Subject: Re: [PATCH 2.6.24-rc8-mm1 09/15] (RFC) IPC: new kernel API to change an ID
Posted by Alexey Dobriyan on Tue, 29 Jan 2008 21:06:56 GMT
View Forum Message <> Reply to Message

On Tue, Jan 29, 2008 at 05:02:38PM +0100, pierre.peiffer@bull.net wrote:
> This patch provides three new API to change the ID of an existing
> System V IPCs.
>
> These APIs are:
> long msg_chid(struct ipc_namespace *ns, int id, int newid);
> long sem_chid(struct ipc_namespace *ns, int id, int newid);
> long shm_chid(struct ipc_namespace *ns, int id, int newid);
>
> They return 0 or an error code in case of failure.
>
> They may be useful for setting a specific ID for an IPC when preparing
> a restart operation.
>
> To be successful, the following rules must be respected:
> - the IPC exists (of course...)
> - the new ID must satisfy the ID computation rule.
> - the entry in the idr corresponding to the new ID must be free.

> ipc/util.c | 48 ++
> ipc/util.h | 1 +
> 8 files changed, 197 insertions(+)

For the record, OpenVZ uses "create with predefined ID" method which
leads to less code. For example, change at the end is all we want from
ipc/util.c .

Also, if ids were A and B at the moment of checkpoint, and during
restart they became B and A you'll get collision in both ways which you
techically can avoid by classic "tmp = A, A = B, B = tmp" but you also
can avoid all other loops just by creating with ID you need.

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