Package 'semaphore'

March 29, 2025

Type Package

Title Shared Memory Atomic Operations

Version 1.2.0

Date 2025-03-28

Description Implements named semaphores from the 'boost' 'C++' library <https://www.boost.org/> for interprocess communication. Multiple 'R' sessions on the same host can block (with optional timeout) on a semaphore until it becomes positive, then atomically decrement it and unblock. Any session can increment the semaphore.

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 2025-03-29 00:30:02 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

`id`	A semaphore identifier (string). `create_semaphore()` defaults to generating a random identifier. A custom id should be at most 251 characters and must not contain slashes (`/`).
`value`	The initial value of the semaphore.
`cleanup`	Remove the semaphore when R session exits.
`wait`	Maximum time (in seconds) to block the process while waiting for the semaphore. `TRUE` maps to `0`; `FALSE` maps to `Inf`. Fractional seconds allowed (e.g. `wait=0.001`).

Value

- `create_semaphore()` - The created semaphore's identifier (string), invisibly unless `id=NULL`.
- `increment_semaphore()` - `TRUE` on success or `FALSE` on error, invisibly.
- `decrement_semaphore()` - `TRUE` if the decrement was successful or `FALSE` otherwise, invisibly when `wait=Inf`.
- `remove_semaphore()` - `TRUE` on success or `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