
Subject: [patch 4/4] Network namespaces: playing and debugging
Posted by [Andrey Savochkin](#) on Mon, 26 Jun 2006 09:55:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Temporary code to play with network namespaces in the simplest way.

Do

```
exec 7< /proc/net/net_ns
```

in your bash shell and you'll get a brand new network namespace.

There you can, for example, do

```
ip link set lo up
```

```
ip addr list
```

```
ip addr add 1.2.3.4 dev lo
```

```
ping -n 1.2.3.4
```

Signed-off-by: Andrey Savochkin <saw@swsoft.com>

```
dev.c | 27 ++++++
```

```
1 files changed, 26 insertions, 1 deletion
```

--- ./net/core/dev.c.vensdbg Fri Jun 23 11:50:16 2006

+++ ./net/core/dev.c Fri Jun 23 11:50:40 2006

```
@@ -3444,6 +3444,8 @@ int net_ns_start(void)
```

```
if (err)
```

```
goto out_register;
```

```
put_net_ns(orig_ns);
```

```
+ printk(KERN_DEBUG "NET_NS: created new netcontext %p for %s (pid=%d)\n",
```

```
+ ns, task->comm, task->tgid);
```

```
return 0;
```

out_register:

```
@@ -3461,6 +3463,7 @@ EXPORT_SYMBOL(net_ns_start);
```

```
void net_ns_free(struct net_namespace *ns)
```

```
{
```

```
+ printk(KERN_DEBUG "NET_NS: netcontext %p freed\n", ns);
```

```
kfree(ns);
```

```
}
```

```
EXPORT_SYMBOL(net_ns_free);
```

```
@@ -3473,8 +3476,13 @@ static void net_ns_destroy(void *data)
```

```
ns = data;
```

```
push_net_ns(ns, orig_ns);
```

```
unregister_netdev(ns->loopback);
```

```
+ if (!list_empty(&ns->dev_base)) {
```

```
+ printk("NET_NS: BUG: context %p has devices! ref %d\n",
```

```
+ ns, atomic_read(&ns->active_ref));
```

```
+ pop_net_ns(orig_ns);
```

```
+ return;
```

```
+ }
```

```

    ip_fib_struct_fini();
- BUG_ON(!list_empty(&ns->dev_base));
    pop_net_ns(orig_ns);

    /* drop (hopefully) final reference */
@@ -3483,9 +3491,23 @@ static void net_ns_destroy(void *data)

void net_ns_stop(struct net_namespace *ns)
{
+ printk(KERN_DEBUG "NET_NS: netcontext %p scheduled for stop\n", ns);
    execute_in_process_context(net_ns_destroy, ns, &ns->destroy_work);
}
EXPORT_SYMBOL(net_ns_stop);
+
+static int net_ns_open(struct inode *i, struct file *f)
+{
+ return net_ns_start();
+}
+static struct file_operations net_ns_fops = {
+ .open = net_ns_open,
+};
+static int net_ns_init(void)
+{
+ return proc_net_fops_create("net_ns", S_IRWXU, &net_ns_fops)
+ ? 0 : -ENOMEM;
+}
#endif

/*
@@ -3550,6 +3572,9 @@ static int __init net_dev_init(void)
    hotcpu_notifier(dev_cpu_callback, 0);
    dst_init();
    dev_mcast_init();
+ #ifdef CONFIG_NET_NS
+ net_ns_init();
+ #endif
    rc = 0;
out:
    return rc;

```
