Filesystems broken due to "KKASSERT(count & TOK_COUNTMASK);"

Many fs including HAMMER2 are broken due to this assert failure. Confirmed the panic with HAMMER2 and ext2.

It didn't happen a few months ago.

433 static __inline
434 void
435 _lwkt_reltokref(lwkt_tokref_t ref, thread_t td)
436 {
437     ...
438     count = atomic_fetchadd_long(&tok->t_count, TOK_INCR);
439     KKASSERT;        /* count prior */    <----------
440 }

History

#1 - 03/15/2021 08:33 AM - tkusumi
Looks like this happens with kernel module filesystems.

I'm going to comment out this KASSERT since it works fine without it. This assert may be correct, but then the real bug needs to be fixed.

#2 - 03/15/2021 10:37 AM - dillon
Negative, that assertion cannot be removed. It asserts that there is a token to release. If there isn't one that means there is a get/rel mismatch somewhere that needs to be tracked down and located.

-Matt

#3 - 03/15/2021 10:39 AM - dillon
What is needed is a kernel core to determine which call stack is misbehaving. It is possible that the mismatch is incurred at a deeper level that has already returned back up the stack, so it can sometimes be difficult to locate. But any sort of mismatch is critical and needs to be found and fixed.

-Matt

#4 - 03/15/2021 12:27 PM - tkusumi
I'll bisect it when I have time. Shouldn't be super difficult if a commit within the past few months changed behavior.

#5 - 03/15/2021 01:21 PM - dillon
If you can find a way to easily reproduce the panic I can go looking too. I have not hit this panic at all yet. It really should have caught whatever the issue was with some of the earlier assertions in those code paths. Every token is 100% tracked in a per-thread array so its a bit worrying that a bug could make it that deep into the token release code.

There have been some notable changes in some of those paths. The rename code paths being the most notable (ad1212685b9caac64c086a), some changes to the exec code, and some optimizations in the GETATTR path.

-Matt