Subject: [PATCH net-2.6.25 3/3][IPV4] Use ctl paths to register devinet sysctls
Posted by Pavel Emelianov on Fri, 30 Nov 2007 18:32:04 GMT
View Forum Message <> Reply to Message

This looks very much like the patch for neighbors.

The path is also located on the stack and is prepared
inside the function. This time, the call to the registering
function is guarded with the RTNL lock, but I decided
to keep it on the stack not to litter the devinet.c file
with unneeded names and to make it look similar to the
neighbors code.

This is also intended to help us with the net namespaces
and saves the vmlinux size as well - this time by more
than 670 bytes.

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

```
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index f65c350..e3ac78a 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -1432,11 +1432,8 @@ int ipv4_doint_and_flush_strategy(ctl_table *table, int __user *name,
int nlen,

 static struct devinet_sysctl_table {
  struct ctl_table_header *sysctl_header;
- ctl_table  devinet_vars[__NET_IPV4_CONF_MAX];
- ctl_table  devinet_dev[2];
- ctl_table  devinet_conf_dir[2];
- ctl_table  devinet_proto_dir[2];
- ctl_table  devinet_root_dir[2];
+ struct ctl_table devinet_vars[__NET_IPV4_CONF_MAX];
+ char *dev_name;
 } devinet_sysctl = {
  .devinet_vars = {
   DEVINET_SYSCTL_COMPLEX_ENTRY(FORWARDING, "forwarding",
@@ -1468,38 +1465,6 @@ static struct devinet_sysctl_table {
   DEVINET_SYSCTL_FLUSHING_ENTRY(PROMOTE_SECONDARIES,
         "promote_secondaries"),
  },
- .devinet_dev = {
-  {
-   .ctl_name = NET_PROTO_CONF_ALL,
-   .procname = "all",
```

```
-   .mode  = 0555,
-   .child  = devinet_sysctl.devinet_vars,
-  },
- },
- .devinet_conf_dir = {
-  {
-   .ctl_name = NET_IPV4_CONF,
-   .procname = "conf",
-   .mode  = 0555,
-   .child  = devinet_sysctl.devinet_dev,
-  },
- },
- .devinet_proto_dir = {
-  {
-   .ctl_name = NET_IPV4,
-   .procname = "ipv4",
-   .mode  = 0555,
-   .child   = devinet_sysctl.devinet_conf_dir,
-  },
- },
- .devinet_root_dir = {
-  {
-   .ctl_name = CTL_NET,
-   .procname  = "net",
-   .mode  = 0555,
-   .child  = devinet_sysctl.devinet_proto_dir,
-  },
- },
 };

 static void __devinet_sysctl_register(char *dev_name, int ctl_name,
@@ -1508,6 +1473,16 @@ static void __devinet_sysctl_register(char *dev_name, int ctl_name,
  int i;
  struct devinet_sysctl_table *t;

+#define DEVINET_CTL_PATH_DEV 3
+
+ struct ctl_path devinet_ctl_path[] = {
+  { .procname = "net", .ctl_name = CTL_NET, },
+  { .procname = "ipv4", .ctl_name = NET_IPV4, },
+  { .procname = "conf", .ctl_name = NET_IPV4_CONF, },
+  { /* to be set */ },
+  { },
+ };
+
  t = kmemdup(&devinet_sysctl, sizeof(*t), GFP_KERNEL);
  if (!t)
   goto out;
```

```
@@ -1517,24 +1492,20 @@ static void __devinet_sysctl_register(char *dev_name, int ctl_name,
   t->devinet_vars[i].extra1 = p;
  }

- t->devinet_dev[0].ctl_name = ctl_name;
-
  /*
   * Make a copy of dev_name, because '.procname' is regarded as const
   * by sysctl and we wouldn't want anyone to change it under our feet
   * (see SIOCSIFNAME).
   */
- dev_name = kstrdup(dev_name, GFP_KERNEL);
- if (!dev_name)
+ t->dev_name = kstrdup(dev_name, GFP_KERNEL);
+ if (!t->dev_name)
   goto free;

- t->devinet_dev[0].procname    = dev_name;
- t->devinet_dev[0].child       = t->devinet_vars;
- t->devinet_conf_dir[0].child  = t->devinet_dev;
- t->devinet_proto_dir[0].child = t->devinet_conf_dir;
- t->devinet_root_dir[0].child  = t->devinet_proto_dir;
+ devinet_ctl_path[DEVINET_CTL_PATH_DEV].procname = t->dev_name;
+ devinet_ctl_path[DEVINET_CTL_PATH_DEV].ctl_name = ctl_name;

- t->sysctl_header = register_sysctl_table(t->devinet_root_dir);
+ t->sysctl_header = register_sysctl_paths(devinet_ctl_path,
+    t->devinet_vars);
  if (!t->sysctl_header)
   goto free_procname;

@@ -1542,7 +1513,7 @@ static void __devinet_sysctl_register(char *dev_name, int ctl_name,
  return;

 free_procname:
- kfree(dev_name);
+ kfree(t->dev_name);
 free:
  kfree(t);
 out:
@@ -1561,7 +1532,7 @@ static void devinet_sysctl_unregister(struct ipv4_devconf *p)
   struct devinet_sysctl_table *t = p->sysctl;
   p->sysctl = NULL;
   unregister_sysctl_table(t->sysctl_header);
-  kfree(t->devinet_dev[0].procname);
+  kfree(t->dev_name);
   kfree(t);
  }
```

}
--
1.5.3.4