

Kirill Korotaev wrote:

```
> +static struct ipc_namespace *clone_ipc_ns(struct ipc_namespace *old_ns)
> +{
> + int err;
> + struct ipc_namespace *ns;
> +
> + err = -ENOMEM;
> + ns = kmalloc(sizeof(struct ipc_namespace), GFP_KERNEL);
> + if (ns == NULL)
> + goto err_mem;
> +
> + err = sem_init_ns(ns);
> + if (err)
> + goto err_sem;
> + err = msg_init_ns(ns);
> + if (err)
> + goto err_msg;
> + err = shm_init_ns(ns);
> + if (err)
> + goto err_shm;
> +
> + kref_init(&ns->kref);
> + return ns;
> +
> +err_shm:
> + msg_exit_ns(ns);
> +err_msg:
> + sem_exit_ns(ns);
> +err_sem:
> + kfree(ns);
> +err_mem:
> + return ERR_PTR(err);
> +}
```

I've used the ipc namespace patchset in rc6-mm2. Thanks for putting this together, it works pretty well ! A few questions when we clone :

* We should do something close to what exit_sem() already does to clear the sem_undo list from the task doing the clone() or unshare().

* I don't like the idea of being able to unshare the ipc namespace and keep some shared memory from the previous ipc namespace mapped in the process mm. Should we forbid the unshare ?

Small fix follows,

thanks,

C.
