
Subject: [patch -mm 16/17] net namespace: add unshare
Posted by [Cedric Le Goater](#) on Tue, 05 Dec 2006 10:28:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

From: Cedric Le Goater <clg@fr.ibm.com>

Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>

```
include/linux/net_namespace.h | 13 ++++++++
kernel/nsproxy.c              | 25 ++++++++
net/core/net_namespace.c      | 35 ++++++++
3 files changed, 70 insertions(+), 3 deletions(-)
```

Index: 2.6.19-rc6-mm2/include/linux/net_namespace.h

=====

```
--- 2.6.19-rc6-mm2.orig/include/linux/net_namespace.h
+++ 2.6.19-rc6-mm2/include/linux/net_namespace.h
@@ -3,6 +3,7 @@
```

```
#include <linux/kref.h>
#include <linux/nsproxy.h>
+#include <linux/errno.h>
```

```
struct net_namespace {
    struct kref kref;
@@ -19,6 +20,9 @@ static inline void get_net_ns(struct net
    kref_get(&ns->kref);
}
```

```
+extern int unshare_net_ns(unsigned long unshare_flags,
+    struct net_namespace **new_net);
+
extern int copy_net_ns(int flags, struct task_struct *tsk);
```

```
extern void free_net_ns(struct kref *kref);
@@ -36,6 +40,15 @@ static inline void get_net_ns(struct net
{
}
```

```
+static inline int unshare_net_ns(unsigned long unshare_flags,
+    struct net_namespace **new_net)
+{
+    if (unshare_flags & NS_NET)
+        return -EINVAL;
+
+    return 0;
+}
+
```

```
static inline int copy_net_ns(int flags, struct task_struct *tsk)
```

```
{  
    return 0;
```

```
Index: 2.6.19-rc6-mm2/kernel/nsproxy.c
```

```
-----  
--- 2.6.19-rc6-mm2.orig/kernel/nsproxy.c
```

```
+++ 2.6.19-rc6-mm2/kernel/nsproxy.c
```

```
@@ -329,6 +329,12 @@ static int switch_ns(int id, unsigned lo  
    put_pid_ns(new_ns->pid_ns);  
    new_ns->pid_ns = ns->pid_ns;  
}
```

```
+ if (flags & NS_NET) {  
+   get_net_ns(ns->net_ns);  
+   put_net_ns(new_ns->net_ns);  
+   new_ns->net_ns = ns->net_ns;  
+ }
```

```
+  
    out_ns:  
    put_nsproxy(ns);  
}
```

```
@@ -446,6 +452,7 @@ asmlinkage long sys_unshare_ns(unsigned  
    struct uts_namespace *uts, *new_uts = NULL;  
    struct ipc_namespace *ipc, *new_ipc = NULL;  
    struct pid_namespace *pid, *new_pid = NULL;  
+ struct net_namespace *net, *new_net = NULL;  
    unsigned long unshare_flags = 0;
```

```
/* Return -EINVAL for all unsupported flags */
```

```
@@ -475,17 +482,19 @@ asmlinkage long sys_unshare_ns(unsigned
```

```
    if ((err = unshare_pid_ns(unshare_ns_flags, &new_pid)))  
        goto bad_unshare_ns_cleanup_ipc;  
+ if ((err = unshare_net_ns(unshare_ns_flags, &new_net)))  
+   goto bad_unshare_ns_cleanup_pid;
```

```
- if (new_mnt || new_uts || new_ipc || new_pid) {  
+ if (new_mnt || new_uts || new_ipc || new_pid || new_net) {  
    old_nsproxy = current->nsproxy;  
    new_nsproxy = dup_namespaces(old_nsproxy);  
    if (!new_nsproxy) {  
        err = -ENOMEM;  
-   goto bad_unshare_ns_cleanup_pid;  
+   goto bad_unshare_ns_cleanup_net;  
    }  
}
```

```
- if (new_fs || new_mnt || new_uts || new_ipc || new_pid) {  
+ if (new_fs || new_mnt || new_uts || new_ipc || new_pid || new_net) {
```

```

task_lock(current);

@@ -524,12 +533,22 @@ asmlinkage long sys_unshare_ns(unsigned
    new_pid = pid;
}

+ if (new_net) {
+   net = current->nsproxy->net_ns;
+   current->nsproxy->net_ns = new_net;
+   new_net = net;
+ }
+
    task_unlock(current);
}

if (new_nsproxy)
    put_nsproxy(new_nsproxy);

+bad_unshare_ns_cleanup_net:
+ if (new_net)
+   put_net_ns(new_net);
+
bad_unshare_ns_cleanup_pid:
    if (new_pid)
        put_pid_ns(new_pid);
Index: 2.6.19-rc6-mm2/net/core/net_namespace.c
=====
--- 2.6.19-rc6-mm2.orig/net/core/net_namespace.c
+++ 2.6.19-rc6-mm2/net/core/net_namespace.c
@@ -18,6 +18,41 @@ struct net_namespace init_net_ns = {

#ifdef CONFIG_NET_NS

+/*
+ * Clone a new ns copying an original net ns, setting refcount to 1
+ * @old_ns: namespace to clone
+ * Return NULL on error (failure to kcalloc), new ns otherwise
+ */
+static struct net_namespace *clone_net_ns(struct net_namespace *old_ns)
+{
+   struct net_namespace *ns;
+
+   ns = kcalloc(sizeof(struct net_namespace), GFP_KERNEL);
+   if (!ns)
+       return NULL;
+
+   kref_init(&ns->kref);

```

```

+ return ns;
+}
+
+/*
+ * unshare the current process' net namespace.
+ */
+int unshare_net_ns(unsigned long unshare_flags,
+    struct net_namespace **new_net)
+{
+ if (unshare_flags & NS_NET) {
+ if (!capable(CAP_SYS_ADMIN))
+ return -EPERM;
+
+ *new_net = clone_net_ns(current->nsproxy->net_ns);
+ if (!*new_net)
+ return -ENOMEM;
+ }
+
+ return 0;
+}
+
+int copy_net_ns(int flags, struct task_struct *tsk)
+{
+    struct net_namespace *old_ns = tsk->nsproxy->net_ns;
+
+    --

```

Containers mailing list

Containers@lists.osdl.org

<https://lists.osdl.org/mailman/listinfo/containers>
