Introducing

Concurrency

Introduction

raywenderlich.com

Concurrency is HARD

raywenderlich.com

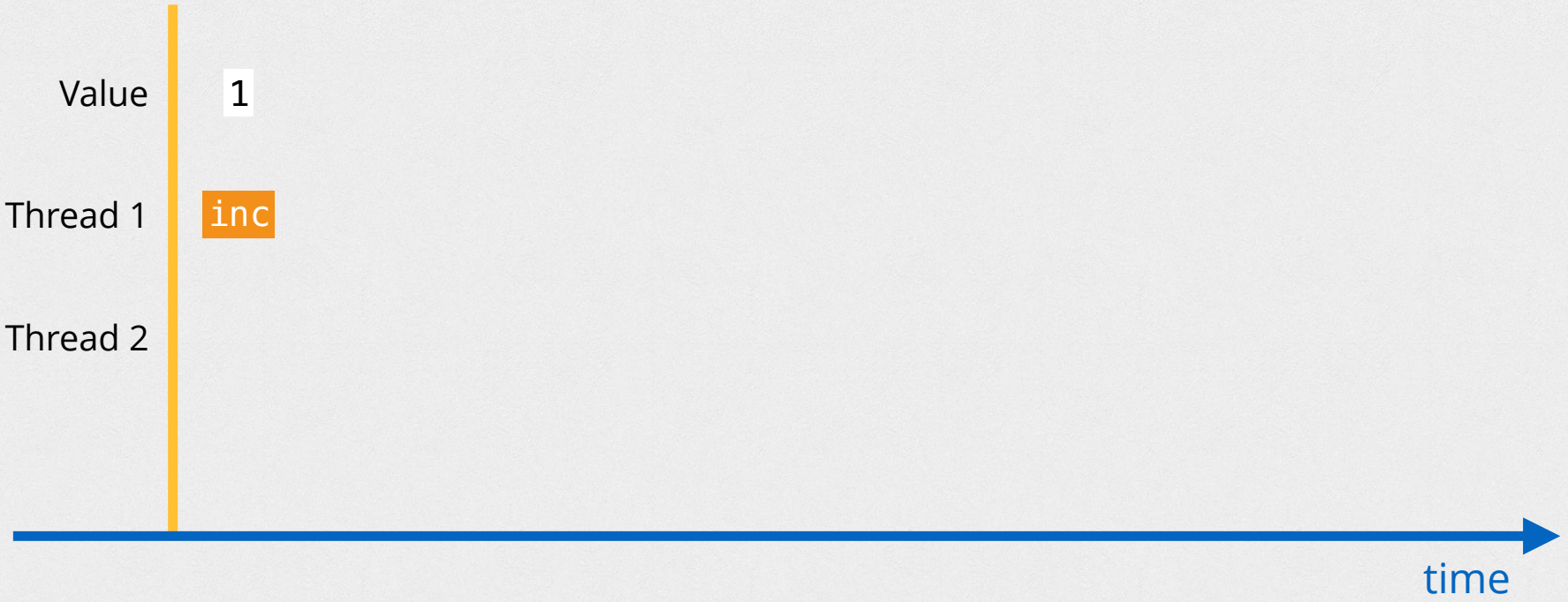
What is Concurrency?

- Multiple tasks taking place simultaneously
- Utilises multi-core CPUs
- Can make apps much more responsive
- Traditionally uses a threading model

