

>>>> A naive question, because I have not followed C/R closely. How do you
>>>> deal with the case that other processes may be reading from the queue?
>>>> (Or is that disabled during checkpointing?)
>>>>
>>>
>>> To be honest, in this case behaviour in user-space is unpredictable.
>>> I.e. if you have, for example, 5 messages in queue and going to peek them
>>> all, and another process is reading the queue in the same time, then,
>>> most
>>> probably, you won't peek all the 5 and receive ENOMSG.
>>> But this case can be easily handled by user-space application (number of
>>> messages in queue can be discovered before peeking).
>>>
>>> Note, that in CRIU IPC resources will be collected when all processes to
>>> migrate are frozen.
>>
>>
>> Perhaps I am missing something fundamental, but how can C/R sanely do
>> anything at all here?
>>
>> For example, suppose a process reads and processes a message after you
>> read it with MSG_COPY. Then the remaining messages are all shifted by
>> one position, and you are going to miss reading one of them. IIUC the
>> idea of MSG_COPY is to allow you to retrieve a copy of all messages in
>> the list. It sounds like there's no way this can be done reliably. So,
>> what possible use does the operation have?
>>
>
> First of all, this problem exist as is regardless to C/R feature or this
> patch set. If you share some resource (like message queue in this particular
> case) system-wide, then any process A can read out a message, which was send
> by process B to process C. So, when processes uses IPC message queues, they
> should be designed to handle such failures.
>
> Second, it's up to user-space how to handle such things. It's implied, that
> user, trying to migrate some process, holding one end of queue, will also
> migrate another process, holding second end.
>
> Third, there is IPC namespace, which isolates IPC objects. It can be used
> for safe migration of process tree.

Is there somewhere a *detailed* description of how this feature would be used? Lacking that, it's really hard to see how anything sane and reliable can be done with MSG_COPY.

--

Michael Kerrisk

Linux man-pages maintainer; <http://www.kernel.org/doc/man-pages/>

Author of "The Linux Programming Interface"; <http://man7.org/tlpi/>
