Frightening small children and disconcerting grown-ups: Concurrency in the Linux kernel

Jade Alglave, Luc Maranget, Paul McKenney,
Andrea Parri and Alan Stern

Concurrency in Linux can be a contentious topic

Model	URL
SPARC	[McKenney, 2001, Spraul, 2001]
LK	[Vaddagiri, 2005]
LK	[McKenney, 2007, Alglave et al., 2013]
LK	[Corbet, 2008]
LK	[Olsa, 2009]
LK	[Heo, 2010]
LK/Itanium	[Blanchard, 2011, Miller, 2011]
LK	[Corbet, 2012]
Itanium	[McKenney, 2013b, Luck, 2013a, Luck, 2013b, Zijlstra, 2013]
Intel	[McKenney, 2013a, Kleen, 2013]
LK/C11	[Corbet, 2014b, Corbet, 2014c]
LK	[Corbet, 2014a]
Alpha	[Torvalds, 2015]
LK	[Feng, 2015]
ARM64	[Deacon, 2015a]
LK	[Deacon, 2015b]
MIPS	[Yegoshin, 2016a, Yegoshin, 2016b, Yegoshin, 2016c, Torvalds, 2016b]
PowerPC	[Feng, 2016a, Feng, 2016b, Ellerman, 2016]
ARM64	[Deacon, 2016]
LK/C11	[Corbet, 2016]
LK	[Molnar, 2017]

“Still confusion situation all round”[sic] [Ziljstra, 2013]

Existing documentation

- [Howells et al., 2017] lists what orderings are guaranteed;
- [Miller, 2017] summarises semantics of read-modify-writes;
- [McKenney, 2017a] documents ways of avoiding counterproductive optimisations.

But [Gorman, 2013]

If Documentation/memory-barriers.txt could not be used to frighten small children before, it certainly can now.

Also [Howells et al., 2017]:

This document is not a specification; it is intentionally (for the sake of brevity) and unintentionally (due to being human) incomplete. [. . .] in case of any doubt (and there are many) please ask.

And anyway [Torvalds, 2012]

*With specs, there really *are* people who spend years discussing what the meaning of the word "access" is or similar [. . .]. Combine that with a big spec that is 500+ pages in size and then try to apply that all to a project that is 15 million lines of code and sometimes *knowingly* has to do things that it simply knows are outside the spec [. . .]"*

What we offer

- a formal consistency model
- written in the cat language
- thus executable within the herd tool

“[I]t is your kernel, so what is your preference?” [McKenney, 2016a]

A common denominator of hardware models? [Torvalds, 2016a]

Weak memory ordering is [. . .] hard to think about [. . .] So the memory ordering rules should [. . .] absolutely be as tight as at all humanly possible, given real hardware constraints.

Not an envelope for the architectures it supports? [Molnar, 2013]

> *it's not true that Linux has to offer a barrier and locking model that panders to the weakest (and craziest!) memory ordering model amongst all the possible Linux platforms—theoretical or real metal. Instead what we want to do is to consciously, intelligently _pick_ a sane, maintainable memory model and offer primitives for that—at least as far as generic code is concerned. Each architecture can map those primitives to the best of its abilities.*

The LK should have a model of its own [Torvalds, 2012]

