

Re: atomic reference counting primitives.

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2004-05/0054.html>

From: John Baldwin (jhb_at_FreeBSD.org)

Date: 05/21/04

To: freebsd-arch@FreeBSD.org

Date: Fri, 21 May 2004 09:59:24 -0400

On Thursday 20 May 2004 10:54 pm, M. Warner Losh wrote:

> *In message:*

> <Pine.BSF.4.21.0405201340590.72391-100000@InterJet.elischer.org>

>

> *Julian Elischer* <julian@elischer.org> writes:

> : *This has been raised before but I've come across uses for it again and*

> : *again so I'm raising it again.*

> : *JHB once posted some atomic referenc counting primitives. (Do you still*

> : *have them John?)*

> : *Alfred once said he had soem somewhere too, and other s have commentted*

> : *on this before, but we still don't seem to have any.*

> :

> : *every object is reference counted with its own code and*

> : *sometimes it's done poorly.*

> :

> : *Some peiople indicated that there are cases where a generic refcounter*

> : *can not be used and usd this as a reason to not have one at all.*

> :

> : *So, here are some possibilities..*

> : *my first "write it down without too much thinking" effort..*

> :

> : *typedef {mumble} refcnt_t*

> :

> : *refcnt_add(refcnt_t *)*

> : *Increments the reference count.. no magic except to be atomic.*

> :

> :

> : *int refcnt_drop(refcnt *, struct mutex *)*

> : *Decrements the refcount. If it goes to 0 it returns 0 and locks the*

> : *mutex (if the mutex is supplied)..*

>

> *What prevents refcnt_add() from happening after ref count drops to 0?*

> *Wouldn't that be a race? Eg, if we have two threads:*

>

>

> *Thread A Thread B*

>

freebsd-arch: Re: atomic reference counting primitives.

> *objp = lookup();*
> *[1] refcnt_drop(&objp->ref, &objp->mtx);*
> *[2] refcnt_add(&obj->ref);*
> *BANG!*
>
> *If [1] happens before [2], then bad things happen at BANG! If [2]*
> *happens before [1], then the mutex won't be locked at BANG and things*
> *is good. Thread A believes it has a valid reference to objp after the*
> *refcnt_add and no way of knowing otherwise.*
>
> *Is there a safe way to use the API into what you are proposing?*

This situation can't happen if you are properly using reference counting. For the reference count to be at 1 in thread B, it has to have the only reference meaning that the object has already been removed from any lists, etc.

--
John Baldwin <jhb@FreeBSD.org> <>< <http://www.FreeBSD.org/~jhb/>
"Power Users Use the Power to Serve" = <http://www.FreeBSD.org>

freebsd-arch@freebsd.org mailing list
<http://lists.freebsd.org/mailman/listinfo/freebsd-arch>
To unsubscribe, send any mail to "freebsd-arch-unsubscribe@freebsd.org"