

| > | > One solution I was thinking of was to possibly queue pending blocked  
| > | > signals to a container init separately 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 ? 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.

---

Containers mailing list  
[Containers@lists.linux-foundation.org](mailto:Containers@lists.linux-foundation.org)  
<https://lists.linux-foundation.org/mailman/listinfo/containers>

---