

Read the F-ing Papers!

This document describes RCU-related publications, and is followed by the corresponding bibtex entries. A number of the publications may be found at <http://www.rdrop.com/users/paulmck/RCU/>.

The first thing resembling RCU was published in 1980, when Kung and Lehman [Kung80] recommended use of a garbage collector to defer destruction of nodes in a parallel binary search tree in order to simplify its implementation. This works well in environments that have garbage collectors, but most production garbage collectors incur significant overhead.

In 1982, Manber and Ladner [Manber82,Manber84] recommended deferring destruction until all threads running at that time have terminated, again for a parallel binary search tree. This approach works well in systems with short-lived threads, such as the K42 research operating system. However, Linux has long-lived tasks, so more is needed.

In 1986, Hennessy, Osisek, and Seigh [Hennessy89] introduced passive serialization, which is an RCU-like mechanism that relies on the presence of "quiescent states" in the VM/XA hypervisor that are guaranteed not to be referencing the data structure. However, this mechanism was not optimized for modern computer systems, which is not surprising given that these overheads were not so expensive in the mid-80s. Nonetheless, passive serialization appears to be the first deferred-destruction mechanism to be used in production. Furthermore, the relevant patent has lapsed, so this approach may be used in non-GPL software, if desired. (In contrast, implementation of RCU is permitted only in software licensed under either GPL or LGPL. Sorry!!!)

In 1990, Pugh [Pugh90] noted that explicitly tracking which threads were reading a given data structure permitted deferred free to operate in the presence of non-terminating threads. However, this explicit tracking imposes significant read-side overhead, which is undesirable in read-mostly situations. This algorithm does take pains to avoid write-side contention and parallelize the other write-side overheads by providing a fine-grained locking design, however, it would be interesting to see how much of the performance advantage reported in 1990 remains in 2004.

At about this same time, Adams [Adams91] described ``chaotic relaxation'', where the normal barriers between successive iterations of convergent numerical algorithms are relaxed, so that iteration n might use data from iteration $n-1$ or even $n-2$. This introduces error, which typically slows convergence and thus increases the number of iterations required. However, this increase is sometimes more than made up for by a reduction in the number of expensive barrier operations, which are otherwise required to synchronize the threads at the end of each iteration. Unfortunately, chaotic relaxation requires highly structured data, such as the matrices used in scientific programs, and is thus inapplicable to most data structures in operating-system kernels.

In 1992, Henry (now Alexia) Massalin completed a dissertation advising parallel programmers to defer processing when feasible to simplify

synchronization. RCU makes extremely heavy use of this advice.

In 1993, Jacobson [Jacobson93] verbally described what is perhaps the simplest deferred-free technique: simply waiting a fixed amount of time before freeing blocks awaiting deferred free. Jacobson did not describe any write-side changes he might have made in this work using SGI's Irix kernel. Aju John published a similar technique in 1995 [AjuJohn95]. This works well if there is a well-defined upper bound on the length of time that reading threads can hold references, as there might well be in hard real-time systems. However, if this time is exceeded, perhaps due to preemption, excessive interrupts, or larger-than-anticipated load, memory corruption can ensue, with no reasonable means of diagnosis. Jacobson's technique is therefore inappropriate for use in production operating-system kernels, except when such kernels can provide hard real-time response guarantees for all operations.

Also in 1995, Pu et al. [Pu95a] applied a technique similar to that of Pugh's read-side-tracking to permit replugging of algorithms within a commercial Unix operating system. However, this replugging permitted only a single reader at a time. The following year, this same group of researchers extended their technique to allow for multiple readers [Cowan96a]. Their approach requires memory barriers (and thus pipeline stalls), but reduces memory latency, contention, and locking overheads.

1995 also saw the first publication of DYNIX/ptx's RCU mechanism [Slingwine95], which was optimized for modern CPU architectures, and was successfully applied to a number of situations within the DYNIX/ptx kernel. The corresponding conference paper appeared in 1998 [McKenney98].