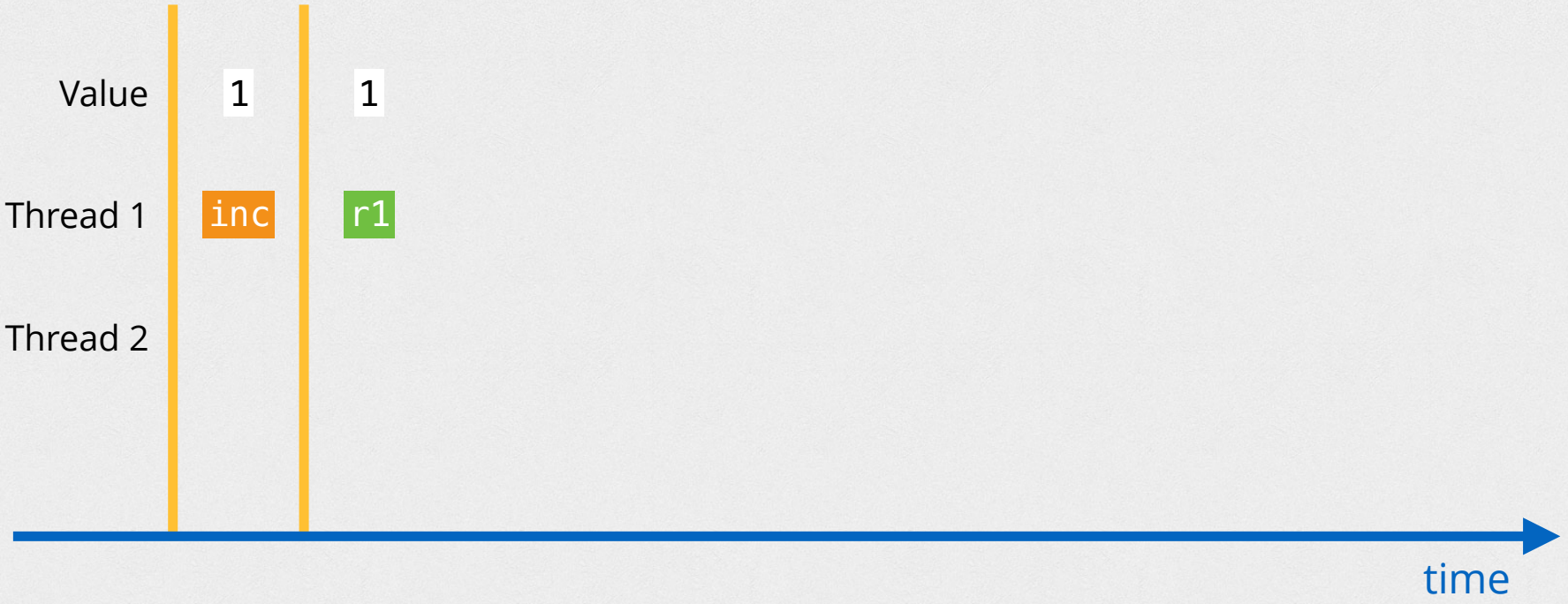
Issues with multi-threading

- Easy to get wrong
- Difficult to debug
- Specific scenarios
  - Race condition
  - Priority inversion
  - Deadlock

