

---

Subject: Re: [patch -mm 10/17] nsproxy: add unshare\_ns and bind\_ns syscalls

Posted by [Daniel Lezcano](#) on Wed, 06 Dec 2006 21:01:32 GMT

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

---

Dmitry Mishin wrote:

> On Tuesday 05 December 2006 13:28, clg@fr.ibm.com wrote:  
>> From: Cedric Le Goater <clg@fr.ibm.com>  
> [skip]  
>> +static int switch\_ns(int id, unsigned long flags)  
>> +{  
>> + int err = 0;  
>> + struct nsproxy \*ns = NULL, \*old\_ns = NULL, \*new\_ns = NULL;  
>> +  
>> + if (flags & ~NS\_ALL)  
>> + return -EINVAL;  
>> +  
>> + /\* Let 0 be a default value ? \*/  
>> + if (!flags)  
>> + flags = NS\_ALL;  
>> +  
>> + if (id < 0) {  
>> + struct task\_struct \*p;  
>> +  
>> + err = -ESRCH;  
>> + read\_lock(&tasklist\_lock);  
>> + p = find\_task\_by\_pid(-id);  
>> + if (p) {  
>> + task\_lock(p);  
>> + get\_nsproxy(p->nsproxy);  
>> + ns = p->nsproxy;  
>> + task\_unlock(p);  
>> + }  
>> + read\_unlock(&tasklist\_lock);  
>> + } else {  
>> + err = -ENOENT;  
>> + spin\_lock\_irq(&ns\_hash\_lock);  
>> + ns = ns\_hash\_find(id);  
>> + spin\_unlock\_irq(&ns\_hash\_lock);  
>> + }  
>> +  
>> + if (!ns)  
>> + goto out;  
>> +  
>> + new\_ns = ns;  
>> +  
>> + /\*  
>> + \* clone current nsproxy and populate it with the namespaces  
>> + \* chosen by flags.  
>> + \*/

```

>> +
>> + */
>> + if (flags != NS_ALL) {
>> +     new_ns = dup_namespaces(current->nsproxy);
>> +     if (!new_ns) {
>> +         err = -ENOMEM;
>> +         goto out_ns;
>> +
>> +
>> +     if (flags & NS_MNT) {
>> +         put_mnt_ns(new_ns->mnt_ns);
>> +         get_mnt_ns(ns->mnt_ns);
>> +         new_ns->mnt_ns = ns->mnt_ns;
>> +
>> +
>> +     if (flags & NS_UTS) {
>> +         put_uts_ns(new_ns->uts_ns);
>> +         get_uts_ns(ns->uts_ns);
>> +         new_ns->uts_ns = ns->uts_ns;
>> +
>> +
>> +     if (flags & NS_IPC) {
>> +         put_ipc_ns(new_ns->ipc_ns);
>> +         new_ns->ipc_ns = get_ipc_ns(ns->ipc_ns);
>> +
> <<<< This code looks useless for me, as at this time new_ns->any_ptr ==
> ns->any_ptr.

```

Yep. Because of the kmemdup in clone\_namespace called by dup\_namespace.

---