In 1999, the Tornado and K42 groups described their "generations" mechanism, which quite similar to RCU [Gamsa99]. These operating systems made pervasive use of RCU in place of "existence locks", which greatly simplifies locking hierarchies.

2001 saw the first RCU presentation involving Linux [McKenney01a] at OLS. The resulting abundance of RCU patches was presented the following year [McKenney02a], and use of RCU in dcache was first described that same year [Linder02a].

Also in 2002, Michael [Michael02b, Michael02a] presented "hazard-pointer" techniques that defer the destruction of data structures to simplify non-blocking synchronization (wait-free synchronization, lock-free synchronization, and obstruction-free synchronization are all examples of non-blocking synchronization). In particular, this technique eliminates locking, reduces contention, reduces memory latency for readers, and parallelizes pipeline stalls and memory latency for writers. However, these techniques still impose significant read-side overhead in the form of memory barriers. Researchers at Sun worked along similar lines in the same timeframe [HerlihyLM02]. These techniques can be thought of as inside-out reference counts, where the count is represented by the number of hazard pointers referencing a given data structure (rather than the more conventional counter field within the data structure itself).

By the same token, RCU can be thought of as a "bulk reference count", where some form of reference counter covers all reference by a given CPU

or thread during a set timeframe. This timeframe is related to, but not necessarily exactly the same as, an RCU grace period. In classic RCU, the reference counter is the per-CPU bit in the "bitmask" field, and each such bit covers all references that might have been made by the corresponding CPU during the prior grace period. Of course, RCU can be thought of in other terms as well.

In 2003, the K42 group described how RCU could be used to create hot-pluggable implementations of operating-system functions [Appavoo03a]. Later that year saw a paper describing an RCU implementation of System V IPC [Arcangeli03], and an introduction to RCU in Linux Journal [McKenney03a].

2004 has seen a Linux-Journal article on use of RCU in dcache [McKenney04a], a performance comparison of locking to RCU on several different CPUs [McKenney04b], a dissertation describing use of RCU in a number of operating-system kernels [PaulEdwardMcKenneyPhD], a paper describing how to make RCU safe for soft-realtime applications [Sarma04c], and a paper describing SELinux performance with RCU [JamesMorris04b].

2005 brought further adaptation of RCU to realtime use, permitting preemption of RCU realtime critical sections [PaulMcKenney05a, PaulMcKenney05b].

2006 saw the first best-paper award for an RCU paper [ThomasEHart2006a], as well as further work on efficient implementations of preemptible RCU [PauleMcKenney2006b], but priority-boosting of RCU read-side critical sections proved elusive. An RCU implementation permitting general blocking in read-side critical sections appeared [PauleMcKenney2006c], Robert Olsson described an RCU-protected trie-hash combination [RobertOlsson2006a].

2007 saw the journal version of the award-winning RCU paper from 2006 [ThomasEHart2007a], as well as a paper demonstrating use of Promela and Spin to mechanically verify an optimization to Oleg Nesterov's QRCU [PauleMcKenney2007QRCUspin], a design document describing preemptible RCU [PauleMcKenney2007PreemptibleRCU], and the three-part LWN "What is RCU?" series [PauleMcKenney2007WhatIsRCUFundamentally, PauleMcKenney2008WhatIsRCUUsage, and PauleMcKenney2008WhatIsRCUAPI].

2008 saw a journal paper on real-time RCU [DinakarGuniguntala2008IBMSysJ], a history of how Linux changed RCU more than RCU changed Linux [PauleMcKenney2008RCUOSR], and a design overview of hierarchical RCU [PauleMcKenney2008HierarchicalRCU].

2009 introduced user-level RCU algorithms [PauleMcKenney2009MaliciousURCU], which Mathieu Desnoyers is now maintaining [MathieuDesnoyers2009URCU] [MathieuDesnoyersPhD]. TINY_RCU [PauleMcKenney2009BloatWatchRCU] made its appearance, as did expedited RCU [PauleMcKenney2009expeditedRCU]. The problem of resizable RCU-protected hash tables may now be on a path to a solution [JoshTriplett2009RPHash].

