

---

Subject: Re: [PATCH 2/6] Rename pid\_nr function

Posted by [serue](#) on Fri, 13 Jul 2007 13:43:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Quoting Pavel Emelianov (xemul@openvz.org):

> sukadev@us.ibm.com wrote:

> > Pavel Emelianov [xemul@openvz.org] wrote:

> > | sukadev@us.ibm.com wrote:

> > | > Subject: [PATCH 2/6] Rename pid\_nr function

> > | >

> > | > From: Sukadev Bhattiprolu <sukadev@us.ibm.com>

> > | >

> > | > Rename pid\_nr() function to pid\_to\_nr() which is more descriptive

> > | > and will hopefully cause less confusion with new structure/functions

> > | > being added to support multiple pid namespaces.

> > |

> > | Don't we need at least two convertors:

> > | pid\_to\_nr\_as\_it\_seen\_by\_current()

> > | and

> > | pid\_to\_nr\_as\_it\_seen\_by\_some\_other\_task()

> >

> > Can you give me an example of where you would use this latter

> > (pid\_to\_nr\_as\_it\_seen\_by\_some\_other\_task()) interface ?

>

> Easy. Consider you're observing /proc/<pid>/status file for a task

> that lives in two namespaces - init and some created. When making

> "cat" on this from init namespace you must get the task's pid as it

> is seen from init namespace, but when making "cat" from the created

> namespace you must see the pid as it is seen by this namespace. And

> so on and so forth.

But 'current' in that case is the process reading the file, so you do in fact want to use pid\_to\_nr\_as\_it\_seen\_by\_current().

-serge

> On the other hand sys\_getpid() should simply return the virtual pid  
> without messing with namespaces.

>

> > I have a similar confusion with following two sets of interfaces

> > in your patchset and have been meaning to discuss it further.

> >

> > pid\_nr() and pid\_vnr()

> > find\_pid() and find\_vpid()

> >

> > I can understand the use of pid\_vnr() and find\_vpid().

> >

> > Where do you see us using pid\_nr() and find\_pid() - i.e the values

```

> > as seen in init_pid_ns ?
>
> find_pid() may be used in keventd_create_thread() as eventd thread
> lives in init namespace only. Any other place in kernel that is
> somehow "global" to all the namespaces and has no user interface
> must use global pids not to confuse the kernel.
>
> Of course, if we plan to replace the notion of global numerical pid
> with struct pid this is not an option. However, the issues with the
> pid_to_nr_as_it_... are still valid.
>
> > Like I pointed out in another thread, sunos_killpg() for instance
> > needs to use find_vpid(). Similarly, kill_proc_info(),
> > __set_special_pids() etc need find_vpid().
>
> Sure. These places are to be patched as well.
>
> > Thanks,
> >
> > Suka
> >
> >
> > |
> > | The first one is essentially the second one with "some other
> > | task" being current, but it can be optimized...
> > |
> > | > Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
> > | > ---
> > | > drivers/char/n_r3964.c | 7 ++++---
> > | > drivers/char/tty_io.c | 4 ++--
> > | > fs/autofs/root.c      | 2 +-
> > | > fs/fcntl.c           | 2 +-
> > | > fs/proc/array.c      | 2 +-
> > | > fs/smbfs/proc.c      | 2 +-
> > | > include/linux/pid.h  | 2 +-
> > | > ipc/mqueue.c         | 2 +-
> > | > kernel/container.c   | 2 +-
> > | > kernel/fork.c        | 2 +-
> > | > kernel/sysctl.c      | 2 +-
> > | > 11 files changed, 15 insertions(+), 14 deletions(-)
> > | >
> > | > Index: lx26-22-rc6-mm1/include/linux/pid.h
> > | > =====
> > | > --- lx26-22-rc6-mm1.orig/include/linux/pid.h 2007-07-05 18:53:43.000000000 -0700
> > | > +++ lx26-22-rc6-mm1/include/linux/pid.h 2007-07-05 18:54:29.000000000 -0700
> > | > @@ -98,7 +98,7 @@ extern struct pid *find_ge_pid(int nr);
> > | > extern struct pid *alloc_pid(void);
> > | > extern void FASTCALL(free_pid(struct pid *pid));

