Subject: Re: [PATCH 8/9] signal: Drop signals before sending them to init.
Posted by ebiederm on Thu, 13 Dec 2007 18:50:09 GMT
View Forum Message <> Reply to Message

Oleg Nesterov <oleg@tv-sign.ru> writes:

> OK, if we change the semantics for /sbin/init signals we can avoid
> a lot of problems,

Yes. Otherwise we must track the source of the signals.

>> No. We should treat signals that we process for /sbin/init completely
>> normally.
>
> ... including this one. I am not arguing.
>
>> This gives /sbin/init completely normal signal handling if the signal is
>> ever enqueued. Something trivial to implement and explain.
>
> Well, I am not sure about "explain" though. Unless I missed something
> this makes the semantics a bit special.

Well the semantics are a bit special for init period. I just
make them special in a slightly different way.

> Suppose that init does sigtimedwait() but the handler == SIG_DFL.

Yes that is a bit surprising. However it is still easy to explain.
The signal is never enqueued so sigtimedwait never gets the chance
to do anything with it. Interestingly enough this is not a problem
for the current sysvinit.

sysvinit does this at start up:

```
	/*
	 *	Ignore all signals.
	 */
	for(f = 1; f <= NSIG; f++)
		SETSIG(sa, f, SIG_IGN, SA_RESTART);
```

So everything is initialized to SIG_IGN by userspace, in the common case.
Which means none of this special case logic will actually kick in, except
for SIGKILL and SIGSTOP. The signals we can't change.

Eric

____
Containers mailing list

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