
Subject: Re: The issues for agreeing on a virtualization/namespaces implementation.

Posted by [serue](#) on Wed, 08 Feb 2006 03:36:33 GMT

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

Quoting Eric W. Biederman (ebiederm@xmission.com):

> Alexey Kuznetsov <kuznet@ms2.inr.ac.ru> writes:

>

> > Hello!

> >

> >> >2) What is the syscall interface to create these namespaces?

> >> > - Do we add clone flags?

> >> > (Plan 9 style)

> >>

> >> Like that approach .. flexible .. particular when one has well specified

> >> namespaces.

> >>

> >> > - Do we add a syscall (similar to setsid) per namespace?

> >> > (Traditional unix style)?

> >>

> >> Where does that approach end .. what's wrong with doing it at clone() time ?

> >

> > That most of those namespaces need a special setup rather than a plain copy?

> >

> > F.e. what are you going to do with NETWORK namespace? The only valid thing

> > to do is to prepare a new context and to configure its content (addresses,

> > routing tables, iptables...) later. So that, in this case it is natural

> > to inherit the context through clone() and to create new context

> > with a separate syscall.

>

> With a NETWORK namespace what I implemented was that you get a empty

> namespace with a loopback interface.

>

> But setting up the namespace from the inside is clearly the sane thing

> todo.

What I tried to do in a proof of concept long ago was to have CLONE_NETNS mean that you get access to all the network devices, but then you could drop/add them. Conceptually I prefer that to getting an empty namespace, but I'm not sure whether there's any practical use where you'd want that...

-serge