```

```

>> |>
>> |> -static inline pid_t pid_nr(struct pid *pid)
>> |> +static inline pid_t pid_to_nr(struct pid *pid)
>> |> {
>> |>   pid_t nr = 0;
>> |>   if (pid)
>> |> Index: lx26-22-rc6-mm1/drivers/char/n_r3964.c
>> |> =====
>> |> --- lx26-22-rc6-mm1.orig/drivers/char/n_r3964.c 2007-07-05 18:53:43.000000000 -0700
>> |> +++ lx26-22-rc6-mm1/drivers/char/n_r3964.c 2007-07-05 18:54:29.000000000 -0700
>> |> @@ -771,7 +771,7 @@ static int enable_signals(struct r3964_i
>> |>
>> |>   if (pClient->pid == pid) {
>> |>     TRACE_PS("removing client %d from client list",
>> |> -   pid_nr(pid));
>> |> +   pid_to_nr(pid));
>> |>     *ppClient = pClient->next;
>> |>     while (pClient->msg_count) {
>> |>       pMsg = remove_msg(pInfo, pClient);
>> |> @@ -801,7 +801,8 @@ static int enable_signals(struct r3964_i
>> |>   if (pClient == NULL)
>> |>     return -ENOMEM;
>> |>
>> |> - TRACE_PS("add client %d to client list", pid_nr(pid));
>> |> + TRACE_PS("add client %d to client list",
>> |> +   pid_to_nr(pid));
>> |>   spin_lock_init(&pClient->lock);
>> |>   pClient->sig_flags = arg;
>> |>   pClient->pid = get_pid(pid);
>> |> @@ -933,7 +934,7 @@ static void remove_client_block(struct r
>> |> {
>> |>   struct r3964_block_header *block;
>> |>
>> |> - TRACE_PS("remove_client_block PID %d", pid_nr(pClient->pid));
>> |> + TRACE_PS("remove_client_block PID %d", pid_to_nr(pClient->pid));
>> |>
>> |>   block = pClient->next_block_to_read;
>> |>   if (block) {
>> |> Index: lx26-22-rc6-mm1/drivers/char/tty_io.c
>> |> =====
>> |> --- lx26-22-rc6-mm1.orig/drivers/char/tty_io.c 2007-07-05 18:53:43.000000000 -0700
>> |> +++ lx26-22-rc6-mm1/drivers/char/tty_io.c 2007-07-05 18:54:29.000000000 -0700
>> |> @@ -3080,7 +3080,7 @@ static int tiocgpggrp(struct tty_struct *
>> |>   */
>> |>   if (tty == real_tty && current->signal->tty != real_tty)
>> |>     return -ENOTTY;
>> |> - return put_user(pid_nr(real_tty->pgrp), p);
>> |> + return put_user(pid_to_nr(real_tty->pgrp), p);

```

```

>> |> }
>> |>
>> |> /**
>> |> @@ -3151,7 +3151,7 @@ static int tiocgsid(struct tty_struct *t
>> |>     return -ENOTTY;
>> |>     if (!real_tty->session)
>> |>         return -ENOTTY;
>> |> - return put_user(pid_nr(real_tty->session), p);
>> |> + return put_user(pid_to_nr(real_tty->session), p);
>> |> }
>> |>
>> |> /**
>> |> Index: lx26-22-rc6-mm1/fs/autofs/root.c
>> |> =====
>> |> --- lx26-22-rc6-mm1.orig/fs/autofs/root.c 2007-07-05 18:53:43.000000000 -0700
>> |> +++ lx26-22-rc6-mm1/fs/autofs/root.c 2007-07-05 18:54:29.000000000 -0700
>> |> @@ -214,7 +214,7 @@ static struct dentry *autofs_root_lookup
>> |>
>> |>     oz_mode = autofs_oz_mode(sbi);
>> |>     DPRINTK(("autofs_lookup: pid = %u, pgrp = %u, catatonic = %d, "
>> |> -     "oz_mode = %d\n", pid_nr(task_pid(current)),
>> |> +     "oz_mode = %d\n", pid_to_nr(task_pid(current)),
>> |>     process_group(current), sbi->catatonic,
>> |>     oz_mode));
>> |>
>> |> Index: lx26-22-rc6-mm1/fs/fcntl.c
>> |> =====
>> |> --- lx26-22-rc6-mm1.orig/fs/fcntl.c 2007-07-05 18:53:43.000000000 -0700
>> |> +++ lx26-22-rc6-mm1/fs/fcntl.c 2007-07-05 18:54:29.000000000 -0700
>> |> @@ -305,7 +305,7 @@ pid_t f_getown(struct file *filp)
>> |> {
>> |>     pid_t pid;
>> |>     read_lock(&filp->f_owner.lock);
>> |> - pid = pid_nr(filp->f_owner.pid);
>> |> + pid = pid_to_nr(filp->f_owner.pid);
>> |>     if (filp->f_owner.pid_type == PIDTYPE_PGID)
>> |>         pid = -pid;
>> |>     read_unlock(&filp->f_owner.lock);
>> |> Index: lx26-22-rc6-mm1/fs/proc/array.c
>> |> =====
>> |> --- lx26-22-rc6-mm1.orig/fs/proc/array.c 2007-07-05 18:53:43.000000000 -0700
>> |> +++ lx26-22-rc6-mm1/fs/proc/array.c 2007-07-05 18:54:29.000000000 -0700
>> |> @@ -398,7 +398,7 @@ static int do_task_stat(struct task_stru
>> |>     struct signal_struct *sig = task->signal;
>> |>
>> |>     if (sig->tty) {
>> |> -     tty_pgrp = pid_nr(sig->tty->pgrp);
>> |> +     tty_pgrp = pid_to_nr(sig->tty->pgrp);