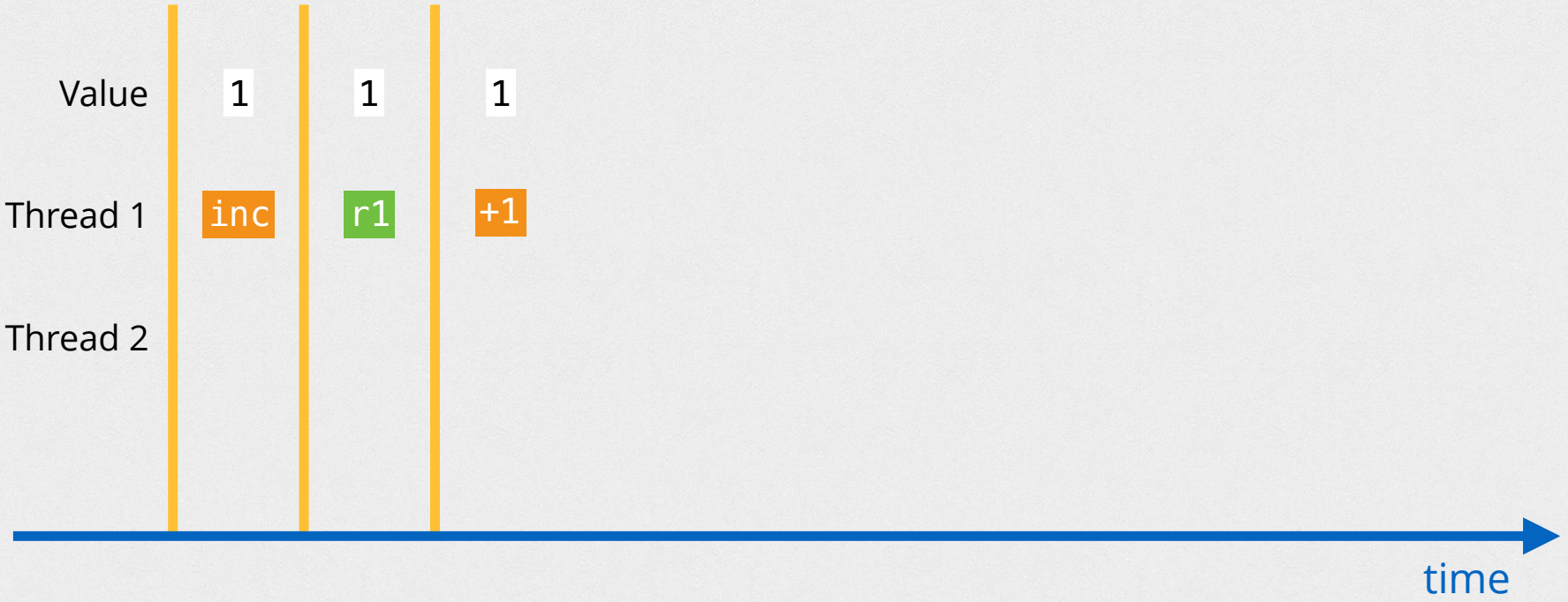
Race Condition



Race Condition

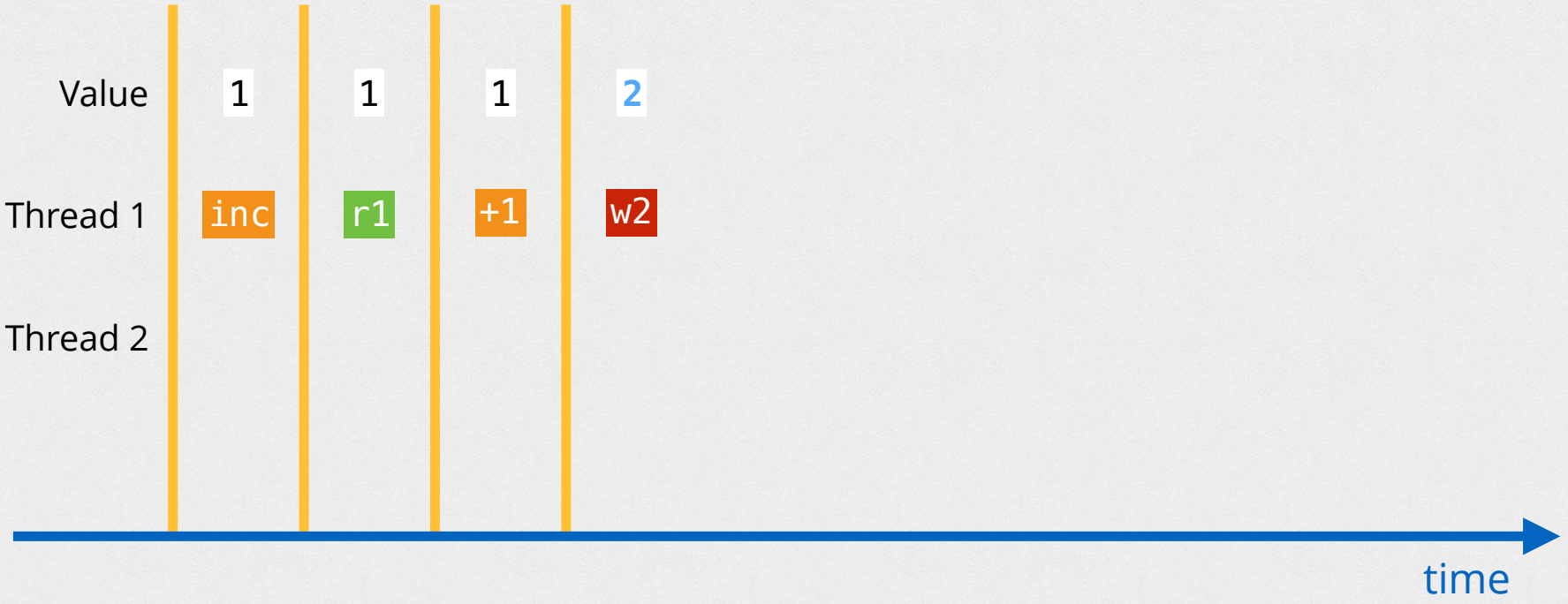


