Wikapedia – Futexes from https://en.wikipedia.org/wiki/Futex

Futex

From Wikipedia, the free encyclopedia

[Jump to navigation](https://en.wikipedia.org/wiki/Futex#mw-head)[Jump to search](https://en.wikipedia.org/wiki/Futex#searchInput)

In [computing](https://en.wikipedia.org/wiki/Computing), a **futex** (short for "fast userspace [mutex](https://en.wikipedia.org/wiki/Mutual_exclusion" \o "Mutual exclusion)") is a [kernel](https://en.wikipedia.org/wiki/Kernel_(operating_system)) [system call](https://en.wikipedia.org/wiki/System_call) that [programmers](https://en.wikipedia.org/wiki/Programmer) can use to implement basic [locking](https://en.wikipedia.org/wiki/Lock_(computer)), or as a building block for higher-level locking abstractions such as [semaphores](https://en.wikipedia.org/wiki/Semaphore_(programming)) and [POSIX](https://en.wikipedia.org/wiki/POSIX) mutexes or [condition variables](https://en.wikipedia.org/wiki/Condition_variable).

A futex consists of a [kernelspace](https://en.wikipedia.org/wiki/Kernel_(operating_system)" \o "Kernel (operating system)) *wait queue* that is attached to an [atomic](https://en.wikipedia.org/wiki/Atomic_operations) [integer](https://en.wikipedia.org/wiki/Integer) in [userspace](https://en.wikipedia.org/wiki/Userspace" \o "Userspace). Multiple [processes](https://en.wikipedia.org/wiki/Process_(computing)) or [threads](https://en.wikipedia.org/wiki/Thread_(computer_science)) operate on the integer entirely in userspace (using [atomic operations](https://en.wikipedia.org/wiki/Atomic_operation) to avoid interfering with one another), and only resort to relatively expensive[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)] [system calls](https://en.wikipedia.org/wiki/System_call) to request operations on the wait queue (for example to wake up waiting processes, or to put the current process on the wait queue). A properly programmed futex-based lock will not use system calls except when the lock is contended; since most operations do not require arbitration between processes, this will not happen in most cases.



**Contents**

* [1History](https://en.wikipedia.org/wiki/Futex#History)
* [2Operations](https://en.wikipedia.org/wiki/Futex#Operations)
* [3See also](https://en.wikipedia.org/wiki/Futex#See_also)
* [4References](https://en.wikipedia.org/wiki/Futex#References)
* [5External links](https://en.wikipedia.org/wiki/Futex#External_links)

History[[edit](https://en.wikipedia.org/w/index.php?title=Futex&action=edit&section=1" \o "Edit section: History)]

|  |  |
| --- | --- |
| [Wiki letter w.svg](https://en.wikipedia.org/wiki/File:Wiki_letter_w.svg) | This section **is missing information** about FUTEX2 by valve, mainly intended to mimic WaitForMultipleObjects in wine/proton “fsync”. Please expand the section to include this information. Further details may exist on the [talk page](https://en.wikipedia.org/wiki/Talk:Futex). *(November 2021)* |

On [Linux](https://en.wikipedia.org/wiki/Linux), Hubertus Franke ([IBM](https://en.wikipedia.org/wiki/IBM) [Thomas J. Watson Research Center](https://en.wikipedia.org/wiki/Thomas_J._Watson_Research_Center)), Matthew Kirkwood, [Ingo Molnár](https://en.wikipedia.org/wiki/Ingo_Moln%C3%A1r) ([Red Hat](https://en.wikipedia.org/wiki/Red_Hat)) and [Rusty Russell](https://en.wikipedia.org/wiki/Rusty_Russell) ([IBM Linux Technology Center](https://en.wikipedia.org/wiki/IBM_Linux_Technology_Center)) originated the futex mechanism. Futexes appeared for the first time in version 2.5.7 of the Linux kernel development series; the semantics stabilized as of version 2.5.40, and futexes have been part of the [Linux kernel mainline](https://en.wikipedia.org/wiki/Linux_kernel_mainline) since the December 2003 release of 2.6.x stable kernel series.

