
Subject: Re: [RFC][PATCH 0/4] Object creation with a pre-defined id (v2)

Posted by [serue](#) on Wed, 02 Apr 2008 15:56:53 GMT

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

Quoting Oren Laadan (orenl@cs.columbia.edu):

>
>
> On Fri, 28 Mar 2008, Nadia.Derbey@bull.net wrote:
>
> >
> > Hi,
> >
> > Here is a second version of what has been proposed 2 weeks ago to create
> > an object with a pre-defined id (this feature would be used during the
> > restart operation) - see thread
> > <https://lists.linux-foundation.org/pipermail/containers/2008-March/thread.html#10287>
> >
> > Main changes since last version:
> > . Pavel's suggestion has been integrated; this makes things more readable:
> > alloc_pidmap() is unchanged and a alloc_fixed_pidmap() is added for the
> > predefined ids.
> > . Oren's suggestion has been integrated:
> > We now have a single file under /proc/self (/proc/self/next_id).
> > When this file is filled, a structure pointed to by the calling task struct
> > is filled with the id(s).
> > Then, when the object is created, the id(s) present in that structure are
> > used, instead of the default ones.
> > The syntax is one of:
> > . echo "LONG XX" > /proc/self/next_id
> > next object to be created will have an id set to XX
> > . echo "LONG<n> X0 ... X<n-1>" > /proc/self/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 suggest that there be a way for the process to reset its task->next_id
> discarding previous settings, e.g. to recover from an error condition and
> prevent subsequent syscalls from using the task->next_id unintentionally.
> Something like "echo RESET > /proc/self/next_id" (or s/RESET/0/ ... etc).

Makes sense, and should be trivial enough.

If there are no objections to this approach on this list, should you be sending this patchset (addressing Oren's comments) on to lkml? Or is it too soon for that?

thanks,
-serge

> > The objects covered here are ipc objects and processes.
> >
> > The patches are still against 2.6.25-rc3-mm1, in the following order:
> >
> > [PATCH 1/4] adds the procfs facility for next object to be created, this
> > object being associated to a single id.
> > [PATCH 2/4] enhances the procfs facility for objects associated to multiple
> > ids (like processes).
> > [PATCH 3/4] makes use of the specified id (if any) to allocate the new IPC
> > object (changes the ipc_addid() path).
> > [PATCH 4/4] uses the specified id(s) (if any) to set the upid nr(s) for a newly
> > allocated process (changes the alloc_pid() path).
> >
> > Any comment and/or suggestions are welcome.
> >
> > Regards,
> > Nadia
> >
> > --
> >
> _____
> Containers mailing list
> Containers@lists.linux-foundation.org
> <https://lists.linux-foundation.org/mailman/listinfo/containers>

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