

Package ‘semaphore’

March 18, 2025

Type Package

Title Shared Memory Atomic Operations

Version 1.1.0

Date 2025-03-17

Description Implements named semaphores from the 'boost' 'C++' library <<https://www.boost.org/>>. A semaphore object is shared amongst several processes. This integer value can be safely incremented or decremented by each process. Processes can also wait (blocking) for the value to become positive.

URL <https://cmmr.github.io/semaphore/>,
<https://github.com/cmmr/semaphore>

BugReports <https://github.com/cmmr/semaphore/issues>

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Config/testthat/edition 3

LinkingTo Rcpp, BH

Imports Rcpp

Suggests testthat

NeedsCompilation yes

Author Daniel P. Smith [aut, cre] (<<https://orcid.org/0000-0002-2479-2044>>),
Alkek Center for Metagenomics and Microbiome Research [cph, fnd]

Maintainer Daniel P. Smith <dansmith01@gmail.com>

Repository CRAN

Date/Publication 2025-03-18 11:30:08 UTC

Contents

create_semaphore	2
Index	4

create_semaphore *Shared Memory Atomic Operations*

Description

A semaphore is an integer that the operating system keeps track of. Any process that knows the semaphore's identifier can increment or decrement its value, though it cannot be decremented below zero.

When the semaphore is zero, calling `decrement_semaphore(wait = FALSE)` will return `FALSE` whereas `decrement_semaphore(wait = TRUE)` will block until the semaphore is incremented by another process. If multiple processes are blocked, a single call to `increment_semaphore()` will only unblock one of the blocked processes.

It is possible to wait for a specific amount of time, for example, `decrement_semaphore(wait = 10)` will wait for 10 seconds. If the semaphore is incremented within those 10 seconds, the function will immediately return `TRUE`. Otherwise it will return `FALSE` at the 10 second mark.

Usage

```
create_semaphore(id = NULL, value = 0, cleanup = TRUE)
```

```
increment_semaphore(id)
```

```
decrement_semaphore(id, wait = TRUE)
```

```
remove_semaphore(id)
```

Arguments

<code>id</code>	A semaphore identifier (string). <code>create_semaphore()</code> defaults to generating a random identifier. A custom id should be at most 251 characters and must not contain slashes (/).
<code>value</code>	The initial value of the semaphore.
<code>cleanup</code>	Remove the semaphore when R session exits.
<code>wait</code>	Whether/how long to wait for the semaphore: <ul style="list-style-type: none"> • <code>FALSE</code>: return immediately. • <code>TRUE</code>: block until semaphore available. • integer: this many seconds at most.

Value

- `create_semaphore()` - The created semaphore's identifier (string), invisibly when semaphore is non-NULL.
- `increment_semaphore()` - `TRUE`, invisibly.

- `decrement_semaphore()` - TRUE if the decrement was successful; FALSE otherwise, invisibly when `wait=TRUE`.
- `remove_semaphore()` - TRUE on success; FALSE on error.

Examples

```
library(semaphore)

s <- create_semaphore()
print(s)

increment_semaphore(s)
decrement_semaphore(s, wait = FALSE)
decrement_semaphore(s, wait = FALSE)

remove_semaphore(s)
```

Index

`create_semaphore`, [2](#)

`decrement_semaphore (create_semaphore)`,
[2](#)

`increment_semaphore (create_semaphore)`,
[2](#)

`remove_semaphore (create_semaphore)`, [2](#)