Subject: Re: [RFC][PATCH 08/16] Define/use pid->upid_list list.
Posted by xemul on Thu, 24 May 2007 08:57:33 GMT
View Forum Message <> Reply to Message

```
> +/*
> + * Return the pid_t by which the process @pid is known in the pid
> + * namespace @ns.
> + *
> + * Return 0 if:
> + *  - @pid is NULL (eg: procfs calls this for task_pgrp(init_task)
> + *    which is NULL).
> + *
> + *  - process does not have pid_t in the namespace @ns (eg: parent
> + *    process of a child reaper does not exist in the child namespace.
> + *    A getppid() call by the child reaper results in 0).
> + */
> +pid_t pid_to_nr_in_ns(struct pid_namespace *ns, struct pid *pid)
> +{
> + int i;
> + struct upid *upid;
> +
> + if (!pid)
> +  return 0;
> +
> + upid = &pid->upid_list[0];
> + for (i = 0; i < pid->num_upids; i++, upid++) {
> +  if (upid->pid_ns == ns)
> +   return upid->nr;
> + }
```

This will make users of the kernel who do not need the
pid namespaces suffer from performance loss. Why not
introduce a CONFIG_PID_NS option?

```
> + return 0;
> +}
> +EXPORT_SYMBOL_GPL(pid_to_nr_in_ns);
> @@ -2182,6 +2185,8 @@ int proc_pid_readdir(struct file * filp,
>   struct task_struct *reaper = get_proc_task(filp->f_path.dentry->d_inode);
>   struct task_struct *task;
>   int tgid;
> + /* TODO get pid_ns from proc mnt rather than current */
> + struct pid_namespace *ns = task_active_pid_ns(current);
```

IMHO this is not a TODO, but MUSTDO.
When you have one /proc mount accessed from different pid
namespaces you may be in situation when one pidnr (like 123)
represents different tasks in these namespaces and thus the

inode staying behind the dentry with d_name "123" must
reference two tasks. When we scale the model to N namespace
inode must reference N tasks. But I don't see it in the
patches...

>
> if (!reaper)
> goto out_no_task;

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