

Re: close() of active socket does not work on FreeBSD 6

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2006-12/msg00070.html>

- *From:* Julian Elischer <jelischer@xxxxxxxxxxxx>
 - *Date:* Wed, 20 Dec 2006 17:48:02 -0800
-

David Xu wrote:

On Thursday 21 December 2006 02:18, Daniel Eischen wrote:

On Wed, 20 Dec 2006, Robert Watson wrote:

On Wed, 13 Dec 2006, Daniel Eischen wrote:

Anyway, this was just a thought/idea. I don't mean to argue against any of the other reasons why this isn't a good idea.

Whatever may be implemented to solve this issue will require a fairly serious re-working of how we implement file descriptor reference counting in the kernel. Do you propose similar "cancellation" of other system calls blocked on the file descriptor, including select(), etc? Typically these system calls interact with the underlying object associated with the file descriptor, not the file descriptor itself, and often, they act directly on the object and release the file descriptor before performing their operation. I think before we can put any reasonable implementation proposal on the table, we need a clear set of requirements:

[...]

While providing Solaris-like semantics here makes some amount of sense, this is a very tricky area, and one where we're still refining performance behavior, reference counting behavior, etc. I

Re: close() of active socket does not work on FreeBSD 6

don't think
there will be any easy answers, and we need to think through
the semantic
and performance implications of any change very carefully
before starting
to implement.

I don't think the behavior here has to be any different that
what we currently (or desire to) do with regard to (unblocked)
signals interrupting threads waiting on IO. You can spend
a lot of time thinking about how close() should affect IO
operations on the same file descriptor, but a very simple
approach is to treat them the same as if the operations were
interrupted by a signal. I'm not suggesting it is implemented
the same way, just that it seems to make a lot of sense to me
that the behavior is consistent between the two.

I think the main concern is if we will record every thread using a
fd, that means, when you call read() on a fd, you record your
thread pointer into the fd's thread list, when one wants to close
the fd, it has to notify all the threads in the list, set a flag
for each thread, the flag indicates a thread is interrupted
because the fd was closed, when the thread returns from deep code
path to read() syscall, it should check the flag, and return EBADF to
user if it was set. whatever, a reserved signal or TDF_INTERRUPT may
interrupt a thread. but since there are many file operations, I don't
know if we are willing to pay such overheads to every file syscall, extra locking is not
welcomed.

I think you are only interested in threads that are sleeping..
so you allow a sleeping thread to save a pointer to the fd (or whatever) on which it is sleeping, along with the
sleep address.

items that are not sleeping are either already returning, or are going to sleep, in which case they can check at
that time.

Regards,
David Xu

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

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

Re: close() of active socket does not work on FreeBSD 6