


```
+ return pnr->ns == &init_pid_ns || pnr->ns == ns;
+}
+#endif
#endif

#define do_each_pid_task(pid, type, task) \
--- ./include/linux/sched.h.multinrs 2007-06-15 15:23:00.000000000 +0400
+++ ./include/linux/sched.h 2007-06-15 15:32:45.000000000 +0400
@@ -1404,6 +1404,70 @@ static inline pid_t task_ppid_nr_ns(stru
return rcu_dereference(tsk->real_parent)->vtgid;
}
#endif
+
+#ifdef CONFIG_PID_NS_MULTILEVEL
+static inline pid_t task_pid_nr_ns(struct task_struct *tsk,
+ struct pid_namespace *ns)
+{
+ struct pid_number *pnr;
+
+ pnr = find_nr_by_pid(task_pid(tsk), ns);
+ return pnr != NULL ? pnr->nr : 0;
+}
+
+#define task_pid_vnr(t) task_pid_nr_ns(t, current->nsproxy->pid_ns)
+#define task_pid_nr(t) task_pid_nr_ns(t, &init_pid_ns)
+#define set_task_vpid(tsk, nr) do { } while (0)
+
+static inline pid_t task_tgid_nr_ns(struct task_struct *tsk,
+ struct pid_namespace *ns)
+{
+ struct pid_number *pnr;
+
+ pnr = find_nr_by_pid(task_tgid(tsk), ns);
+ return pnr != NULL ? pnr->nr : 0;
+}
+
+#define task_tgid_vnr(t) task_tgid_nr_ns(t, current->nsproxy->pid_ns)
+#define task_tgid_nr(t) task_tgid_nr_ns(t, &init_pid_ns)
+#define set_task_vtgid(tsk, nr) do { } while (0)
+
+static inline pid_t task_pgrp_nr_ns(struct task_struct *tsk,
+ struct pid_namespace *ns)
+{
+ struct pid_number *pnr;
+
+ pnr = find_nr_by_pid(task_pgrp(tsk), ns);
+ return pnr != NULL ? pnr->nr : 0;
+}
```

```

+
+ #define task_pgrp_vnr(t) task_pgrp_nr_ns(t, current->nsproxy->pid_ns)
+ #define task_pgrp_nr(t) task_pgrp_nr_ns(t, &init_pid_ns)
+ #define set_task_vpgrp(tsk, nr) do { } while (0)
+
+ static inline pid_t task_session_nr_ns(struct task_struct *tsk,
+ struct pid_namespace *ns)
+ {
+ struct pid_number *pnr;
+
+ pnr = find_nr_by_pid(task_session(tsk), ns);
+ return pnr != NULL ? pnr->nr : 0;
+ }
+
+ #define task_session_vnr(t) task_session_nr_ns(t, current->nsproxy->pid_ns)
+ #define task_session_nr(t) task_session_nr_ns(t, &init_pid_ns)
+ #define set_task_vsession(tsk, nr) do { } while (0)
+
+ static inline pid_t task_ppid_nr_ns(struct task_struct *tsk,
+ struct pid_namespace *ns)
+ {
+ struct pid_number *pnr;
+
+ pnr = find_nr_by_pid(task_tgid(rcu_dereference(tsk->real_parent)), ns);
+ return pnr != NULL ? pnr->nr : 0;
+ }
+
+ #endif
+ #endif
+
+ /**

```

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