
Subject: [PATCH] Uninline the task_xid_nr_ns() calls
Posted by [Pavel Emelianov](#) on Tue, 25 Sep 2007 10:53:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

Since these are expanded into call to pid_nr_ns() anyway, it's OK to move the whole routine out-of-line. This is a cheap way to save ~100 bytes from vmlinux. Together with the previous two patches, it saves half-a-kilo from the vmlinux.

Un-inline other (currently inlined) functions must be done with additional performance testing.

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

```
diff --git a/include/linux/sched.h b/include/linux/sched.h
index fe238fb..4108cce 100644
--- a/include/linux/sched.h
+++ b/include/linux/sched.h
@@ -1228,11 +1228,7 @@ static inline pid_t task_pid_nr(struct t
    return tsk->pid;
}

-static inline pid_t task_pid_nr_ns(struct task_struct *tsk,
- struct pid_namespace *ns)
-{
- return pid_nr_ns(task_pid(tsk), ns);
-}
+pid_t task_pid_nr_ns(struct task_struct *tsk, struct pid_namespace *ns);

static inline pid_t task_pid_vnr(struct task_struct *tsk)
{
@@ -1245,11 +1241,7 @@ static inline pid_t task_tgid_nr(struct
    return tsk->tgid;
}

-static inline pid_t task_tgid_nr_ns(struct task_struct *tsk,
- struct pid_namespace *ns)
-{
- return pid_nr_ns(task_tgid(tsk), ns);
-}
+pid_t task_tgid_nr_ns(struct task_struct *tsk, struct pid_namespace *ns);

static inline pid_t task_tgid_vnr(struct task_struct *tsk)
{
@@ -1262,11 +1254,7 @@ static inline pid_t task_pgrp_nr(struct
    return tsk->signal->__pgrp;
```

```

}

-static inline pid_t task_pgrp_nr_ns(struct task_struct *tsk,
- struct pid_namespace *ns)
-{
- return pid_nr_ns(task_pgrp(tsk), ns);
-}
+pid_t task_pgrp_nr_ns(struct task_struct *tsk, struct pid_namespace *ns);

static inline pid_t task_pgrp_vnr(struct task_struct *tsk)
{
@@ -1279,11 +1267,7 @@ static inline pid_t task_session_nr(stru
return tsk->signal->__session;
}

-static inline pid_t task_session_nr_ns(struct task_struct *tsk,
- struct pid_namespace *ns)
-{
- return pid_nr_ns(task_session(tsk), ns);
-}
+pid_t task_session_nr_ns(struct task_struct *tsk, struct pid_namespace *ns);

static inline pid_t task_session_vnr(struct task_struct *tsk)
{
diff --git a/kernel/pid.c b/kernel/pid.c
index e2e060e..0032a88 100644
--- a/kernel/pid.c
+++ b/kernel/pid.c
@@ -412,6 +412,30 @@ pid_t pid_nr_ns(struct pid *pid, struct
return nr;
}

+pid_t task_pid_nr_ns(struct task_struct *tsk, struct pid_namespace *ns)
+{
+ return pid_nr_ns(task_pid(tsk), ns);
+}
+EXPORT_SYMBOL(task_pid_nr_ns);
+
+pid_t task_tgid_nr_ns(struct task_struct *tsk, struct pid_namespace *ns)
+{
+ return pid_nr_ns(task_tgid(tsk), ns);
+}
+EXPORT_SYMBOL(task_tgid_nr_ns);
+
+pid_t task_pgrp_nr_ns(struct task_struct *tsk, struct pid_namespace *ns)
+{
+ return pid_nr_ns(task_pgrp(tsk), ns);
+}

```

```
+EXPORT_SYMBOL(task_pgrp_nr_ns);
+
+pid_t task_session_nr_ns(struct task_struct *tsk, struct pid_namespace *ns)
+{
+ return pid_nr_ns(task_session(tsk), ns);
+}
+EXPORT_SYMBOL(task_session_nr_ns);
+
+/*
+ * Used by proc to find the first pid that is greater then or equal to nr.
+ */
```