In 2002 discussions took place on a proposal to make futexes accessible via the file system by creating a special node in /dev or /proc. However, [Linus Torvalds](https://en.wikipedia.org/wiki/Linus_Torvalds) strongly opposed this idea and rejected any related patches.[[1]](https://en.wikipedia.org/wiki/Futex#cite_note-1)

Futexes have been implemented in Microsoft Windows since Windows 8 or Windows Server 2012 under the name WaitOnAddress.[[2]](https://en.wikipedia.org/wiki/Futex#cite_note-2)

In 2013 Microsoft patented futexes and the patent was granted in 2014.[[3]](https://en.wikipedia.org/wiki/Futex#cite_note-3)

In May 2014 the [CVE](https://en.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures) system announced a vulnerability discovered in the Linux kernel's futex subsystem that allowed denial-of-service attacks or local privilege escalation.[[4]](https://en.wikipedia.org/wiki/Futex#cite_note-4)[[5]](https://en.wikipedia.org/wiki/Futex#cite_note-5)

In May 2015 the [Linux kernel](https://en.wikipedia.org/wiki/Kernel_(operating_system)) introduced a deadlock bug via [Commit b0c29f79ecea](https://github.com/torvalds/linux/commit/b0c29f79ecea) that caused a hang in user applications. The bug affected many enterprise Linux distributions, including 3.x and 4.x kernels, and Red Hat Enterprise Linux version 5, 6 and 7, SUSE Linux 12 and Amazon Linux.[[6]](https://en.wikipedia.org/wiki/Futex#cite_note-6)

Futexes have been implemented in OpenBSD since 2016.[[7]](https://en.wikipedia.org/wiki/Futex#cite_note-7)

The futex mechanism is one of the core concepts of the Zircon kernel[[8]](https://en.wikipedia.org/wiki/Futex" \l "cite_note-8) in [Google](https://en.wikipedia.org/wiki/Google)'s [Fuchsia operating system](https://en.wikipedia.org/wiki/Google_Fuchsia) since at least April 2018.[[9]](https://en.wikipedia.org/wiki/Futex#cite_note-9)

Operations[[edit](https://en.wikipedia.org/w/index.php?title=Futex&action=edit&section=2" \o "Edit section: Operations)]

Futexes have two basic operations, WAIT and WAKE.

* WAIT(addr, val)

If the value stored at the address addr is val, puts the current thread to sleep.

* WAKE(addr, num)

Wakes up num number of threads waiting on the address addr.

For more advanced uses there are a number of other operations the most used being REQUEUE and WAKE\_OP, which both function as more generic WAKE operations.[[10]](https://en.wikipedia.org/wiki/Futex#cite_note-Ulrich_Drepper-10)

* CMP\_REQUEUE(old\_addr, new\_addr, num\_wake, num\_move, val)

If the value stored at the address old\_addr is val, wakes num\_wake threads waiting on the address old\_addr, and enqueues num\_move threads waiting on the address old\_addr to now wait on the address new\_addr. This can be used to avoid the [thundering herd problem](https://en.wikipedia.org/wiki/Thundering_herd_problem) on wake.[[11]](https://en.wikipedia.org/wiki/Futex#cite_note-11)[[12]](https://en.wikipedia.org/wiki/Futex#cite_note-12)

* WAKE\_OP(addr1, addr2, num1, num2, op, op\_arg, cmp, cmp\_arg)

Will read addr2, perform op with op\_arg on it, and storing the result back to addr2. Then it will wake num1 threads waiting on addr1, and if the previously read value from addr2 matches cmp\_arg using comparison cmp will wake num2 threads waiting on addr2. This very flexible and generic wake mechanism is useful for implementing many synchronization primitives.

See also[[edit](https://en.wikipedia.org/w/index.php?title=Futex&action=edit&section=3" \o "Edit section: See also)]

* [Synchronization](https://en.wikipedia.org/wiki/Synchronization_(computer_science))
* [Fetch-and-add](https://en.wikipedia.org/wiki/Fetch-and-add)
* [Compare and swap](https://en.wikipedia.org/wiki/Compare_and_swap)

References[[edit](https://en.wikipedia.org/w/index.php?title=Futex&action=edit&section=4" \o "Edit section: References)]