Race Condition



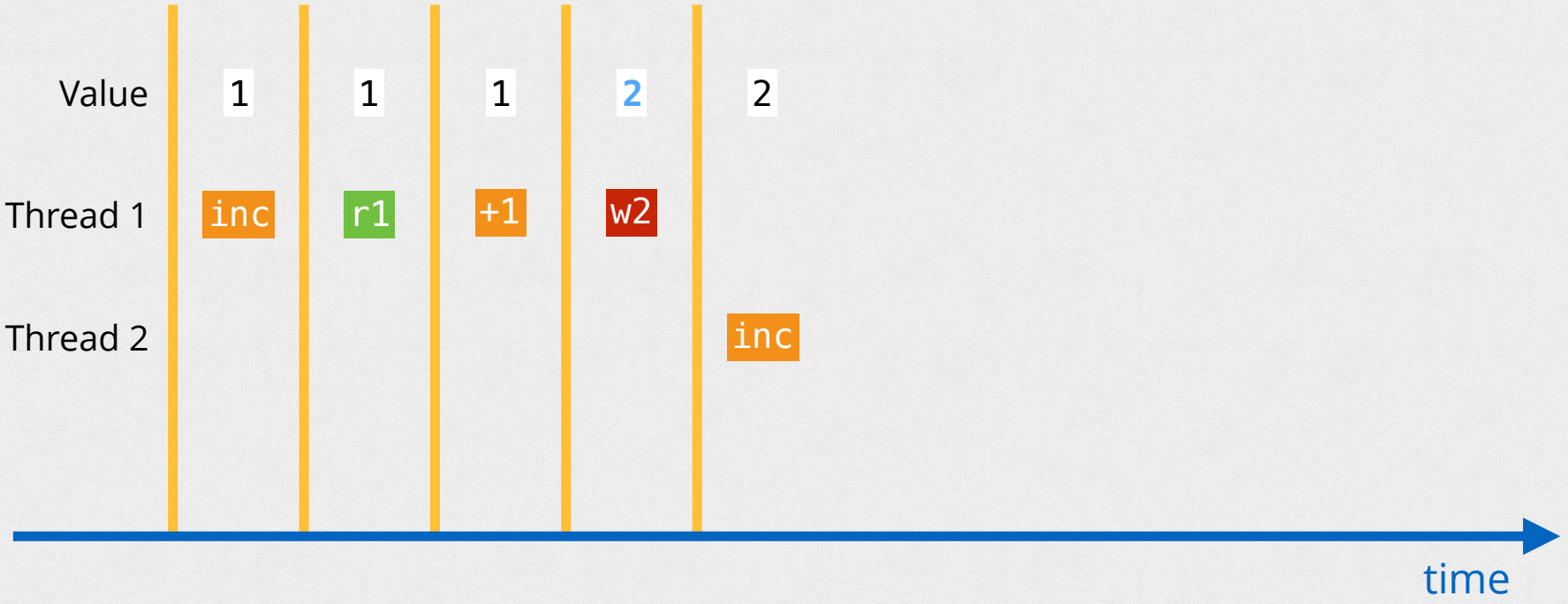
raywenderlich.com

Race Condition



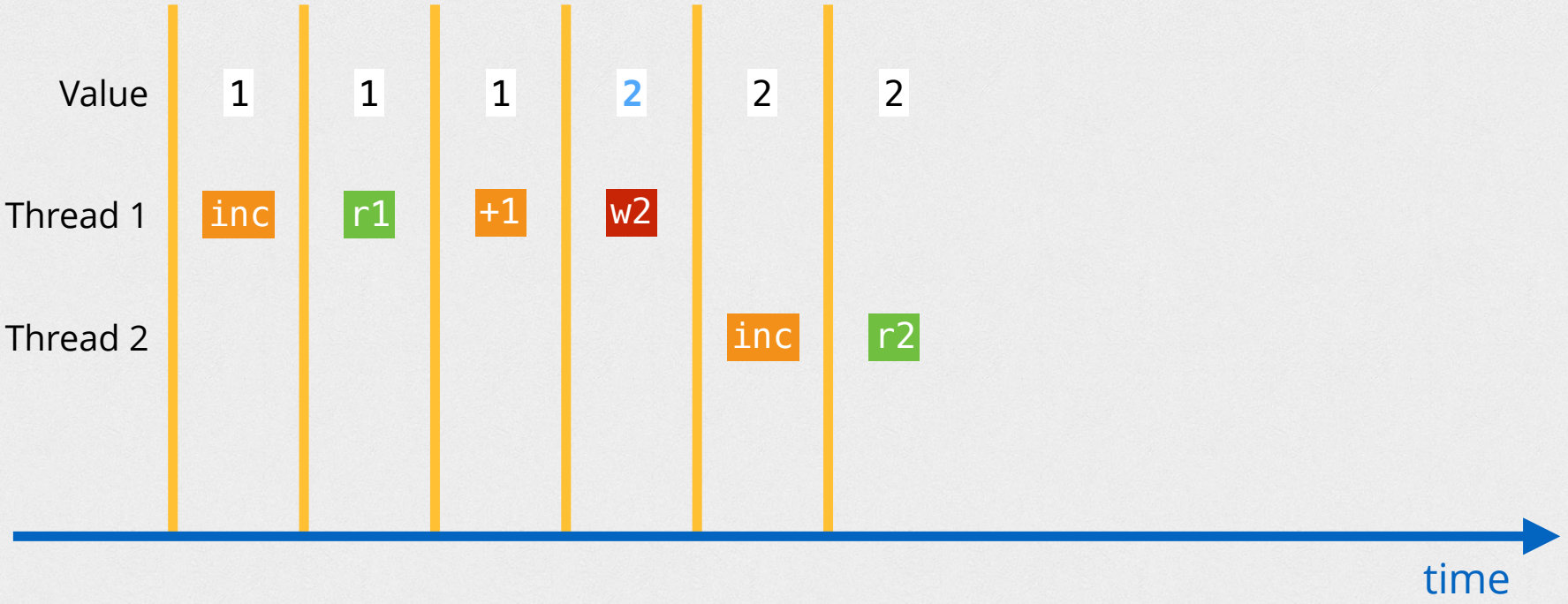
