

---

Subject: Re: [RFC][PATCH 1/5] Virtualization/containers: startup

Posted by [dev](#) on Mon, 06 Feb 2006 17:19:54 GMT

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

---

>>Please, also note, in OpenVZ we have 2 pointers on task\_struct:  
>>One is owner of a task (owner\_env), 2nd is a current context (exec\_env).  
>>exec\_env pointer is used to avoid adding of additional argument to all the  
>>functions where current context is required.

>

>

> That naming \_has\_ to change.

I agree.

> "exec" has a very clear meaning in unix: it talks about the notion of  
> switching to another process image, or perhaps the bit that says that a  
> file contains an image that can be executed. It has nothing to do with  
> "current".  
> What you seem to be talking about is the \_effective\_ environment. Ie the  
> same way we have "uid" and "euid", you'd have a "container" and the  
> "effective container".  
agree on this either. Good point.

> The "owner" name also makes no sense. The security context doesn't "own"  
> tasks. A task is \_part\_ of a context.

> So if some people don't like "container", how about just calling it  
> "context"? The downside of that name is that it's very commonly used in  
> the kernel, because a lot of things have "contexts". That's why "container"  
> would be a lot better.

>

> I'd suggest

>

> current->container - the current EFFECTIVE container

> current->master\_container - the "long term" container.

>

> (replace "master" with some other non-S&M term if you want)

maybe task\_container? i.e. where task actually is.

Sounds good for you?

The only problem with such names I see, that task will be an exception then compared to other objects. I mean, on other objects field "container" will mean the container which object is part of. But for tasks this will mean effective one. Only tasks need these 2 containers pointers and I would prefer having the common one to be called simply "container".

Maybe then

current->econtainer - effective container

current->container - "long term" container

> (It would make sense to just have the prepend-"e" semantics of uid/gid,  
> but the fact is, "euid/egid" has a long unix history and is readable only  
> for that reason. The same wouldn't be true of containers. And  
> "effective\_container" is probably too long to use for the field that is  
> actually the `_common_` case. Thus the above suggestion).

Your proposal looks quite nice.

Then we will have eventually "container" field on objects (not on task only) which sounds good to me. I will prepare patches right now.

Kirill

---