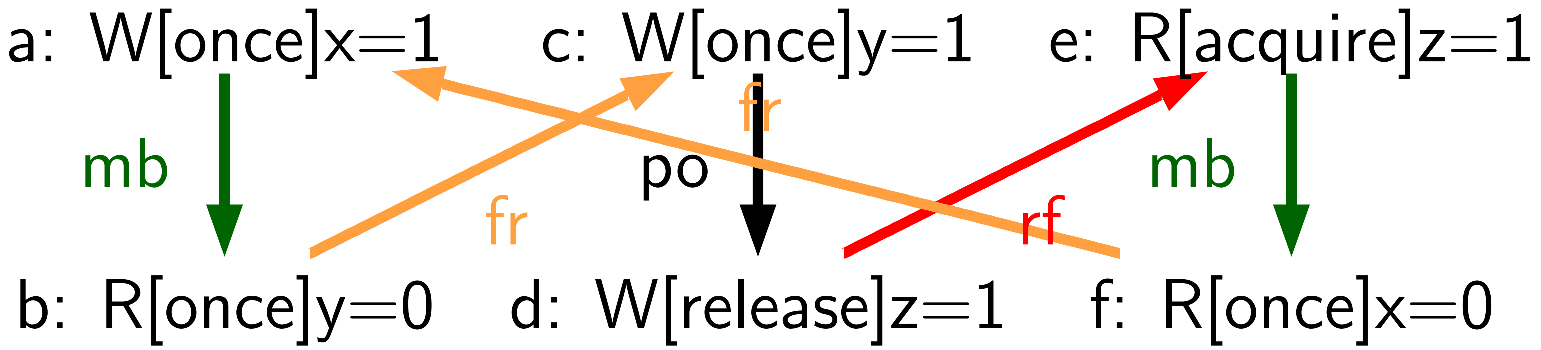
> *I do not believe for a second that we can actually use the C11 memory model in the kernel [. . .] We will continue to have to do things that are "outside the specs" [. . .] with models that C11 simply doesn't cover.*

