

---

Subject: Re: [PATCH 2.6.24-rc8-mm1 09/15] (RFC) IPC: new kernel API to change an ID

Posted by [Pierre Peiffer](#) on Wed, 30 Jan 2008 09:52:14 GMT

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

---

Alexey Dobriyan wrote:

> 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 .

>

Yes, indeed, I saw that. The idea here is, at the end, to propose a more  
"userspace oriented" solution.

As we can't use msgget(), etc, API to specify an ID, I think we can at least  
change it afterwards

> 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"

In the general case, yes, you're right.

In the case of the checkpoint/restart, this is not necessarily a problem, as we  
will probably restart an application in an empty "container"/"namespace"; Thus  
we can create all needed IPCs in an empty IPC namespace like this:

1. create first IPC

2. change its ID
3. create the second IPC
4. change its ID
5. etc..

But yes, I agree that if we can directly create an IPC with the right ID, it would be better; may be with an IPC\_CREATE command or something like that if the direction is to do that from userspace.

--

Pierre Peiffer

---

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>

---