```

```

>> |>   tty_nr = new_encode_dev(tty_devnum(sig->tty));
>> |>   }
>> |>
>> |> Index: lx26-22-rc6-mm1/fs/smbfs/proc.c
>> |> =====
>> |> --- lx26-22-rc6-mm1.orig/fs/smbfs/proc.c 2007-07-05 18:53:43.000000000 -0700
>> |> +++ lx26-22-rc6-mm1/fs/smbfs/proc.c 2007-07-05 18:54:29.000000000 -0700
>> |> @@ -972,7 +972,7 @@ smb_newconn(struct smb_sb_info *server,
>> |>
>> |>   VERBOSE("protocol=%d, max_xmit=%d, pid=%d capabilities=0x%x\n",
>> |>   server->opt.protocol, server->opt.max_xmit,
>> |>   - pid_nr(server->conn_pid), server->opt.capabilities);
>> |> + pid_to_nr(server->conn_pid), server->opt.capabilities);
>> |>
>> |>   /* FIXME: this really should be done by smbmount. */
>> |>   if (server->opt.max_xmit > SMB_MAX_PACKET_SIZE) {
>> |> Index: lx26-22-rc6-mm1/ipc/mqueue.c
>> |> =====
>> |> --- lx26-22-rc6-mm1.orig/ipc/mqueue.c 2007-07-05 18:53:43.000000000 -0700
>> |> +++ lx26-22-rc6-mm1/ipc/mqueue.c 2007-07-05 18:54:29.000000000 -0700
>> |> @@ -336,7 +336,7 @@ static ssize_t mqueue_read_file(struct f
>> |>   (info->notify_owner &&
>> |>   info->notify.sigev_notify == SIGEV_SIGNAL) ?
>> |>   info->notify.sigev_signo : 0,
>> |>   - pid_nr(info->notify_owner));
>> |> + pid_to_nr(info->notify_owner));
>> |>   spin_unlock(&info->lock);
>> |>   buffer[sizeof(buffer)-1] = '\0';
>> |>   slen = strlen(buffer)+1;
>> |> Index: lx26-22-rc6-mm1/kernel/fork.c
>> |> =====
>> |> --- lx26-22-rc6-mm1.orig/kernel/fork.c 2007-07-05 18:53:43.000000000 -0700
>> |> +++ lx26-22-rc6-mm1/kernel/fork.c 2007-07-05 18:54:29.000000000 -0700
>> |> @@ -1032,7 +1032,7 @@ static struct task_struct *copy_process(
>> |>   p->did_exec = 0;
>> |>   delayacct_tsk_init(p); /* Must remain after dup_task_struct() */
>> |>   copy_flags(clone_flags, p);
>> |>   - p->pid = pid_nr(pid);
>> |> + p->pid = pid_to_nr(pid);
>> |>   INIT_LIST_HEAD(&p->children);
>> |>   INIT_LIST_HEAD(&p->sibling);
>> |>   p->vfork_done = NULL;
>> |> Index: lx26-22-rc6-mm1/kernel/sysctl.c
>> |> =====
>> |> --- lx26-22-rc6-mm1.orig/kernel/sysctl.c 2007-07-05 18:53:43.000000000 -0700
>> |> +++ lx26-22-rc6-mm1/kernel/sysctl.c 2007-07-05 18:54:29.000000000 -0700
>> |> @@ -2313,7 +2313,7 @@ static int proc_do_cad_pid(ctl_table *ta
>> |>   pid_t tmp;

```

```

>> |> int r;
>> |>
>> |> - tmp = pid_nr(cad_pid);
>> |> + tmp = pid_to_nr(cad_pid);
>> |>
>> |> r = __do_proc_dointvec(&tmp, table, write, filp, buffer,
>> |> lenp, ppos, NULL, NULL);
>> |> Index: lx26-22-rc6-mm1/kernel/container.c
>> |> =====
>> |> --- lx26-22-rc6-mm1.orig/kernel/container.c 2007-07-05 18:53:43.000000000 -0700
>> |> +++ lx26-22-rc6-mm1/kernel/container.c 2007-07-05 18:54:29.000000000 -0700
>> |> @@ -1600,7 +1600,7 @@ static int pid_array_load(pid_t *pidarra
>> |> while ((tsk = container_iter_next(cont, &it))) {
>> |>     if (unlikely(n == npids))
>> |>         break;
>> |> - pidarray[n++] = pid_nr(task_pid(tsk));
>> |> + pidarray[n++] = pid_to_nr(task_pid(tsk));
>> |> }
>> |> container_iter_end(cont, &it);
>> |> return n;
>> |>
>> |> _____
>> |> Containers mailing list
>> |> Containers@lists.linux-foundation.org
>> |> https://lists.linux-foundation.org/mailman/listinfo/containers
>> |>
>> |> _____
>> |> Devel mailing list
>> |> Devel@openvz.org
>> |> https://openvz.org/mailman/listinfo/devel
>> |>
>>
>
> _____
> Containers mailing list
> Containers@lists.linux-foundation.org
> https://lists.linux-foundation.org/mailman/listinfo/containers

```

---

```

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

```

---