raywenderlich.com

Race Condition



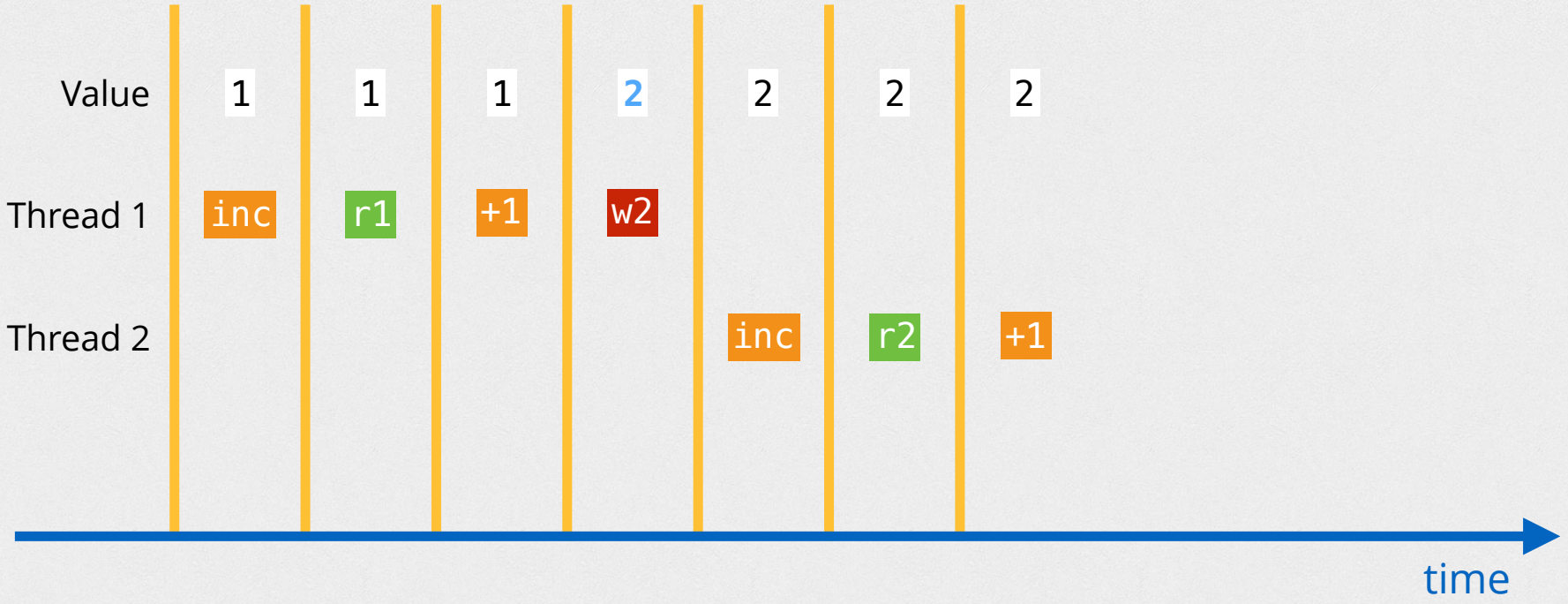
raywenderlich.com

