Subject: [patch -mm 04/17] nsproxy: externalizes exit_task_namespaces
Posted by Cedric Le Goater on Tue, 05 Dec 2006 10:27:56 GMT
View Forum Message <> Reply to Message

From: Cedric Le Goater <clg@fr.ibm.com>

this is required to remove a header dependency in sched.h which breaks
next patches.

Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>

---
 include/linux/nsproxy.h | 15 ++++----------
 kernel/nsproxy.c | 11 ++++++++++
 2 files changed, 15 insertions(+), 11 deletions(-)

Index: 2.6.19-rc6-mm2/include/linux/nsproxy.h
===
--- 2.6.19-rc6-mm2.orig/include/linux/nsproxy.h
+++ 2.6.19-rc6-mm2/include/linux/nsproxy.h
@@ -2,7 +2,8 @@
 #define _LINUX_NSPROXY_H

 #include <linux/spinlock.h>
-#include <linux/sched.h>
+
+struct task_struct;

 struct mnt_namespace;
 struct uts_namespace;
@@ -48,14 +49,6 @@ static inline void put_nsproxy(struct ns
  }
 }

-static inline void exit_task_namespaces(struct task_struct *p)
-{
- struct nsproxy *ns = p->nsproxy;
- if (ns) {
- task_lock(p);
- p->nsproxy = NULL;
- task_unlock(p);
- put_nsproxy(ns);
- }
-}
+extern void exit_task_namespaces(struct task_struct *p);
+
 #endif
Index: 2.6.19-rc6-mm2/kernel/nsproxy.c

```
===================================================================
--- 2.6.19-rc6-mm2.orig/kernel/nsproxy.c
+++ 2.6.19-rc6-mm2/kernel/nsproxy.c
@@ -37,6 +37,17 @@ void get_task_namespaces(struct task_str
  }
 }

+void exit_task_namespaces(struct task_struct *p)
+{
+ struct nsproxy *ns = p->nsproxy;
+ if (ns) {
+  task_lock(p);
+  p->nsproxy = NULL;
+  task_unlock(p);
+  put_nsproxy(ns);
+ }
+}
+
 /*
  * creates a copy of "orig" with refcount 1.
  * This does not grab references to the contained namespaces,

--
```