Bibtex Entries

```
@article{Kung80
,author="H. T. Kung and Q. Lehman"
```

```
, title="Concurrent Maintenance of Binary Search Trees"
, Year="1980"
, Month="September"
, journal="ACM Transactions on Database Systems"
, volume="5"
, number="3"
, pages="354-382"
}
```

```
@techreport {Manber82
, author="Udi Manber and Richard E. Ladner"
, title="Concurrency Control in a Dynamic Search Structure"
, institution="Department of Computer Science, University of Washington"
, address="Seattle, Washington"
, year="1982"
, number="82-01-01"
, month="January"
, pages="28"
}
```

```
@article {Manber84
, author="Udi Manber and Richard E. Ladner"
, title="Concurrency Control in a Dynamic Search Structure"
, Year="1984"
, Month="September"
, journal="ACM Transactions on Database Systems"
, volume="9"
, number="3"
, pages="439-455"
}
```

```
@techreport {Hennessy89
, author="James P. Hennessy and Damian L. Osisek and Joseph W. {Seigh II}"
, title="Passive Serialization in a Multitasking Environment"
, institution="US Patent and Trademark Office"
, address="Washington, DC"
, year="1989"
, number="US Patent 4,809,168 (lapsed)"
, month="February"
, pages="11"
}
```

```
@techreport {Pugh90
, author="William Pugh"
, title="Concurrent Maintenance of Skip Lists"
, institution="Institute of Advanced Computer Science Studies, Department of
Computer Science, University of Maryland"
, address="College Park, Maryland"
, year="1990"
, number="CS-TR-2222.1"
, month="June"
}
```

```
@Book {Adams91
, Author="Gregory R. Adams"
, title="Concurrent Programming, Principles, and Practices"
```

```
, Publisher="Benjamin Cummins"
, Year="1991"
}
```

```
@phdthesis {HMassalinPhD
, author="H. Massalin"
, title="Synthesis: An Efficient Implementation of Fundamental Operating
System Services"
, school="Columbia University"
, address="New York, NY"
, year="1992"
, annotation="
    Mondo optimizing compiler.
    Wait-free stuff.
    Good advice: defer work to avoid synchronization.
"
}
```

```
@unpublished {Jacobson93
, author="Van Jacobson"
, title="Avoid Read-Side Locking Via Delayed Free"
, year="1993"
, month="September"
, note="Verbal discussion"
}
```

```
@Conference {AjuJohn95
, Author="Aju John"
, Title="Dynamic vnodes -- Design and Implementation"
, Booktitle="{USENIX Winter 1995}"
, Publisher="USENIX Association"
, Month="January"
, Year="1995"
, pages="11-23"
, Address="New Orleans, LA"
}
```

```
@conference {Pu95a,
Author = "Calton Pu and Tito Autrey and Andrew Black and Charles Consel and
Crispin Cowan and Jon Inouye and Lakshmi Kethana and Jonathan Walpole and
Ke Zhang",
Title = "Optimistic Incremental Specialization: Streamlining a Commercial
Operating System",
Booktitle = "15\textsuperscript{th} ACM Symposium on
Operating Systems Principles (SOSP'95)",
address = "Copper Mountain, CO",
month="December",
year="1995",
pages="314-321",
annotation="
    Uses a replugger, but with a flag to signal when people are
    using the resource at hand. Only one reader at a time.
"
}
```

```
@conference {Cowan96a,
```

```

Author = "Crispin Cowan and Tito Autrey and Charles Krasic and
Calton Pu and Jonathan Walpole",
Title = "Fast Concurrent Dynamic Linking for an Adaptive Operating System",
Booktitle = "International Conference on Configurable Distributed Systems
(ICCDs' 96)",
address = "Annapolis, MD",
month="May",
year="1996",
pages="108",
isbn="0-8186-7395-8",
annotation="