Race Condition



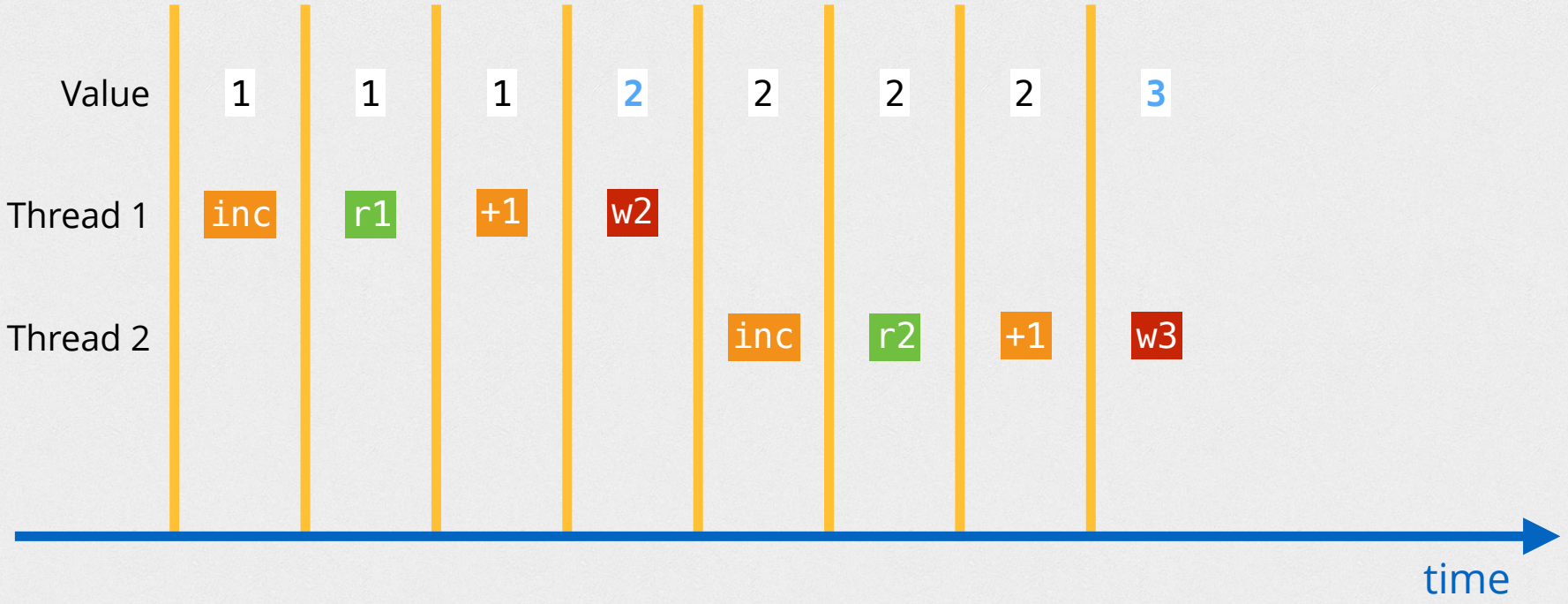
raywenderlich.com

Race Condition



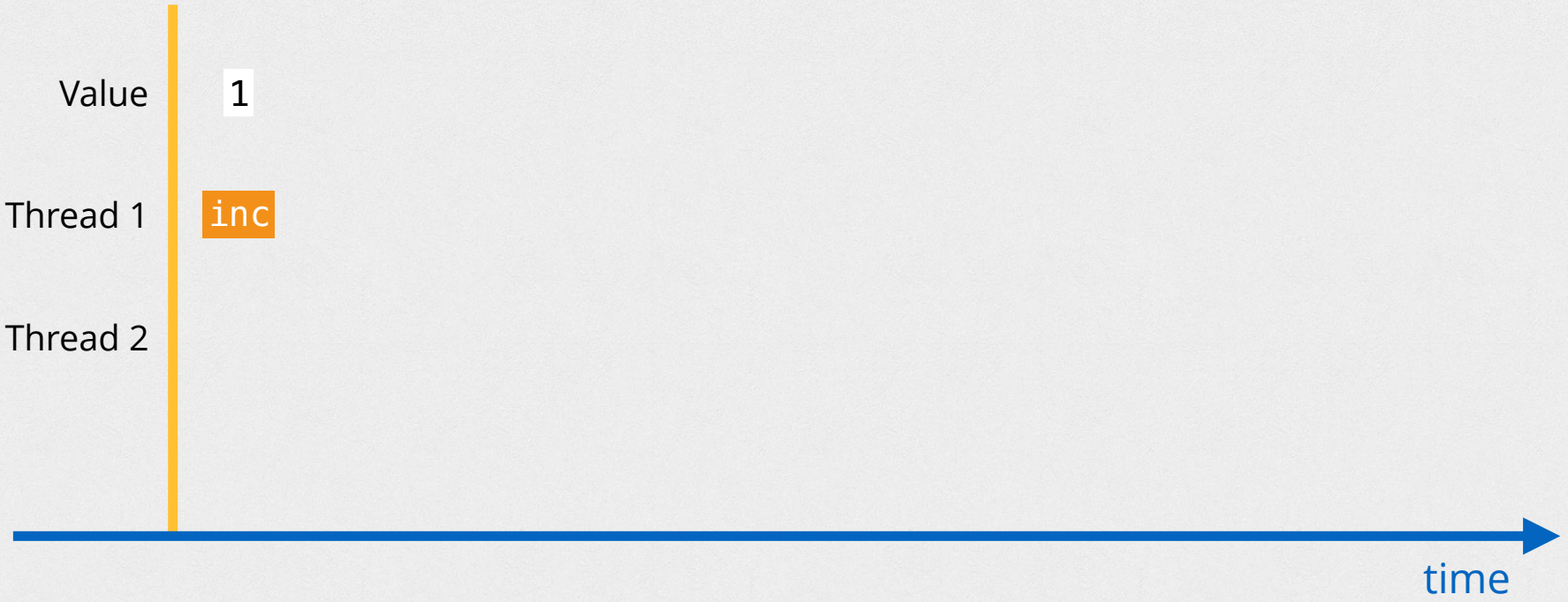
