Subject: [patch 20/20] [Network namespace] For debug purpose, used to unshare the network namespace.
Posted by Daniel Lezcano on Sun, 10 Dec 2006 21:58:37 GMT
View Forum Message <> Reply to Message

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

---

 fs/debugfs/net_ns.c | 131 ++--
 1 files changed, 127 insertions(+), 4 deletions(-)

Index: 2.6.19-rc6-mm2/fs/debugfs/net_ns.c
===
--- 2.6.19-rc6-mm2.orig/fs/debugfs/net_ns.c
+++ 2.6.19-rc6-mm2/fs/debugfs/net_ns.c
@@ -15,8 +15,10 @@
 #include <linux/debugfs.h>
 #include <linux/sched.h>
 #include <linux/netdevice.h>
+#include <linux/inetdevice.h>
 #include <linux/syscalls.h>
 #include <linux/net_namespace.h>
+#include <linux/rtnetlink.h>

 static struct dentry *net_ns_dentry;
 static struct dentry *net_ns_dentry_dev;
@@ -52,8 +54,6 @@ static ssize_t net_ns_start_read_file(st
  return 0;
 }

-int net_ns_start(void);
-
 static ssize_t net_ns_start_write_file(struct file *file,
           const char __user *user_buf,
           size_t count, loff_t *ppos)
@@ -63,6 +63,8 @@ static ssize_t net_ns_start_write_file(s
  const char __user *p;
  char c;
  unsigned long flags;
+ struct net_namespace *net, *new_net;
+ struct nsproxy *new_nsproxy = NULL, *old_nsproxy = NULL;

  if (current_net_ns != &init_net_ns)
   return -EBUSY;
@@ -87,10 +89,37 @@ static ssize_t net_ns_start_write_file(s
   return -EINVAL;

```
  flags = (c=='2'?NS_NET2:NS_NET3);
- err = sys_unshare_ns(flags);
+
+ err = unshare_net_ns(flags, &new_net);
  if (err)
   return err;

+ old_nsproxy = current->nsproxy;
+ new_nsproxy = dup_namespaces(old_nsproxy);
+
+ if (!new_nsproxy) {
+  put_net_ns(new_net);
+  task_unlock(current);
+  return -ENOMEM;
+ }
+
+ task_lock(current);
+
+ if (new_nsproxy) {
+  current->nsproxy = new_nsproxy;
+  new_nsproxy = old_nsproxy;
+ }
+
+ new_net->ns = current->nsproxy;
+ net = current->nsproxy->net_ns;
+ current->nsproxy->net_ns = new_net;
+ new_net = net;
+
+ task_unlock(current);
+
+ put_nsproxy(new_nsproxy);
+ put_net_ns(new_net);
+
  return count;
 }

@@ -152,8 +181,102 @@ static ssize_t net_ns_info_write_file(st
         const char __user *user_buf,
         size_t count, loff_t *ppos)
 {
+ struct net_namespace *net_ns = current_net_ns;
+ struct net_device *dev;
+ struct in_device *in_dev;
+ struct in_ifaddr **ifap = NULL;
+ struct in_ifaddr *ifa = NULL;
+ char *colon;
+ int err;
+
```

```
+ char buff[1024];
+ char *eth, *addr, *s;
+ __be32 address = 0;
+ __be32 p;
+
+ if (!capable(CAP_NET_ADMIN))
+  return -EPERM;
+
+ if (net_ns->level != NET_NS_LEVEL3)
+  return -EPERM;
+
+ if (count > sizeof(buff))
+  return -EINVAL;
+
+ if (copy_from_user(buff, user_buf, count))
+  return -EFAULT;
+
+ buff[count] = '\0';
+
+       eth = buff;
+       s = strchr(eth, ' ');
+       if (!s)
+               return -EINVAL;
+       *s = 0;
+
+       addr = s + 1;
+       s = strchr(addr, '.');
+       if (!s)
+               return -EINVAL;
+       *s = 0;
+       p = simple_strtoul(addr, NULL, 0);
+       ((char *)&address)[3] = p;
+       addr = s + 1;
+
+       s = strchr(addr, '.');
+       if (!s)
+               return -EINVAL;
+       *s = 0;
+       p = simple_strtoul(addr, NULL, 0);
+       ((char *)&address)[2] = p;
+       addr = s + 1;
+
+       s = strchr(addr, '.');
+       if (!s)
+               return -EINVAL;
+       *s = 0;
+       p = simple_strtoul(addr, NULL, 0);
+       ((char *)&address)[1] = p;
```

```
+       addr = s + 1;
+
+       p = simple_strtoul(addr, NULL, 0);
+       ((char *)&address)[0]  = p;
+
+ colon = strchr(eth, ':');
+ if (colon)
+  *colon = 0;
+
+       address = htonl(address);
+
+ rtnl_lock();
+
+ err = -ENODEV;
+ dev = __dev_get_by_name(eth);
+ if (!dev)
+  goto out;
+
+ if (colon)
+  *colon = ':';
+
+ err = -EADDRNOTAVAIL;
+ in_dev = __in_dev_get_rtnl(dev);
+ if (!in_dev)
+  goto out;
+
+ for (ifap = &in_dev->ifa_list; (ifa = *ifap) != NULL;
+      ifap = &ifa->ifa_next)
+  if (!strcmp(eth, ifa->ifa_label) &&
+      address == ifa->ifa_local)
+   break;
+ if (!ifa)
+  goto out;
+
+ ifa->ifa_net_ns = net_ns;

- return -EPERM;
+ err = count;
+out:
+ rtnl_unlock();
+ return err;
 }



--
_____
Containers mailing list
```