

Re: msleep() on recursively locked mutexes

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/hackers/2007-04/msg00238.html>

- *From:* Robert Watson <rwatson@xxxxxxxxxxxxx>
 - *Date:* Sat, 28 Apr 2007 10:57:34 +0100 (BST)
-

On Fri, 27 Apr 2007, Julian Elischer wrote:

Basically you shouldn't have a recursed mutex FULL STOP. We have a couple of instances in the kernel where we allow a mutex to recurse, but they had to be hard fought, and the general rule is "Don't". If you are recursing on a mutex you need to switch to some other method of doing things. e.g. reference counts, turnstiles, whatever.. use the mutex to create these but don't hold the mutex for long enough to need to recurse on it. A mutex should generally lock, dash-in and work, unlock. We have some cases where that is not true, but we are trying to get rid of them, not add more.

Most of these instances have to do with legacy code and data structures that involve high levels of code recursion and reentrance. This is frequently an unreliable way to organize code anyway, and often involves other bugs that are less visible. Over time, it's my hope that we can eliminate quite a few sources of remaining lock recursion, but there are some tricky cases involving repeated callbacks between layers that make that harder. For example, in the socket/network pcb relationship, there's a lack of clarity on which side drives the overlapping state machines present in both sets of data structures. Over time, we're migrating towards a model in which the socket infrastructure is more of a "library" in service to network protocols that will drive the actual transitions, but in the mean time, lock recursion is required.

For any significantly rewritten or new code, I would expect that recursion would be avoided in almost all cases.

Robert N M Watson
Computer Laboratory
University of Cambridge

freebsd-hackers@xxxxxxxxxxxx mailing list

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

To unsubscribe, send any mail to "freebsd-hackers-unsubscribe@xxxxxxxxxxxx"