raywenderlich.com

Race Condition

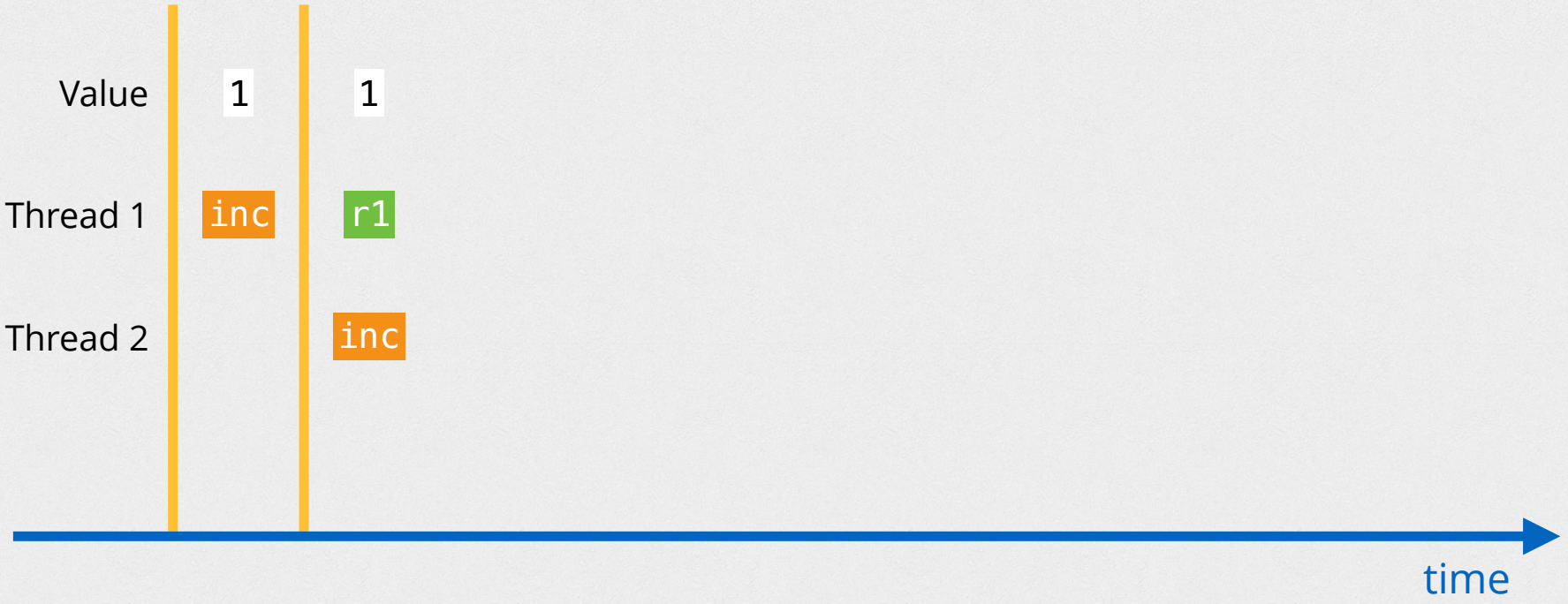


raywenderlich.com

