

---

Subject: Re: [PATCH 0/4] - v2 - Object creation with a specified id  
Posted by [Nadia Derby](#) on Mon, 21 Apr 2008 11:32:41 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dave Hansen wrote:

> On Fri, 2008-04-18 at 07:44 +0200, Nadia.Derbey@bull.net wrote:

>

>>The syntax is one of:

>> . echo "LONG XX" > /proc/self/task/<my\_tid>/next\_id

>> next object to be created will have an id set to XX

>> . echo "LONG<n> X0 ... X<n-1>" > /proc/self/task/<my\_tid>/next\_id

>> next object to be created will have its ids set to XX0, ... X<n-1>

>> This is particularly useful for processes that may have several ids if

>> they belong to nested namespaces.

>

>

> I still think, in its current form, this is pretty dangerous.

>

> It might be nice to have a central place to set ids, but there's no real

> safety here for when we get a mishmash of 50 of these things. We have a

> queue and we fix the order in which things are inserted, but how do we

> control the order in which the kernel extracts them?

I thought that we were the ones who are going to be controlling the restart phase?

I think sysV IPCs are particular, since they could be recreated early during the restart phase:

we could define a callback routine that would be called by each process being restarted and that would do:

for each ipc type:

1. extract next ipc id for my uid and current ipc type
2. write(fd\_next\_id, id, id\_sz)
3. call the XXXget() routine for current ipc\_type
4. remove that id from the checkpoint file

I think this could only be applied to msg queues and sems, since the creator pid is stored in the perm structure in the shm case. But in the shm case, we can replace in 1. "my uid" by "my tgid\_vnr"

Then the processes hierarchy restore would take place, so no confusion between the ids.

> Can we 100%

> conform to that order in the future and be consistent? Won't that also

> be dictated by the way userspace behaves?

>

> If we're going to go this route, I think we need to do a few things.  
> First, these need to be more "type safe". That means that sticking an  
> ids in here that you mean to go to an IPC should never, ever, ever be  
> able to get interpreted as a PID or as an inode number or anything else.  
>  
> echo "IPC\_SEM\_ID NN" > /proc/self/task/<my\_tid>/next\_id  
> echo "PID MM" > /proc/self/task/<my\_tid>/next\_id  
>  
> Maybe we should include the types in there as well, but those are kinda  
> implied by the kind of id you're setting. Have you thought about 32-bit  
> userspace with 64-bit kernels?

No, but the ids are int in the ipc case and in the pid\_t case too.

> Does that cause any issues?  
>  
> I also feel like we should have the kernel able to enumerate which  
> things in a process \*are\* settable. What are the valid values for that  
> first word being echoed in the file?  
>  
> Does anyone have a list of these ids that we're going to want to set  
> like this? Oren? I'm a bit worried that we might want to set things  
> other than ids with an interface like this and that it won't transition  
> well.  
>

Regards,  
Nadia

---

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

---