Subject: Re: [RFC PATCH 0/5] Resend - Use procfs to change a syscall behavior
Posted by Pavel Machek on Tue, 08 Jul 2008 10:51:43 GMT
View Forum Message <> Reply to Message

Hi!

>>> An alternative to this solution consists in defining a new field in the
>>> task structure (let's call it next_syscall_data) that, if set, would change
>>> the behavior of next syscall to be called. The sys_fork_with_id() previously
>>> cited can be replaced by
>>> 1) set next_syscall_data to a target upid nr
>>> 2) call fork().
>>
>>
>> ...bloat task struct and
>>
>>
>>> A new file is created in procfs: /proc/self/task/<my_tid>/next_syscall_data.
>>> This makes it possible to avoid races between several threads belonging to
>>> the same process.
>>
>>
>> ...introducing this kind of uglyness.
>>
>> Actually, there were proposals for sys_indirect(), which is slightly
>> less ugly, but IIRC we ended up with adding syscalls, too.

> I had a look at the lwn.net article that describes the sys_indirect()
> interface.
> It does exactly what we need here, so I do like it, but it has the same
> drawbacks as the one you're complaining about:
> . a new field is needed in the task structure
> . looks like many people found it ugly...

> Now, coming back to what I'm proposing: what we need is actually to change
> the behavior of *existing* syscalls, since we are in a very particular
> context (restarting an application).

Changing existing syscalls is _bad_: for backwards compatibility
reasons. strace will be very confusing to read, etc...

> Defining brand new syscalls is very touchy: needs to be careful about the
> interface + I can't imagine the number of syscalls that would be
> needed.

Of course new syscalls is touchy... modifying _existing_ should be
even more touchy.

Pavel

--
(english) http://www.livejournal.com/~pavelmachek
(cesky, pictures) http://atrey.karlin.mff.cuni.cz/~pavel/picture/horses/blog.html

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