
Subject: [PATCH 3/3] Use the __pid_nr() calls where appropriate
Posted by [Pavel Emelianov](#) on Wed, 03 Oct 2007 14:23:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

Since we now have all the tasks have non-zero pids we may call the __pid_nr() instead of pid_nr() when the pid is get from task with task_pid() or similar call.

Besides, there are some other places where the check for pid to be not NULL is already done, so change them too.

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

```
diff --git a/fs/proc/base.c b/fs/proc/base.c
index 6ea12de..d94fda9 100644
--- a/fs/proc/base.c
+++ b/fs/proc/base.c
@@ -2394,7 +2394,7 @@ retry:
     task = NULL;
     pid = find_ge_pid(tgid, ns);
     if (pid) {
-    tgid = pid_nr_ns(pid, ns) + 1;
+    tgid = __pid_nr_ns(pid, ns) + 1;
     task = pid_task(pid, PIDTYPE_PID);
     /* What we to know is if the pid we have find is the
      * pid of a thread_group_leader. Testing for task
diff --git a/include/linux/sched.h b/include/linux/sched.h
index 4f21af1..e17b8f8 100644
--- a/include/linux/sched.h
+++ b/include/linux/sched.h
@@ -1229,7 +1229,7 @@ pid_t task_pid_nr_ns(struct task_struct

static inline pid_t task_pid_vnr(struct task_struct *tsk)
{
- return pid_vnr(task_pid(tsk));
+ return __pid_vnr(task_pid(tsk));
}

@@ -1242,7 +1242,7 @@ pid_t task_tgid_nr_ns(struct task_struct

static inline pid_t task_tgid_vnr(struct task_struct *tsk)
{
- return pid_vnr(task_tgid(tsk));
+ return __pid_vnr(task_tgid(tsk));
}
```

```
@@ -1255,7 +1255,7 @@ pid_t task_pgrp_nr_ns(struct task_struct
```

```
static inline pid_t task_pgrp_vnr(struct task_struct *tsk)
{
- return pid_vnr(task_pgrp(tsk));
+ return __pid_vnr(task_pgrp(tsk));
}
```

```
@@ -1268,14 +1268,14 @@ pid_t task_session_nr_ns(struct task_str
```

```
static inline pid_t task_session_vnr(struct task_struct *tsk)
{
- return pid_vnr(task_session(tsk));
+ return __pid_vnr(task_session(tsk));
}
```

```
static inline pid_t task_ppid_nr_ns(struct task_struct *tsk,
    struct pid_namespace *ns)
{
- return pid_nr_ns(task_pid(rcu_dereference(tsk->real_parent)), ns);
+ return __pid_nr_ns(task_pid(rcu_dereference(tsk->real_parent)), ns);
}
```

```
/**
```

```
diff --git a/kernel/fork.c b/kernel/fork.c
```

```
index f85731a..b640a10 100644
```

```
--- a/kernel/fork.c
```

```
+++ b/kernel/fork.c
```

```
@@ -1161,7 +1161,7 @@ static struct task_struct *copy_process(
}
}
```

```
- p->pid = pid_nr(pid);
+ p->pid = __pid_nr(pid);
p->tgid = p->pid;
if (clone_flags & CLONE_THREAD)
p->tgid = current->tgid;
```

```
diff --git a/kernel/pid.c b/kernel/pid.c
```

```
index d7388d7..b7a11cf 100644
```

```
--- a/kernel/pid.c
```

```
+++ b/kernel/pid.c
```

```
@@ -88,7 +99,7 @@ int is_cgroup_init(struct task_struct *t
```

```
rcu_read_lock();
```

```

    pid = task_pid(tsk);
- if (pid != NULL && pid->numbers[pid->level].nr == 1)
+ if (pid->numbers[pid->level].nr == 1)
    ret = 1;
    rcu_read_unlock();

```

@@ -443,25 +454,25 @@ pid_t pid_nr_ns(struct pid *pid, struct

```

pid_t task_pid_nr_ns(struct task_struct *tsk, struct pid_namespace *ns)
{
- return pid_nr_ns(task_pid(tsk), 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);
+ 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);
+ 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);
+ return __pid_nr_ns(task_session(tsk), ns);
}
EXPORT_SYMBOL(task_session_nr_ns);

```

diff --git a/kernel/sys.c b/kernel/sys.c

index 796299c..fa187d2 100644

--- a/kernel/sys.c

+++ b/kernel/sys.c

```

@@ -975,7 +975,7 @@ asmlinkage long sys_setpgid(pid_t pid, p
    detach_pid(p, PIDTYPE_PGID);
    pid = find_vpid(pgid);
    attach_pid(p, PIDTYPE_PGID, pid);
- set_task_pgrp(p, pid_nr(pid));
+ set_task_pgrp(p, __pid_nr(pid));
}

```

```
err = 0;
diff --git a/kernel/sysctl.c b/kernel/sysctl.c
index b11d22b..9bda3b5 100644
--- a/kernel/sysctl.c
+++ b/kernel/sysctl.c
@@ -2286,7 +2286,7 @@ static int proc_do_cad_pid(struct ctl_ta
    pid_t tmp;
    int r;

- tmp = pid_nr_ns(cad_pid, current->nsproxy->pid_ns);
+ tmp = __pid_nr_ns(cad_pid, current->nsproxy->pid_ns);

    r = __do_proc_dointvec(&tmp, table, write, filp, buffer,
        lenp, ppos, NULL, NULL);
```
