
Subject: [net-2.6.24][NETNS][patch 3/3] fix bad macro definition
Posted by [Daniel Lezcano](#) on Wed, 12 Sep 2007 12:38:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

From: Daniel Lezcano <dlezcano@fr.ibm.com>

The macro definition is bad. When calling next_net_device with parameter name "dev", the resulting code is:

struct net_device *dev = dev and that leads to an unexpected behavior. Especially when llc_core is compiled in, the kernel panics at boot time.

The patchset change macro definition with static inline functions as they were defined before.

Signed-off-by: Benjamin Thery <benjamin.thery@bull.net>

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

```
include/linux/netdevice.h | 35 ++++++-----  
1 file changed, 17 insertions(+), 18 deletions(-)
```

Index: net-2.6.24/include/linux/netdevice.h

```
-----  
--- net-2.6.24.orig/include/linux/netdevice.h
```

```
+++ net-2.6.24/include/linux/netdevice.h
```

```
@@ -41,7 +41,8 @@
```

```
#include <linux/dmaengine.h>
```

```
#include <linux/workqueue.h>
```

```
-struct net;
```

```
+#include <net/net_namespace.h>
```

```
+
```

```
struct vlan_group;
```

```
struct ethtool_ops;
```

```
struct netpoll_info;
```

```
@@ -739,23 +740,21 @@
```

```
list_for_each_entry_continue(d, &(net)->dev_base_head, dev_list)
```

```
#define net_device_entry(lh) list_entry(lh, struct net_device, dev_list)
```

```
-#define next_net_device(d) \
```

```
-( { \
```

```
- struct net_device *dev = d; \
```

```
- struct list_head *lh; \
```

```
- struct net *net; \
```

```
- \
```

```
- net = dev->nd_net; \
```

```
- lh = dev->dev_list.next; \
```

```
- lh == &net->dev_base_head ? NULL : net_device_entry(lh); \
```

```
-})
```

```

-
-#define first_net_device(N) \
-({ \
- struct net *NET = (N); \
- list_empty(&NET->dev_base_head) ? NULL : \
- net_device_entry(NET->dev_base_head.next); \
-})
+static inline struct net_device *next_net_device(struct net_device *dev)
+{
+ struct list_head *lh;
+ struct net *net;
+
+ net = dev->nd_net;
+ lh = dev->dev_list.next;
+ return lh == &net->dev_base_head ? NULL : net_device_entry(lh);
+}
+
+static inline struct net_device *first_net_device(struct net *net)
+{
+ return list_empty(&net->dev_base_head) ? NULL :
+ net_device_entry(net->dev_base_head.next);
+}

```

```

extern int netdev_boot_setup_check(struct net_device *dev);
extern unsigned long netdev_boot_base(const char *prefix, int unit);

```

--

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
