
Subject: Re: [PATCH 1/2] rcfs core patch

Posted by [Srivatsa Vaddagiri](#) on Fri, 09 Mar 2007 17:57:07 GMT

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

On Fri, Mar 09, 2007 at 01:38:19AM +0100, Herbert Poetzl wrote:

> > 2) you allow a task to selectively reshare namespaces/subsystems with
> > another task, i.e. you can update current->task_proxy to point to
> > a proxy that matches your existing task_proxy in some ways and the
> > task_proxy of your destination in others. In that case a trivial
> > implementation would be to allocate a new task_proxy and copy some
> > pointers from the old task_proxy and some from the new. But then
> > whenever a task moves between different groupings it acquires a
> > new unique task_proxy. So moving a bunch of tasks between two
> > groupings, they'd all end up with unique task_proxy objects with
> > identical contents.

>

> this is exactly what Linux-VServer does right now, and I'm
> still not convinced that the nsproxy really buys us anything
> compared to a number of different pointers to various spaces
> (located in the task struct)

Are you saying that the current scheme of storing pointers to different spaces (uts_ns, ipc_ns etc) in nsproxy doesn't buy anything?

Or are you referring to storage of pointers to resource (name)spaces in nsproxy doesn't buy anything?

In either case, doesn't it buy speed and storage space?

> I'd prefer to do accounting (and limits) in a very simple
> and especially performant way, and the reason for doing
> so is quite simple:

Can you elaborate on the relationship between data structures used to store those limits to the task_struct? Does task_struct store pointers to those objects directly?

--

Regards,
vatsa

Containers mailing list

Containers@lists.osdl.org

<https://lists.osdl.org/mailman/listinfo/containers>