* 1. [**^**](https://en.wikipedia.org/wiki/Futex#cite_ref-1) *Torvalds, Linus.*[*"Futex Asynchronous Interface"*](http://yarchive.net/comp/linux/everything_is_file.html)*.*
  2. [**^**](https://en.wikipedia.org/wiki/Futex#cite_ref-2) [*"WaitOnAddress function"*](https://docs.microsoft.com/en-us/windows/win32/api/synchapi/nf-synchapi-waitonaddress)*. Retrieved 2019-11-01.*
  3. [**^**](https://en.wikipedia.org/wiki/Futex#cite_ref-3) [*"US8782674B2 Wait on address synchronization interface"*](https://patents.google.com/patent/US8782674B2/en)*. Retrieved 2019-11-01.*
  4. [**^**](https://en.wikipedia.org/wiki/Futex#cite_ref-4) [CVE-2014-3153](http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3153)
  5. [**^**](https://en.wikipedia.org/wiki/Futex#cite_ref-5) [*"[SECURITY] [DSA 2949-1] linux security update"*](https://lists.debian.org/debian-security-announce/2014/msg00130.html)*. Lists.debian.org. 2014-06-05. Retrieved 2014-06-08.*
  6. [**^**](https://en.wikipedia.org/wiki/Futex#cite_ref-6) [*"Linux futex\_wait() bug..."*](https://groups.google.com/forum/#!topic/mechanical-sympathy/QbmpZxp6C64)*2015-05-13. Retrieved 2018-03-24.*
  7. [**^**](https://en.wikipedia.org/wiki/Futex#cite_ref-7) *Mazurek, Michal.*[*"'Futexes for OpenBSD' - MARC"*](https://marc.info/?l=openbsd-tech&m=147282346309815&w=2)*. marc.info. Retrieved 30 April 2017.*
  8. [**^**](https://en.wikipedia.org/wiki/Futex#cite_ref-8) [*"Zircon Kernel Concepts"*](https://fuchsia.dev/fuchsia-src/zircon/concepts#futexes)*. fuchsia.dev. Retrieved 20 October 2019.*
  9. [**^**](https://en.wikipedia.org/wiki/Futex#cite_ref-9) [*"zx\_futex\_wait"*](https://fuchsia.dev/fuchsia-src/reference/syscalls/futex_wait)*. fuchsia.dev. Retrieved 20 October 2019.*
  10. [**^**](https://en.wikipedia.org/wiki/Futex#cite_ref-Ulrich_Drepper_10-0) [Futexes Are Tricky](https://www.akkadia.org/drepper/futex.pdf) Ulrich Drepper (Red Hat, v1.6, 2011)
  11. [**^**](https://en.wikipedia.org/wiki/Futex#cite_ref-11) [Linux futex(2) man page, FUTEX\_CMP\_REQUEUE section](http://man7.org/linux/man-pages/man2/futex.2.html)
  12. [**^**](https://en.wikipedia.org/wiki/Futex#cite_ref-12) [Zircon zx\_futex\_requeue documentation](https://fuchsia.dev/fuchsia-src/reference/syscalls/futex_requeue.md)

External links[[edit](https://en.wikipedia.org/w/index.php?title=Futex&action=edit&section=5)]

* [futex(2)](https://man7.org/linux/man-pages/man2/futex.2.html) - futex() system call
* [futex(7)](https://man7.org/linux/man-pages/man7/futex.7.html) - futex semantics and usage
* Hubertus Franke, Rusty Russell, Matthew Kirkwood. [*Fuss, futexes and furwocks: Fast Userlevel Locking in Linux*](https://www.kernel.org/doc/ols/2002/ols2002-pages-479-495.pdf), [Ottawa Linux Symposium](https://en.wikipedia.org/wiki/Ottawa_Linux_Symposium) 2002.
* *Drepper, Ulrich (2011).*[*"Futexes are Tricky"*](https://www.akkadia.org/drepper/futex.pdf)*(PDF). 1.6. Red Hat.*
* Bert Hubert (2004). [Unofficial Futex manpages](http://ds9a.nl/futex-manpages/)
* Ingo Molnar. "[Robust Futexes](https://www.kernel.org/doc/Documentation/robust-futexes.txt)", *Linux Kernel Documentation*
* "[Priority Inheritance Futexes](https://www.kernel.org/doc/Documentation/pi-futex.txt)", *Linux Kernel Documentation*