* ids, int id);
 void ipc_lock_by_ptr(struct kern_ipc_perm *ipcp);
```