

Re: sysctl locking

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/current/2004-12/0736.html>

From: John-Mark Gurney (gurney_j_at_resnet.uoregon.edu)

Date: 12/17/04

Date: Fri, 17 Dec 2004 13:00:05 -0800

To: Max Laier <max@love2party.net>

Max Laier wrote this message on Mon, Dec 13, 2004 at 12:32 +0100:

- > *If we come to the conclusion that it is required to protect these values*
- > *better, I suggest the following:*
- >
- > *1) Extend sysctl_add_oid() to accept an additional mutex argument.*
- > *2) Extend the simple sysctl handler to use this mutex to protect the actual*
- > *write(?read?). We must not hold the mutex during the useland copy in/out so*
- > *we must move to temporary storage.*
- > *3) To maintain the current API and behavior we use &Giant as the default*
- > *fallback argument. This might need some extension for complex handler (i.e.*
- > *no mutex given -> acquire Giant before calling the complex handler).*
- >
- > *What do people think of this? Does it make any sense? Should we be concerned*
- > *at all? Does the extension make sense? Comments?*

If all one is doing is a single read or a single write, then a mutex is not needed... Since you are not synchronizing with something else or doing a write based on a read (i.e atomic increment, etc), the read/write could happen at any order..

Only if the entire value can not be written atomicly (like 64 bit ints of 486's, Pentiums have an 8 byte xchng op) would a mutex be helpful...

In the kqueue case, there is a function knlist_empty that originally would obtain the lock, read the value and then unlock.. This was pointless since any state returned by the function would be stale and any decision made on it would be bogus... It was changed to require that the calling function hold the lock so that the state was consistent with the subsequent decision...

If a sysctl needs that level of protection, it seems like they need to be writing their own custom handler to handle the required interactions..

--

John-Mark Gurney

Voice: +1 415 225 5579

"All that I will do, has been done, All that I have, has not."

freebsd-current@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-current>

Re: sysctl locking

freebsd-current: Re: sysctl locking

To unsubscribe, send any mail to "freebsd-current-unsubscribe@freebsd.org"