Subject: Re: [PATCH 1/3] Signal semantics for /sbin/init
Posted by Sukadev Bhattiprolu on Mon, 08 Oct 2007 15:42:00 GMT
View Forum Message <> Reply to Message

Serge E. Hallyn [serue@us.ibm.com] wrote:
| Quoting sukadev@us.ibm.com (sukadev@us.ibm.com):
| > | > | > One solution I was thinking of was to possibly queue pending blocked
| > | > | > signals to a container init seperately and then requeue them on the
| > | > | > normal queue when signals are unblocked. Its definitely not an easier
| > | > | > solution, but might be less intrusive than the "signal from parent ns
| > | > | > flag" solution.
| >
| > Hmm. Stumbled upon an issue while coding this up.
| >
| > For real time signals, is the order in which signals are delivered
| > important ?
|
| A very quick, inadequate google search suggests that while order is
| important, the order in which they should be delivered is in increasing
| signal number. So that should be easy enough to maintain with this
| type of patch, right?

Yes. I could stick all the signals to the end of the queue and the
existing code should pick them up in proper order.

Thanks,

Suka

|
| -serge
|
| > If so, the above solution won't work, because when requeing
| > signals from pid namespace to the task, we may lose the order of signals.
| >
| > i.e:
| > - signal sig1 is blocked.
| >
| > - sig1 is posted and queued on the ns->cinit_blocked_pending queue
| >
| > - just as we enter sigprocmask() to unblock sig1, if sig2 is
| >   posted, we queue sig2 on task->pending queue since sig2 is not
| >   blocked.
| >
| > - when we actually unblock sig1, if we requeue ns->cinit_blocked_pending
| >   signals (i.e sig1) after sig2 we end up reversing the order of the
| >   signals sig1 and sig2.
| >

| >   if we requeue the ns-> signals before sig2 and another sig1
| >   was received after the sig2 and just before the unblock, we
| >   again lose the order.
| >
| > The order of signals is not important to legacy signals (< SIGRTMIN) but is
| > probably an issue for signals SIGRTMIN..SIGRTMAX.

_____