
Subject: [PATCH 1/2] Don't forget to unlock uids_mutex on error paths

Posted by [Pavel Emelianov](#) on Wed, 21 Nov 2007 10:49:33 GMT

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

The commit

commit 5cb350baf580017da38199625b7365b1763d7180

Author: Dhaval Giani <dhaval@linux.vnet.ibm.com>

Date: Mon Oct 15 17:00:14 2007 +0200

sched: group scheduling, sysfs tunables

introduced the uids_mutex and the helpers to lock/unlock it.
Unfortunately, the error paths of alloc_uid() were not patched
to unlock it.

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

diff --git a/kernel/user.c b/kernel/user.c

index 56327ee..3549c4b 100644

--- a/kernel/user.c

+++ b/kernel/user.c

@@ -343,8 +343,11 @@ struct user_struct * alloc_uid(struct user_namespace *ns, uid_t uid)
 struct user_struct *new;

new = kmem_cache_alloc(uid_cachep, GFP_KERNEL);

- if (!new)

+ if (!new) {

+ uids_mutex_unlock();

return NULL;

+ }

+

new->uid = uid;

atomic_set(&new->__count, 1);

atomic_set(&new->processes, 0);

@@ -361,6 +364,7 @@ struct user_struct * alloc_uid(struct user_namespace *ns, uid_t uid)

if (alloc_uid_keyring(new, current) < 0) {

kmem_cache_free(uid_cachep, new);

+ uids_mutex_unlock();

return NULL;

}

@@ -368,6 +372,7 @@ struct user_struct * alloc_uid(struct user_namespace *ns, uid_t uid)

key_put(new->uid_keyring);

key_put(new->session_keyring);

```
    kmem_cache_free(uid_cachep, new);  
+   uids_mutex_unlock();  
    return NULL;  
}
```

--

1.5.3.4