```

```

    Uses a replugger, but with a counter to signal when people are
    using the resource at hand.  Allows multiple readers.
"

```

```

}

```

```

@techreport {Slingwine95
,author="John D. Slingwine and Paul E. McKenney"
,title="Apparatus and Method for Achieving Reduced Overhead Mutual
Exclusion and Maintaining Coherency in a Multiprocessor System
Utilizing Execution History and Thread Monitoring"
,institution="US Patent and Trademark Office"
,address="Washington, DC"
,year="1995"
,number="US Patent 5,442,758 (contributed under GPL)"
,month="August"
}

```

```

@techreport {Slingwine97
,author="John D. Slingwine and Paul E. McKenney"
,title="Method for maintaining data coherency using thread
activity summaries in a multicomputer system"
,institution="US Patent and Trademark Office"
,address="Washington, DC"
,year="1997"
,number="US Patent 5,608,893 (contributed under GPL)"
,month="March"
}

```

```

@techreport {Slingwine98
,author="John D. Slingwine and Paul E. McKenney"
,title="Apparatus and method for achieving reduced overhead
mutual exclusion and maintaining coherency in a multiprocessor
system utilizing execution history and thread monitoring"
,institution="US Patent and Trademark Office"
,address="Washington, DC"
,year="1998"
,number="US Patent 5,727,209 (contributed under GPL)"
,month="March"
}

```

```

@Conference {McKenney98
,Author="Paul E. McKenney and John D. Slingwine"
,Title="Read-Copy Update: Using Execution History to Solve Concurrency
Problems"
,Booktitle="{Parallel and Distributed Computing and Systems}"

```

```
,Month="October"
,Year="1998"
,pages="509-518"
,Address="Las Vegas, NV"
}
```

```
@Conference{Gamsa99
,Author="Ben Gamsa and Orran Krieger and Jonathan Appavoo and Michael Stumm"
,Title="Tornado: Maximizing Locality and Concurrency in a Shared Memory
Multiprocessor Operating System"
,Booktitle="{Proceedings of the 3\textsuperscript{rd} Symposium on
Operating System Design and Implementation}"
,Month="February"
,Year="1999"
,pages="87-100"
,Address="New Orleans, LA"
}
```

```
@techreport{Slingwine01
,author="John D. Slingwine and Paul E. McKenney"
,title="Apparatus and method for achieving reduced overhead
mutual exclusion and maintaining coherency in a multiprocessor
system utilizing execution history and thread monitoring"
,institution="US Patent and Trademark Office"
,address="Washington, DC"
,year="2001"
,number="US Patent 5,219,690 (contributed under GPL)"
,month="April"
}
```

```
@Conference{McKenney01a
,Author="Paul E. McKenney and Jonathan Appavoo and Andi Kleen and
Orran Krieger and Rusty Russell and Dipankar Sarma and Maneesh Soni"
,Title="Read-Copy Update"
,Booktitle="{Ottawa Linux Symposium}"
,Month="July"
,Year="2001"
,note="Available:
\url{http://www.linuxsymposium.org/2001/abstracts/readcopy.php}
\url{http://www.rdrop.com/users/paulmck/rclock/rclock_OLS.2001.05.01c.pdf}
[Viewed June 23, 2004]"
annotation="
Described RCU, and presented some patches implementing and using it in
the Linux kernel.
"
}
```

```
@Conference{Linder02a
,Author="Hanna Linder and Dipankar Sarma and Maneesh Soni"
,Title="Scalability of the Directory Entry Cache"
,Booktitle="{Ottawa Linux Symposium}"
,Month="June"
,Year="2002"
,pages="289-300"
}
```

```

@Conference{McKenney02a
, Author="Paul E. McKenney and Dipankar Sarma and
Andrea Arcangeli and Andi Kleen and Orran Krieger and Rusty Russell"
, Title="Read-Copy Update"
, Booktitle="{Ottawa Linux Symposium}"
, Month="June"
, Year="2002"
, pages="338-367"
, note="Available:
\url{http://www.linux.org.uk/~ajh/ols2002_proceedings.pdf.gz}
[Viewed June 23, 2004]"
}

@conference{Michael02a
, author="Maged M. Michael"
, title="Safe Memory Reclamation for Dynamic Lock-Free Objects Using Atomic
Reads and Writes"
, Year="2002"
, Month="August"
, booktitle="{Proceedings of the 21\textsuperscript{st} Annual ACM
Symposium on Principles of Distributed Computing}"
, pages="21-30"
, annotation="
      Each thread keeps an array of pointers to items that it is
      currently referencing. Sort of an inside-out garbage collection
      mechanism, but one that requires the accessing code to explicitly
      state its needs. Also requires read-side memory barriers on
      most architectures.
"
}

@conference{Michael02b
, author="Maged M. Michael"
, title="High Performance Dynamic Lock-Free Hash Tables and List-Based Sets"
, Year="2002"
, Month="August"
, booktitle="{Proceedings of the 14\textsuperscript{th} Annual ACM
Symposium on Parallel
Algorithms and Architecture}"
, pages="73-82"
, annotation="
      Like the title says...
"
}

@InProceedings{HerlihyLM02
, author={Maurice Herlihy and Victor Luchangco and Mark Moir}
, title="The Repeat Offender Problem: A Mechanism for Supporting Dynamic-Sized,
Lock-Free Data Structures"
, booktitle={Proceedings of 16\textsuperscript{th} International
Symposium on Distributed Computing}
, year=2002
, month="October"
, pages="339-353"
}

