

# Re: Sharing memory between kernelspace and userspace

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-09/msg03871.html>

---

- *From:* "Ray Lee" <[madrabbit@xxxxxxxx](mailto:madrabbit@xxxxxxxx)>
  - *Date:* Thu, 14 Sep 2006 11:05:39 -0700
- 

On 9/13/06, Jonathan Day <[imipak@xxxxxxxx](mailto:imipak@xxxxxxxx)> wrote:

1. I need a kernel driver to be able to allocate and deallocate, on a totally dynamic basis, userspace memory that is reachable by multiple applications.

Is there no possible way you can do the memory allocation fully in userspace? Let userspace allocate shared memory visible to multiple processes, and pass that into the kernel for it to write to.

If this is as high-speed as you imply, then you *\*really\** don't want to be doing this on the fly (let alone from inside the kernel). All the high-bandwidth stuff I've worked on preallocated its buffers, as later on in the processing there are no hard guarantees as to either how much memory one can acquire *\*or\** how long it will take to do so. Either one of those is a deal-breaker when dealing with fast data rates.

3. I would truly value some suggestions on which of the many ways of communicating with the kernel would offer the best way to handle a truly horrible number of very simple signals. Speed, here, is an absolute must. According to my boss, residual sanity on my part is not.

Just send a message down an fd? The wakeup/context-switch is the expensive part of that, I think, at least when dealing with only a few dozen fds. Or change it to be level- rather than edge-triggered, to coalesce the horrific number of events getting crammed down the processing side.

It sounds like this is the cheapest part of the process, so not worth the effort of optimizing it as much as finding better ways to do the rest of it.

## Re: Sharing memory between kernelspace and userspace

I'm having what is probably the world's second-dumbest problem. What I want to do is have a driver in kernelspace be able to allocate multiple chunks of memory that can be shared with userspace without having to do copies.

Have userspace ask the driver if anyone has allocated it yet, if not, userspace allocates it and hands the pointer to it back to the driver so that it can hand it back out to the other calling processes who also want to subscribe to that data stream.

If another userspace process races with the first trying to ask if a buffer is available (first asked already, but hasn't yet given the driver the info), then the driver puts the second process to sleep until it's ready to hand back the buffer.

There are several problems, however, that make this nasty. First, since the time before the kernel driver or user application can start (and therefore finish) processing a block of data is non-deterministic and there is a requirement the mechanism be as close to non-blocking as possible, so I need to be able to create and destroy chunks entirely on-the-fly with the least risk of either the driver or the application ending up with unexpectedly invalid pointers.

Creating chunks on the fly is, I think, an operation that can take an arbitrary amount of time. If you *\*really\** want to do that, you should instead just allocate as much as possible in the first place, and then you do your own memory management on blocks inside of that, never invoking the kernel's memory services after the initial allocation.

The second problem is that the interface between kernelspace and userspace is handling messages rather than packets, so I've absolutely no idea in advance how big a chunk is going to be. That is only known just prior to the data being put in the chunk allocated for it. Messages can be big (the specs require the ability to send a message up to 2Gb in size) which – if I'm reading the docs correctly – means I can't create the chunks in kernelspace.

Surely this is just an issue of writing the chunk to free memory? Either you have the memory available or you don't. If you do, no problem. If you don't, then you're screwed regardless, as allocating on the fly is not guaranteed to help you out.

Re: Sharing memory between kernelspace and userspace

Good luck,

Ray

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>