Subject: [patch 02/10][NETNS][IP6_FIB] dynamically allocate fib tables
Posted by Daniel Lezcano on Thu, 15 Nov 2007 14:01:45 GMT
View Forum Message <> Reply to Message

The fib tables are dynamically allocated at the init and exit functions.
That provides the ability to do multiple instanciations of these tables.

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>
Signed-off-by: Benjamin Thery <benjamin.thery@bull.net>
---
 net/ipv6/ip6_fib.c | 43 ++---------
 1 file changed, 25 insertions(+), 18 deletions(-)

Index: linux-2.6-netns/net/ipv6/ip6_fib.c
===
--- linux-2.6-netns.orig/net/ipv6/ip6_fib.c
+++ linux-2.6-netns/net/ipv6/ip6_fib.c
@@ -166,22 +166,14 @@ static __inline__ void rt6_release(struc
   dst_free(&rt->u.dst);
 }

-static struct fib6_table __fib6_main_tbl = {
- .tb6_id = RT6_TABLE_MAIN,
- .tb6_root = {
- .leaf = &ip6_null_entry,
- .fn_flags = RTN_ROOT | RTN_TL_ROOT | RTN_RTINFO,
- },
-};
-
-static struct fib6_table *fib6_main_tbl = &__fib6_main_tbl;
+static struct fib6_table *fib6_main_tbl;

 #ifdef CONFIG_IPV6_MULTIPLE_TABLES
 #define FIB_TABLE_HASHSZ 256
 #else
 #define FIB_TABLE_HASHSZ 1
 #endif
-static struct hlist_head fib_table_hash[FIB_TABLE_HASHSZ];
+static struct hlist_head *fib_table_hash;

 static void fib6_link_table(struct fib6_table *tb)
 {
@@ -203,15 +195,8 @@ static void fib6_link_table(struct fib6_
 }

 #ifdef CONFIG_IPV6_MULTIPLE_TABLES
-static struct fib6_table __fib6_local_tbl = {
- .tb6_id = RT6_TABLE_LOCAL,

```
- .tb6_root  = {
-  .leaf  = &ip6_null_entry,
-  .fn_flags = RTN_ROOT | RTN_TL_ROOT | RTN_RTINFO,
- },
-};

-static struct fib6_table *fib6_local_tbl = &__fib6_local_tbl;
+static struct fib6_table *fib6_local_tbl;

 static struct fib6_table *fib6_alloc_table(u32 id)
 {
@@ -1484,6 +1469,28 @@ void __init fib6_init(void)
        0, SLAB_HWCACHE_ALIGN|SLAB_PANIC,
        NULL);

+ fib_table_hash = kzalloc(sizeof(*fib_table_hash)*FIB_TABLE_HASHSZ, GFP_KERNEL);
+ if (!fib_table_hash)
+  panic("IPV6: Failed to allocate fib_table_hash.\n");
+
+ fib6_main_tbl = kzalloc(sizeof(*fib6_main_tbl), GFP_KERNEL);
+ if (!fib6_main_tbl)
+  panic("IPV6: Failed to allocate fib6_main_tbl.\n");
+
+ fib6_main_tbl->tb6_id = RT6_TABLE_MAIN;
+ fib6_main_tbl->tb6_root.leaf = &ip6_null_entry;
+ fib6_main_tbl->tb6_root.fn_flags = RTN_ROOT | RTN_TL_ROOT | RTN_RTINFO;
+
+#ifdef CONFIG_IPV6_MULTIPLE_TABLES
+ fib6_local_tbl = kzalloc(sizeof(*fib6_local_tbl), GFP_KERNEL);
+ if (!fib6_local_tbl)
+  panic("IPV6: Failed to allocate fib6_local_tbl.\n");
+
+ fib6_local_tbl->tb6_id = RT6_TABLE_LOCAL;
+ fib6_local_tbl->tb6_root.leaf = &ip6_null_entry;
+ fib6_local_tbl->tb6_root.fn_flags = RTN_ROOT | RTN_TL_ROOT | RTN_RTINFO;
+#endif
+
  fib6_tables_init();

  __rtnl_register(PF_INET6, RTM_GETROUTE, NULL, inet6_dump_fib);

--
```


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