Subject: [PATCH 23/28] [MULTI 2/6] Helpers to obtain pid numbers
Posted by Pavel Emelianov on Fri, 15 Jun 2007 16:25:17 GMT
View Forum Message <> Reply to Message

The is the implementation of [PREP 2/14] patch for the multilevel model

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

```
 pid.h   |   30 ++++++++++++++++++++++++++++++
 sched.h |   64
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 2 files changed, 94 insertions(+)

--- ./include/linux/pid.h.multinrs 2007-06-15 15:32:15.000000000 +0400
+++ ./include/linux/pid.h 2007-06-15 15:32:45.000000000 +0400
@@ -229,6 +229,36 @@ static inline int pid_ns_accessible(stru
  return pid->ns == &init_pid_ns || pid->ns == ns;
 }
 #endif
+
+#ifdef CONFIG_PID_NS_MULTILEVEL
+extern struct pid_number *find_nr_by_pid(struct pid *, struct pid_namespace *);
+
+static inline pid_t pid_nr_ns(struct pid *pid, struct pid_namespace *ns)
+{
+ struct pid_number *pnr;
+ pid_t nr = 0;
+ if (pid) {
+  pnr = find_nr_by_pid(pid, ns);
+  if (pnr != NULL)
+   nr = pnr->nr;
+ }
+ return nr;
+}
+#define pid_nr(pid) pid_nr_ns(pid, &init_pid_ns)
+#define pid_vnr(pid) pid_nr_ns(pid, current->nsproxy->pid_ns)
+
+static inline int pid_ns_accessible(struct pid_namespace *ns, struct pid *pid)
+{
+ struct pid_number *pnr;
+
+ /*
+  * the namespace that pid actually lives in is always at the first
+  * pid_number in the list (see alloc_pid_nrs)
+  */
+ pnr = pid->pid_nrs;
```

```
+ 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_nr(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