

freebsd-arch: mtx\_lock\_recurse/mtx\_unlock\_recurse functions (proof-of-concept).

# mtx\_lock\_recurse/mtx\_unlock\_recurse functions (proof-of-concept).

**Source:** <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2004-04/0017.html>

---

**From:** Pawel Jakub Dawidek ([pjd\\_at\\_FreeBSD.org](mailto:pjd_at_FreeBSD.org))

**Date:** 04/08/04

Date: Thu, 8 Apr 2004 23:36:47 +0200

To: [freebsd-arch@freebsd.org](mailto:freebsd-arch@freebsd.org)

Hi.

As was discussed, it will be helpful to have functions, that are able to acquire lock recursively, even if lock itself isn't recursable.

Here is a patch, that should implement this functionality:

[http://people.freebsd.org/~pjd/patches/mtx\\_lock\\_recurse.patch](http://people.freebsd.org/~pjd/patches/mtx_lock_recurse.patch)

I also added a KASSERT() to protect against mixed locking modes, e.g.:

```
mtx_lock(&mtx);  
[...]  
mtx_lock_recurse(&mtx);  
[...]  
mtx_unlock(&mtx);  
[...]  
mtx_unlock_recurse(&mtx);
```

This is not permitted, but this should work just fine:

```
mtx_lock(&mtx);  
[...]  
mtx_lock_recurse(&mtx);  
[...]  
mtx_unlock_recurse(&mtx);  
[...]  
mtx_unlock(&mtx);
```

--

Pawel Jakub Dawidek  
[pjd@FreeBSD.org](mailto:pjd@FreeBSD.org)  
FreeBSD committer

<http://www.FreeBSD.org>  
<http://garage.freebsd.pl>  
Am I Evil? Yes, I Am!

---

freebsd-arch: mtx\_lock\_recurse/mtx\_unlock\_recurse functions (proof-of-concept).

- application/pgp-signature attachment: stored