



Mastodon
@nelson@tech.lgbt

Linkblog

Thu 2023-01-26
[Flent](#)
[Carbon offset sham](#)

Tue 2023-01-24
[Google ads antitrust](#)
[Mayor of Slow Sanchez](#)
[irtt: Isochronous Round-Trip Tester](#)

Mon 2023-01-23
[mozilla.org's 25th anniversary](#)
[EV across the Nullarbor](#)

Sun 2023-01-22
[Harvey's closes](#)
[Huge Pages are a Good Idea](#)

Sat 2023-01-21
[HC-Tree \(sqlite\)](#)
[Wikipedia redesign](#)

Thu 2023-01-19
[Kerrygold butter ban](#)

Tue 2023-01-17
[Twitter API key workaround](#)
[Inside Elon Musk's "extremely hardcore" Twitter](#)
[Ham Stand new owners](#)

Mon 2023-01-16
[Python packaging](#)

Sun 2023-01-15
[Berliner Schnauze](#)
[sqllean](#)

Sat 2023-01-14
[USGS Mars Map](#)
[London: Pavement Oddities](#)

Search

Leap Second crashes half the internet

Yesterday's leap second [killed half the Internet](#), including [Pirate Bay](#), [Reddit](#), [LinkedIn](#), [Gawker Media](#) and a [host of other sites](#). Even [an airline](#). Any Linux user processes that depends on kernel threads had a high chance of failing. That includes MySQL and many Java servers like webapps, Hadoop, Cassandra, etc. The symptom was the user process spinning at 100% CPU even after being restarted. A quick fix seems to be [setting the system clock](#) which apparently resets the bad state in the kernel (we hope).



The underlying cause is something about how the kernel handled the extra second broke the futex locks used by threaded processes. Here's a [very detailed analysis on the failing code](#) but I'm not sure it's correct. According to [this analysis](#) the bug was [introduced in 2008](#), then [fixed in March 2012](#). But it may be the March fix [is part of the problem](#). OTOH most of the systems that failed will be running kernels older than March so the problem must go further back. [There's a kernel fix](#) and also a [detailed analysis](#). [Time is hard](#), let's go shopping.

It's frustrating that these bugs keep popping up; the [theory](#) is [not so difficult](#). The NTP daemon tells the kernel a leap second is coming via `adjtime()`, the kernel should handle it by slewing or holding the clock, all is well. But it didn't work in 2012. Didn't work in 2009 either; a [logging bug](#) caused kernels to crash on the leap second. 2005 was better. [Google's solution](#) of giving up on the kernel entirely and having the NTP daemon lie about what time it is seems more clever now.

I got hit by this bug myself, the CrashPlan backup daemon runs Java and got caught in a spin. And none of my machines really kept time right because [POSIX does not](#) account for [leap seconds](#). Both Ubuntu boxes just ran 23:59:59 twice, so time went backwards on a subsecond basis. My Mac was even worse, it actually flipped over to 00:00:00 before going backwards to 23:59:59 briefly. ~~No doubt my GPS devices are off by a second now; most consumer devices have no facility to update the leap second database.~~ (Correction: GPS satellites broadcast the UTC offset.) The only thing that worked right was [NIST's clock widget](#) pictured above, showing 23:59:60.

[tech](#) • [bad](#)

10 years ago 2012-07-03 00:00 Z

[Nelson's Weblog](#) • [tech](#) • [bad](#)

Archives

2021

[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2020

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)

[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2019

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)

[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2018

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)

[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2017

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)

[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2016

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)

[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2015

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)

[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2014

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)

[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2013

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)

[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2012

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)

[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2011

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)

[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2010

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)

[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2009

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)

[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2008

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)

[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2007

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)

[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2006

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)
[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2005

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)
[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2004

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)
[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2003

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)
[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2002

[12](#) [11](#) [10](#) [09](#) [08](#) [07](#)
[06](#) [05](#) [04](#) [03](#) [02](#) [01](#)

2001

[12](#) [11](#) [10](#)



One good site

[MDN](#)



[Nelson Minar](#)

nelson@monkey.org



Blog licensed under a
[Creative Commons](#)
[License](#)