Core model

An example from PeterZ

`https://www.spinics.net/lists/kernel/msg2421883.html`

PeterZ: Forbidden

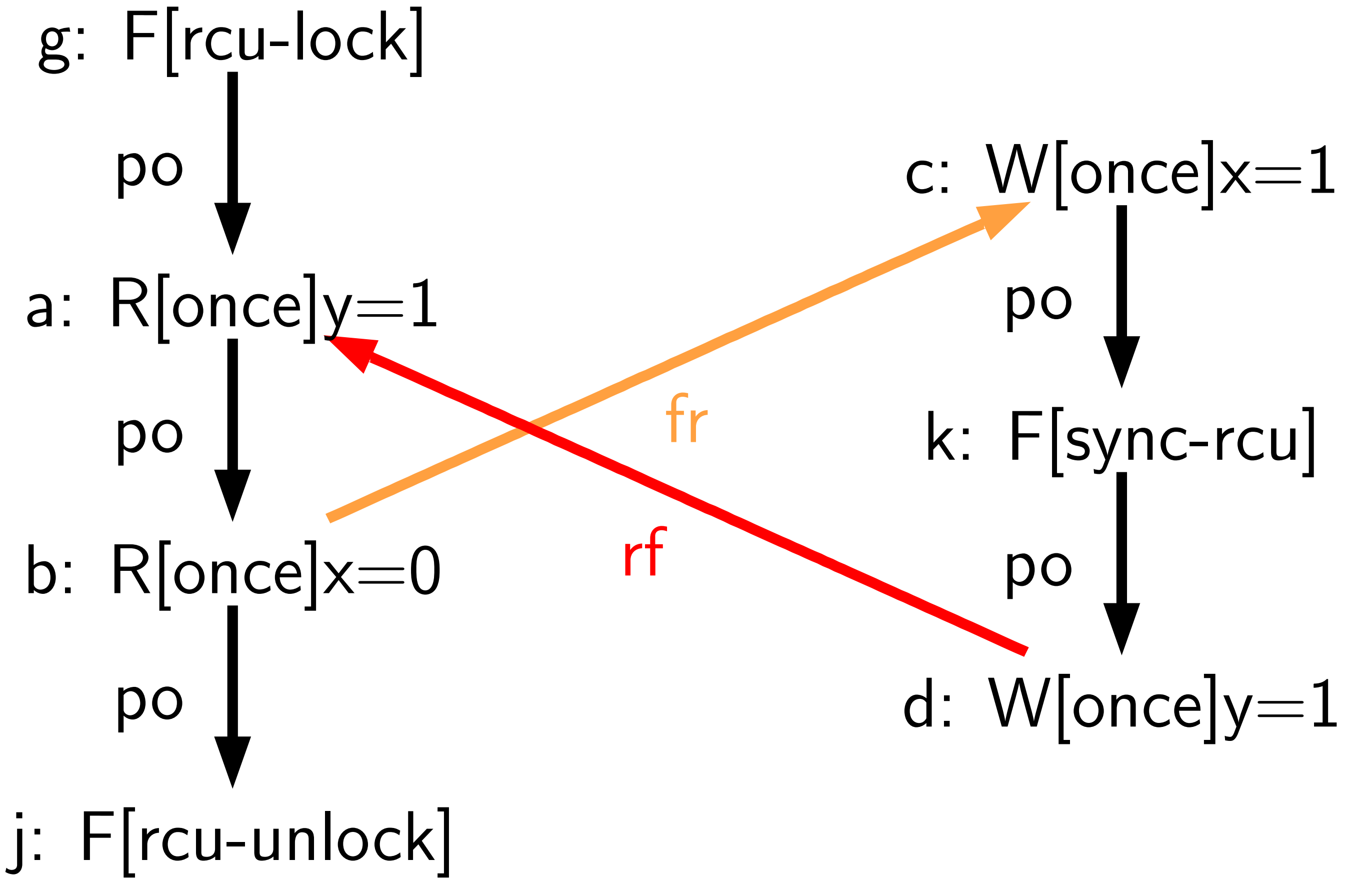


RCU

Fundamental law of RCU [McKenney et al., 2013]

Read-side critical sections cannot span grace periods.

RCU-MP: Forbidden



Validating the model

Experimentally

	Model	Power8	ARMv8	ARMv7	X86	C11
LB	Allow	0/15G	0/10G	0/3.0G	0/33G	Allow
LB+ctrl+mb	Forbid	0/17G	0/5.1G	0/4.4G	0/18G	Allow
WRC	Allow	741k/7.7G	13k/5.2G	0/1.6G	0/17G	Allow
WRC+wmb+acq	Allow	0/7.5G	0/4.6G	0/1.6G	0/16G	Forbid
WRC+po-rel+rmb	Forbid	0/7.7G	0/5.3G	0/1.6G	0/17G	Forbid
SB	Allow	4.4G/15G	2.4G/10G	429M/3.0G	765M/33G	Allow
SB+mbs	Forbid	0/15G	0/10G	0/3.0G	0/33G	Forbid
MP	Allow	57M/15G	104M/10G	3.0M/3.0G	0/33G	Allow
MP+wmb+rmb	Forbid	0/15G	0/9.4G	0/2.6G	0/33G	Forbid
PeterZ-No-Synchro	Allow	26M/4.6G	3.6M/900M	10k/380M	351k/7.2G	Allow
PeterZ	Forbid	0/8.7G	0/2.5G	0/2.2G	0/9.1G	Allow
RCU-deferred-free	Forbid	0/256M	0/576M	0/15M	0/25M	—
RCU-MP	Forbid	0/672M	0/336M	0/336M	0/336M	—
RWC	Allow	88M/11G	94M/4.8G	7.5M/1.6G	5.6M/17G	Allow
RWC+mbs	Forbid	0/8.7G	0/2.5G	0/2.2G	0/9.1G	Allow

Socially

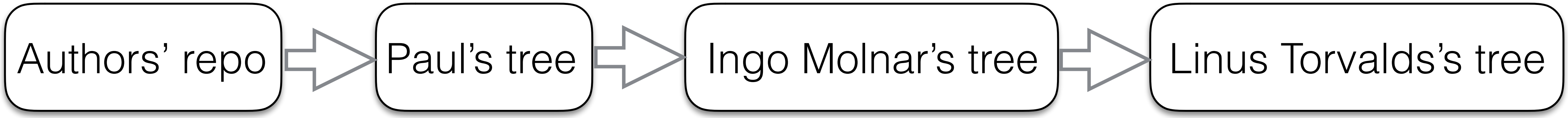
`https://www.spinics.net/lists/kernel/msg2421883.html`

Issues that our work helped address

LK issue	URL
locking on ARM64	[Deacon, 2015a]
ambiguities in [Howells et al., 2017]	[McKenney, 2016c]
ambiguities in RCU documentation	[McKenney, 2016b]
CPU hotplug	[Zijlstra, 2016]
assumption about lock-unlock	[McKenney, 2017b]
semantics of `spin_unlock_wait`	[Torvalds, 2017]

As well as:

- updates to documentation [McKenney, 2016b, McKenney, 2016c, Howells et al., 2017];



Authors' repo

Paul's tree

Ingo Molnar's tree

Linus Torvalds's tree

2018-01-18

2018-01-31

aim:
v4.17 merge window
April 2018

Alglave, J., Kroening, D., and Tautschnig, M. (2013).
Partial orders for efficient Bounded Model Checking of concurrent software.
In Computer Aided Verification (CAV), volume 8044 of LNCS, pages 141–157. Springer.

