

Semaphores

Semaphores are synchronization mechanisms to control access to a shared resource.

A semaphore is initialized with a counter greater than 0

- Requesting the semaphore decrements the counter
 - Releasing the semaphore increments the counter
 - A requesting thread is blocked if the counter is 0
-
- C++20 support two semaphores
 - `std::counting_semaphore`
 - `std::binary_semaphore` (`std::counting_semaphore<1>`)

Semaphores

Member Function	Description
<code>counting_semaphore::max()</code>	Returns the maximum value of the <code>counter</code>
<code>sem.release(upd = 1)</code>	Atomically increases the counter by <code>upd</code> and unblocks threads acquiring the semaphore
<code>sem.acquire()</code>	Decrements counter by 1 or blocks until the <code>counter</code> is greater than 0
<code>sem.try_acquire()</code>	Tries to decrement the <code>counter</code> by 1 if it is greater than 0
<code>sem.try_acquire_for(relT)</code>	Tries to decrement the <code>counter</code> by 1 or blocks for at most <code>relT</code> if <code>counter</code> is 0
<code>sem.try_acquire_until(absT)</code>	Tries to decrement the <code>counter</code> by 1 or blocks at most until <code>absT</code> if <code>counter</code> is 0