```



```
@article{Appavoo03a
,author="J. Appavoo and K. Hui and C. A. N. Soules and R. W. Wisniewski and
D. M. {Da Silva} and O. Krieger and M. A. Auslander and D. J. Edelsohn and
B. Gamsa and G. R. Ganger and P. McKenney and M. Ostrowski and
B. Rosenburg and M. Stumm and J. Xenidis"
,title="Enabling Autonomic Behavior in Systems Software With Hot Swapping"
,Year="2003"
,Month="January"
,journal="IBM Systems Journal"
,volume="42"
,number="1"
,pages="60-76"
}
```

```
@Conference{Arcangeli03
,Author="Andrea Arcangeli and Mingming Cao and Paul E. McKenney and
Dipankar Sarma"
,Title="Using Read-Copy Update Techniques for {System V IPC} in the
{Linux} 2.5 Kernel"
,Booktitle="Proceedings of the 2003 USENIX Annual Technical Conference
(FREENIX Track)"
,Publisher="USENIX Association"
,year="2003"
,month="June"
,pages="297-310"
}
```

```
@article{McKenney03a
,author="Paul E. McKenney"
,title="Using {RCU} in the {Linux} 2.5 Kernel"
,Year="2003"
,Month="October"
,journal="Linux Journal"
,volume="1"
,number="114"
,pages="18-26"
}
```

```
@techreport{Friedberg03a
,author="Stuart A. Friedberg"
,title="Lock-Free Wild Card Search Data Structure and Method"
,institution="US Patent and Trademark Office"
,address="Washington, DC"
,year="2003"
,number="US Patent 6,662,184 (contributed under GPL)"
,month="December"
,pages="112"
}
```

```
@article{McKenney04a
,author="Paul E. McKenney and Dipankar Sarma and Maneesh Soni"
,title="Scaling dcache with {RCU}"
,Year="2004"
,Month="January"
,journal="Linux Journal"
,volume="1"
```

```
, number="118"
, pages="38-46"
}
```

```
@Conference {McKenney04b
, Author="Paul E. McKenney"
, Title="{RCU} vs. Locking Performance on Different {CPUs}"
, Booktitle="{linux.conf.au}"
, Month="January"
, Year="2004"
, Address="Adelaide, Australia"
, note="Available:
\url{http://www.linux.org.au/conf/2004/abstracts.html#90}
\url{http://www.rdrop.com/users/paulmck/rclock/lockperf.2004.01.17a.pdf}
[Viewed June 23, 2004]"
}
```

```
@phdthesis {PaulEdwardMcKenneyPhD
, author="Paul E. McKenney"
, title="Exploiting Deferred Destruction:
An Analysis of Read-Copy-Update Techniques
in Operating System Kernels"
, school="OGI School of Science and Engineering at
Oregon Health and Sciences University"
, year="2004"
, note="Available:
\url{http://www.rdrop.com/users/paulmck/RCU/RCUdissertation.2004.07.14e1.pdf}
[Viewed October 15, 2004]"
}
```

```
@Conference {Sarma04c
, Author="Dipankar Sarma and Paul E. McKenney"
, Title="Making RCU Safe for Deep Sub-Millisecond Response Realtime Applications"
, Booktitle="Proceedings of the 2004 USENIX Annual Technical Conference
(FREENIX Track)"
, Publisher="USENIX Association"
, year="2004"
, month="June"
, pages="182-191"
}
```

```
@unpublished {JamesMorris04b
, Author="James Morris"
, Title="Recent Developments in {SELinux} Kernel Performance"
, month="December"
, year="2004"
, note="Available:
\url{http://www.livejournal.com/users/james_morris/2153.html}
[Viewed December 10, 2004]"
}
```

```
@unpublished {PaulMcKenney05a
, Author="Paul E. McKenney"
, Title="{[RFC]} {RCU} and {CONFIG\_PREEMPT\_RT} progress"
, month="May"
, year="2005"
```

```
,note="Available:
\url{http://lkml.org/lkml/2005/5/9/185}
[Viewed May 13, 2005]"
,annotation="
    First publication of working lock-based deferred free patches
    for the CONFIG_PREEMPT_RT environment.
"
}

@conference{PaulMcKenney05b
,Author="Paul E. McKenney and Dipankar Sarma"
,Title="Towards Hard Realtime Response from the Linux Kernel on SMP Hardware"
,Booktitle="linux.conf.au 2005"
,month="April"
,year="2005"
,address="Canberra, Australia"
,note="Available:
\url{http://www.rdrop.com/users/paulmck/RCU/realtimeRCU.2005.04.23a.pdf}
[Viewed May 13, 2005]"
,annotation="
    Realtime turns into making RCU yet more realtime friendly.
"
}

@conference{ThomasEHart2006a
,Author="Thomas E. Hart and Paul E. McKenney and Angela Demke Brown"
,Title="Making Lockless Synchronization Fast: Performance Implications
of Memory Reclamation"
,Booktitle="20th {IEEE} International Parallel and
Distributed Processing Symposium"
,month="April"
,year="2006"
,day="25-29"
,address="Rhodes, Greece"
,annotation="
    Compares QSBR (AKA "classic RCU"), HPBR, EBR, and lock-free
    reference counting.
"
}

@Conference{PaulEMcKenney2006b
,Author="Paul E. McKenney and Dipankar Sarma and Ingo Molnar and
Suparna Bhattacharya"
,Title="Extending RCU for Realtime and Embedded Workloads"
,Booktitle="{Ottawa Linux Symposium}"
,Month="July"
,Year="2006"
,pages="v2 123-138"
,note="Available:
\url{http://www.linuxsymposium.org/2006/view_abstract.php?content_key=184}
\url{http://www.rdrop.com/users/paulmck/RCU/OLSrtRCU.2006.08.11a.pdf}
[Viewed January 1, 2007]"
,annotation="
    Described how to improve the -rt implementation of realtime RCU.
"
}
```

```
@unpublished{PauleMcKenney2006c
, Author="Paul E. McKenney"
, Title="Sleepable {RCU}"
, month="October"
, day="9"
, year="2006"
, note="Available:
\url{http://lwn.net/Articles/202847/}
Revised:
\url{http://www.rdrop.com/users/paulmck/RCU/srcu.2007.01.14a.pdf}
[Viewed August 21, 2006]"
, annotation="
    LWN article introducing SRCU.
"
}

@unpublished{RobertOlsson2006a
, Author="Robert Olsson and Stefan Nilsson"
, Title="{TRASH}: A dynamic {LC}-trie and hash data structure"
, month="August"
, day="18"
, year="2006"
, note="Available:
\url{http://www.nada.kth.se/~snilsson/public/papers/trash/trash.pdf}
[Viewed February 24, 2007]"
, annotation="
    RCU-protected dynamic trie-hash combination.
"
}

@unpublished{ThomasEHart2007a
, Author="Thomas E. Hart and Paul E. McKenney and Angela Demke Brown and Jonathan
Walpole"
, Title="Performance of memory reclamation for lockless synchronization"
, journal="J. Parallel Distrib. Comput."
, year="2007"
, note="To appear in J. Parallel Distrib. Comput.
\url{doi=10.1016/j.jpdc.2007.04.010}"
, annotation={
    Compares QSBR (AKA "classic RCU"), HPBR, EBR, and lock-free
    reference counting. Journal version of ThomasEHart2006a.
}
}

@unpublished{PauleMcKenney2007QRCUspin
, Author="Paul E. McKenney"
, Title="Using Promela and Spin to verify parallel algorithms"
, month="August"
, day="1"
, year="2007"
, note="Available:
\url{http://lwn.net/Articles/243851/}
[Viewed September 8, 2007]"
, annotation="
    LWN article describing Promela and spin, and also using Oleg
```