Blanchard, A. (2011).
RE: [PATCH] smp_call_function_many SMP race.
https://lkml.org/lkml/2011/1/11/489.

Corbet, J. (2008).
The lockless page cache.
https://lwn.net/Articles/291826/.

Corbet, J. (2012).
ACCESS_ONCE().
https://lwn.net/Articles/508991/.

Corbet, J. (2014a).
ACCESS_ONCE() and compiler bugs.
https://lwn.net/Articles/624126/.

Corbet, J. (2014b).
C11 atomic variables and the kernel.
https://lwn.net/Articles/586838/.

Corbet, J. (2014c).
C11 atomics part 2: “consume” semantics.
https://lwn.net/Articles/588300/.

Corbet, J. (2016).
Time to move to C11 atomics?
https://lwn.net/Articles/691128/.

Deacon, W. (2015a).
[PATCH] arm64: spinlock: serialise spin_unlock_wait against concurrent lockers.
https://marc.info/?l=linux-arm-kernel&m=144862480822027.

Deacon, W. (2015b).
Re: [PATCH] arm64: spinlock: serialise spin_unlock_wait against concurrent lockers.

https://marc.info/?l=linux-arm-kernel&m=144898777124295.

Deacon, W. (2016).
[PATCH v2 1/3] arm64: spinlock: order spin_{is_locked, unlock_wait} against local locks.
http://lists.infradead.org/pipermail/linux-arm-kernel/2016-June/434765.html.

Desnoyers, M., McKenney, P. E., Stern, A. S., Dagenais, M. R., and Walpole, J. (2012).
User-level implementations of Read-Copy Update.
IEEE Trans. Parallel Distrib. Syst., 23(2):375–382.

Ellerman, M. (2016).
[PATCH v3] powerpc: spinlock: Fix spin_unlock_wait().
https://marc.info/?l=linux-kernel&m=146521336230748&w=2.

Feng, B. (2015).
Re: [PATCH 4/4] locking: Introduce smp_cond_acquire().

https://marc.info/?l=linux-kernel&m=144723482232285.

Feng, B. (2016a).
[PATCH v2] powerpc: spinlock: Fix spin_unlock_wait().
https://marc.info/?l=linux-kernel&m=146492558531292&w=2.

Feng, B. (2016b).
[PATCH v4] powerpc: spinlock: Fix spin_unlock_wait().
https://marc.info/?l=linuxppc-embedded&m=146553051027169&w=2.

Gorman, M. (2013).
LWN Quotes of the week, 2013-12-11.
http://lwn.net/Articles/575835/.

Heo, T. (2010).
[PATCH 3/4] scheduler: replace migration_thread with cpuhog.

https://marc.info/?l=linux-kernel&m=126806371630498.

Howells, D., McKenney, P. E., Deacon, W., and Zijlstra, P. (2017).
Linux kernel memory barriers.
https://www.kernel.org/doc/Documentation/memory-barriers.txt.

Kleen, A. (2013).
Re: [patch v6 4/5] MCS lock: Barrier corrections.
https://marc.info/?l=linux-mm&m=138619237606428.

Luck, T. (2013a).
RE: Does Itanium permit speculative stores?
https://marc.info/?l=linux-kernel&m=138427925823852.

Luck, T. (2013b).
RE: Does Itanium permit speculative stores?

https://marc.info/?l=linux-kernel&m=138428203211477.

McKenney, P. (2001).
RFC: patch to allow lock-free traversal of lists with insertion.
https://lists.gt.net/linux/kernel/223665#223508.

McKenney, P. (2013a).
Re: [patch v6 4/5] MCS lock: Barrier corrections.
https://marc.info/?l=linux-mm&m=138540258209368.

McKenney, P. (2016a).
Re: [RFC][PATCH] mips: Fix arch_spin_unlock().
http://lkml.kernel.org/r/20160202120252.GI6719@linux.vnet.ibm.com.

McKenney, P. (2017a).
PROPER CARE AND FEEDING OF RETURN VALUES FROM rcu_dereference().
https://www.kernel.org/doc/Documentation/RCU/rcu_dereference.txt.

