

Adrian Bunk <bunk@kernel.org> writes:

> CONFIG_EXPERIMENTAL is a weak hint that some code might not (yet) be in
> a perfect state, but it does not have any semantics regarding
> userspace ABIs.

Code that might not (yet) be in a perfect state sums it up pretty well. There is not plan or expectation to change magic numbers or things like that but the behavior of the code may change as bug and such are fixed.

> A dependency on BROKEN seems more appropriate.

Since you can't select that it seems a little strong.

...

One of the things we talked about at the kernel summit is how almost inevitably when new user space interfaces are introduced there are problems. Someone over looks something, something gets changed to get through the review something like that. There was discussion but no consensus on how do introduce something like that but still allow our selves the ability to fix it. Keeping the code under CONFIG_EXPERIMENTAL is the best suggest I have seen so far. Even if it is slightly expanding the definition of CONFIG_EXPERIMENTAL.

Every place the kernel uses pids is a huge scope. It is very easy to miss something with a scope that wide. So the engineer in me says chances of us missing something are pretty huge. Especially since I know we have bugs in -rc1.

If it turns out that making multiple pid namespaces work is hopeless we can always change the dependency to BROKEN later.

As for ABI and behavioral characteristics currently that is largely well defined. You are supposed to get the exact same thing as you would on the system if you only had a single pid namespace. The places where we have questionable semantics is in the intersections between namespaces.

That is not an area I am willing to stand up and say we got it perfect the first time, I'm going to support our behavior quirks forever if I can find soon enough. Very few applications

will care, and the differences might really matter at some point.

So does any one have any better suggestions on how to deal with features that are enough work you aren't going to get them perfect the first time. You need the code merged or else you can not complete the feature (too many dependencies through out the code). You want early adopters to start playing with the feature so you can get feed back and you can test to make certain everything is going ok. You want to retain the ability to fix implementation details even if those details are user visible, for a time until things seem as good as they can reasonably get.

Roughly that sounds like CONFIG_EXPERIMENTAL to me. But I would be happy to hear if someone has a better idea.

Eric

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