Nesterov's QRCU as an example (with Paul McKenney's fastpath).

”
}

```
@unpublished{PauleMcKenney2007PreemptibleRCU
,Author="Paul E. McKenney"
,Title="The design of preemptible read-copy-update"
,month="October"
,day="8"
,year="2007"
,note="Available:
\url{http://lwn.net/Articles/253651/}
[Viewed October 25, 2007]"
,annotation="
```

LWN article describing the design of preemptible RCU.

”
}

```
#####
#
# "What is RCU?" LWN series.
#
```

```
@unpublished{PauleMcKenney2007WhatIsRCUFundamentally
,Author="Paul E. McKenney and Jonathan Walpole"
,Title="What is {RCU}, Fundamentally?"
,month="December"
,day="17"
,year="2007"
,note="Available:
\url{http://lwn.net/Articles/262464/}
[Viewed December 27, 2007]"
,annotation="
```

Lays out the three basic components of RCU: (1) publish-subscribe,
(2) wait for pre-existing readers to complete, and (2) maintain
multiple versions.

”
}

```
@unpublished{PauleMcKenney2008WhatIsRCUUsage
,Author="Paul E. McKenney"
,Title="What is {RCU}? Part 2: Usage"
,month="January"
,day="4"
,year="2008"
,note="Available:
\url{http://lwn.net/Articles/263130/}
[Viewed January 4, 2008]"
,annotation="
```

Lays out six uses of RCU:

