NAME

Thread::Semaphore - thread-safe semaphores

SYNOPSIS

```
use Thread::Semaphore;
my $s = new Thread::Semaphore;
$s->down; # Also known as the semaphore P operation.
# The guarded section is here
$s->up; # Also known as the semaphore V operation.

# The default semaphore value is 1.
my $s = new Thread::Semaphore($initial_value);
$s->down($down_value);
$s->up($up_value);
```

DESCRIPTION

Semaphores provide a mechanism to regulate access to resources. Semaphores, unlike locks, aren't tied to particular scalars, and so may be used to control access to anything you care to use them for.

Semaphores don't limit their values to zero or one, so they can be used to control access to some resource that there may be more than one of. (For example, filehandles.) Increment and decrement amounts aren't fixed at one either, so threads can reserve or return multiple resources at once.

FUNCTIONS AND METHODS

new

new NUMBER

new creates a new semaphore, and initializes its count to the passed number. If no number is passed, the semaphore's count is set to one.

down

down NUMBER

The down method decreases the semaphore's count by the specified number, or by one if no number has been specified. If the semaphore's count would drop below zero, this method will block until such time that the semaphore's count is equal to or larger than the amount you're downing the semaphore's count by.

This is the semaphore "P operation" (the name derives from the Dutch word "pak", which means "capture" -- the semaphore operations were named by the late Dijkstra, who was Dutch).

up

up NUMBER

The up method increases the semaphore's count by the number specified, or by one if no number has been specified. This will unblock any thread blocked trying to down the semaphore if the up raises the semaphore count above the amount that the downs are trying to decrement it by.

This is the semaphore "V operation" (the name derives from the Dutch word "vrij", which means "release").