McKenney, P. E. (2007).
QRCU with lockless fastpath.
https://lwn.net/Articles/223752/.

McKenney, P. E. (2013b).
Does Itanium permit speculative stores?
https://marc.info/?l=linux-kernel&m=138419150923282.

McKenney, P. E. (2016b).
documentation: Present updated RCU guarantee.
https://patchwork.kernel.org/patch/9428001/.

McKenney, P. E. (2016c).
documentation: Transitivity is not cumulativity.
http://www.spinics.net/lists/linux-tip-commits/msg32905.html.

McKenney, P. E. (2017b).
srcu: Force full grace-period ordering.
https://patchwork.kernel.org/patch/9535987/.

McKenney, P. E., Desnoyers, M., Jiangshan, L., and Triplett, J. (2013).
The RCU-barrier menagerie.
https://lwn.net/Articles/573497/.

Miller, D. S. (2017).
Semantics and behavior of atomic and bitmask operations.
https://www.kernel.org/doc/core-api/atomic_ops.rst.

Miller, M. (2011).
[PATCH 0/4 v3] smp_call_function_many issues from review.
https://marc.info/?l=linux-kernel&m=130021726530804.

Molnar, I. (2013).
Re: [patch v6 4/5] MCS lock: Barrier corrections.
https://marc.info/?l=linux-mm&m=138513336717990&w=2.

Molnar, I. (2017).

Re: [PATCH v2 0/9] remove spin_unlock_wait().
https://marc.info/?l=linux-kernel&m=149942365927828&w=2.

Olsa, J. (2009).
[PATCHv5 2/2] memory barrier: adding smp_mb__after_lock.
https://marc.info/?l=linux-netdev&m=124839648220382&w=2.

Spraul, M. (2001).
Re: RFC: patch to allow lock-free traversal of lists with insertion.
http://lkml.iu.edu/hypermail/linux/kernel/0110.1/0410.html.

Torvalds, L. (2012).
Re: Memory corruption due to word sharing.
https://gcc.gnu.org/ml/gcc/2012-02/msg00013.html.

Torvalds, L. (2015).
Re: [patch 4/4] locking: Introduce smp_cond_acquire().

http://lkml.kernel.org/r/CA+55aFyXu5iFJfdm7o-RKUm_9a850iSzeM+whmtUAotkY0EvTw@mail.gmail.com.

Torvalds, L. (2016a).
Re: [rfc][patch] mips: Fix arch_spin_unlock().
https://lkml.org/lkml/2016/2/2/80.

Torvalds, L. (2016b).
Re: [v3,11/41] mips: reuse asm-generic/barrier.h.
https://marc.info/?l=linux-kernel&m=145384764324700&w=2.

Torvalds, L. (2017).
Re: [GIT PULL rcu/next] RCU commits for 4.13.
https://lkml.org/lkml/2017/6/27/1052.

Vaddagiri, S. (2005).
[PATCH] Fix RCU race in access of nohz_cpu_mask.
http://lkml.iu.edu/hypermail/linux/kernel/0512.0/0976.html.

- Yegoshin, L. (2016a).
  Re: [v3,11/41] mips: reuse asm-generic/barrier.h.
  https://marc.info/?l=linux-kernel&m=145263153305591&w=2.
- Yegoshin, L. (2016b).
  Re: [v3,11/41] mips: reuse asm-generic/barrier.h.
  https://marc.info/?l=linux-kernel&m=145280444229608&w=2.
- Yegoshin, L. (2016c).
  Re: [v3,11/41] mips: reuse asm-generic/barrier.h.
  https://marc.info/?l=linux-kernel&m=145280241129008&w=2.
- Zijlstra, P. (2013).
  Re: Does Itanium permit speculative stores?
  https://marc.info/?l=linux-kernel&m=138428080207125.
- Zijlstra, P. (2016).

[tip:perf/urgent] perf/core: Fix sys_perf_event_open() vs. hotplug.
https://www.spinics.net/lists/kernel/msg2421883.html.

Ziljstra, P. (2013).
Re: [patch v6 4/5] MCS lock: Barrier corrections.
https://marc.info/?l=linux-mm&m=138514629508662&w=2.