1. RCU is a Reader-Writer Lock Replacement
2. RCU is a Restricted Reference-Counting Mechanism
3. RCU is a Bulk Reference-Counting Mechanism
4. RCU is a Poor Man's Garbage Collector
5. RCU is a Way of Providing Existence Guarantees
6. RCU is a Way of Waiting for Things to Finish

```
"
}
```

```
@unpublished{PauleMcKenney2008WhatIsRCUAPI
, Author="Paul E. McKenney"
, Title="{RCU} part 3: the {RCU} {API}"
, month="January"
, day="17"
, year="2008"
, note="Available:
\url{http://lwn.net/Articles/264090/}
[Viewed January 10, 2008]"
, annotation="
    Gives an overview of the Linux-kernel RCU API and a brief annotated RCU
    bibliography.
"
}
```

```
#
# "What is RCU?" LWN series.
#
#####
```

```
@article{DinakarGuniguntala2008IBMSysJ
, author="D. Guniguntala and P. E. McKenney and J. Triplett and J. Walpole"
, title="The read-copy-update mechanism for supporting real-time applications on
shared-memory multiprocessor systems with {Linux}"
, Year="2008"
, Month="April"
, journal="IBM Systems Journal"
, volume="47"
, number="2"
, pages="@-@"
, annotation="
    RCU, realtime RCU, sleepable RCU, performance.
"
}
```

```
@article{PauleMcKenney2008RCUOSR
, author="Paul E. McKenney and Jonathan Walpole"
, title="Introducing technology into the {Linux} kernel: a case study"
, Year="2008"
, journal="SIGOPS Oper. Syst. Rev."
, volume="42"
, number="5"
, pages="4--17"
, issn="0163-5980"
, doi={http://doi.acm.org/10.1145/1400097.1400099}
, publisher="ACM"
, address="New York, NY, USA"
, annotation={
    Linux changed RCU to a far greater degree than RCU has changed Linux.
}
}
```

```
@unpublished{PauleMcKenney2008HierarchicalRCU
```

```
,Author="Paul E. McKenney"
,Title="Hierarchical {RCU}"
,month="November"
,day="3"
,year="2008"
,note="Available:
\url{http://lwn.net/Articles/305782/}
[Viewed November 6, 2008]"
,annotation="
    RCU with combining-tree-based grace-period detection,
    permitting it to handle thousands of CPUs.
"
}

@conference{PaulEMcKenney2009MaliciousURCU
,Author="Paul E. McKenney"
,Title="Using a Malicious User-Level {RCU} to Torture {RCU}-Based Algorithms"
,Booktitle="linux.conf.au 2009"
,month="January"
,year="2009"
,address="Hobart, Australia"
,note="Available:
\url{http://www.rdrop.com/users/paulmck/RCU/urcutorture.2009.01.22a.pdf}
[Viewed February 2, 2009]"
,annotation="
    Realtime RCU and torture-testing RCU uses.
"
}

@unpublished{MathieuDesnoyers2009URCU
,Author="Mathieu Desnoyers"
,Title="[{RFC} git tree] Userspace {RCU} (urcu) for {Linux}"
,month="February"
,day="5"
,year="2009"
,note="Available:
\url{http://lkml.org/lkml/2009/2/5/572}
\url{git://lttng.org/userspace-rcu.git}
[Viewed February 20, 2009]"
,annotation="
    Mathieu Desnoyers's user-space RCU implementation.
    git://lttng.org/userspace-rcu.git
"
}

@unpublished{PaulEMcKenney2009BloatWatchRCU
,Author="Paul E. McKenney"
,Title="{RCU}: The {Bloatwatch} Edition"
,month="March"
,day="17"
,year="2009"
,note="Available:
\url{http://lwn.net/Articles/323929/}
[Viewed March 20, 2009]"
,annotation="
    Uniprocessor assumptions allow simplified RCU implementation.

```

```
"
}
```

```
@unpublished{PaulEMcKenney2009expeditedRCU
, Author="Paul E. McKenney"
, Title="[PATCH] -tip 0/3] expedited 'big hammer' {RCU} grace periods"
, month="June"
, day="25"
, year="2009"
, note="Available:
\url{http://lkml.org/lkml/2009/6/25/306}
[Viewed August 16, 2009]"
, annotation="
    First posting of expedited RCU to be accepted into -tip.
"
}
```

```
@unpublished{JoshTriplett2009RPHash
, Author="Josh Triplett"
, Title="Scalable concurrent hash tables via relativistic programming"
, month="September"
, year="2009"
, note="Linux Plumbers Conference presentation"
, annotation="
    RP fun with hash tables.
"
}
```

```
@phdthesis{MathieuDesnoyersPhD
, title = "Low-Impact Operating System Tracing"
, author = "Mathieu Desnoyers"
, school = "Ecole Polytechnique de Montr\'{e}al"
, month = "December"
, year = 2009
, note="Available:
\url{http://www.lttng.org/pub/thesis/desnoyers-dissertation-2009-12.pdf}
[Viewed December 9, 2009]"
}
```