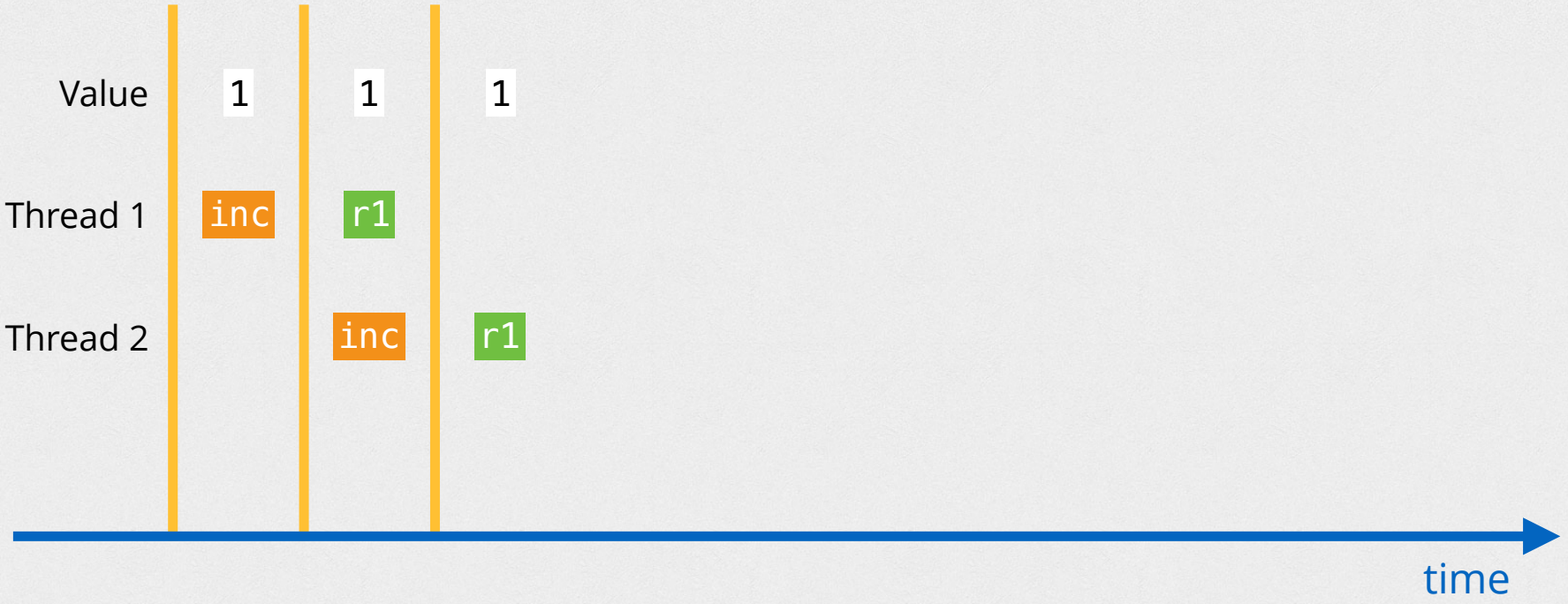
Race Condition



Race Condition

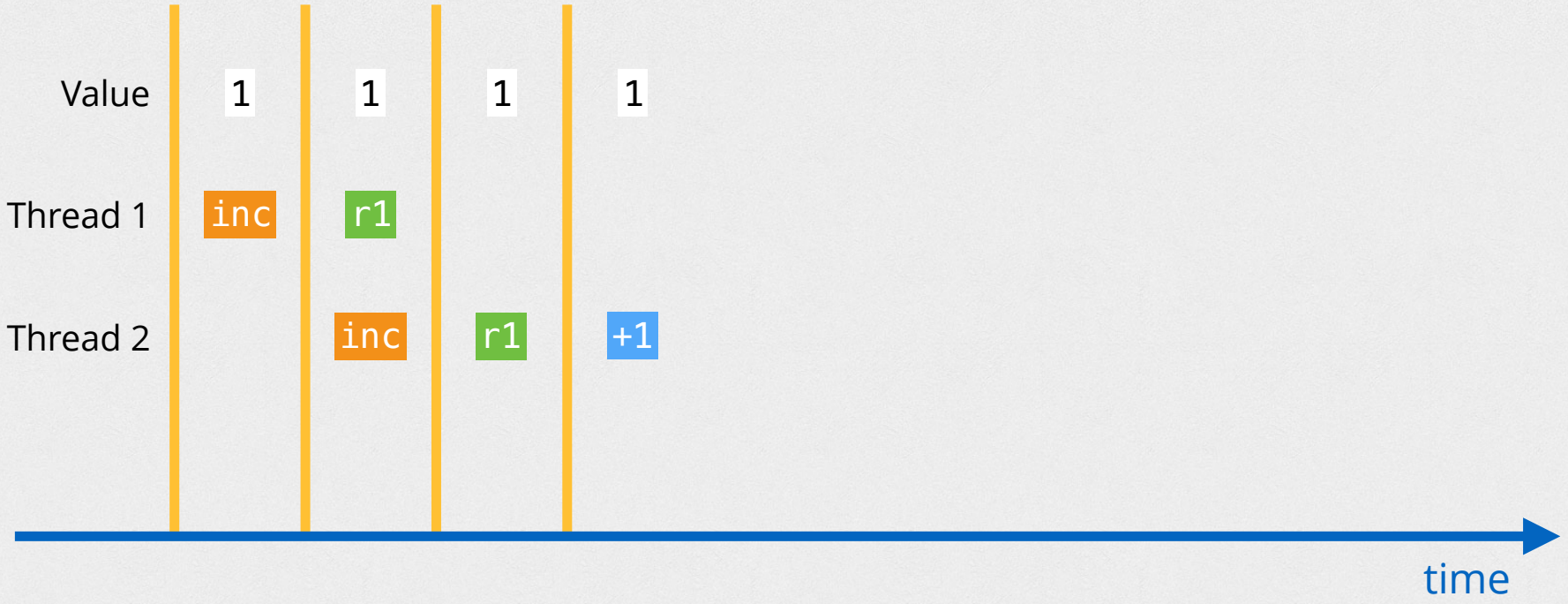


Race Condition



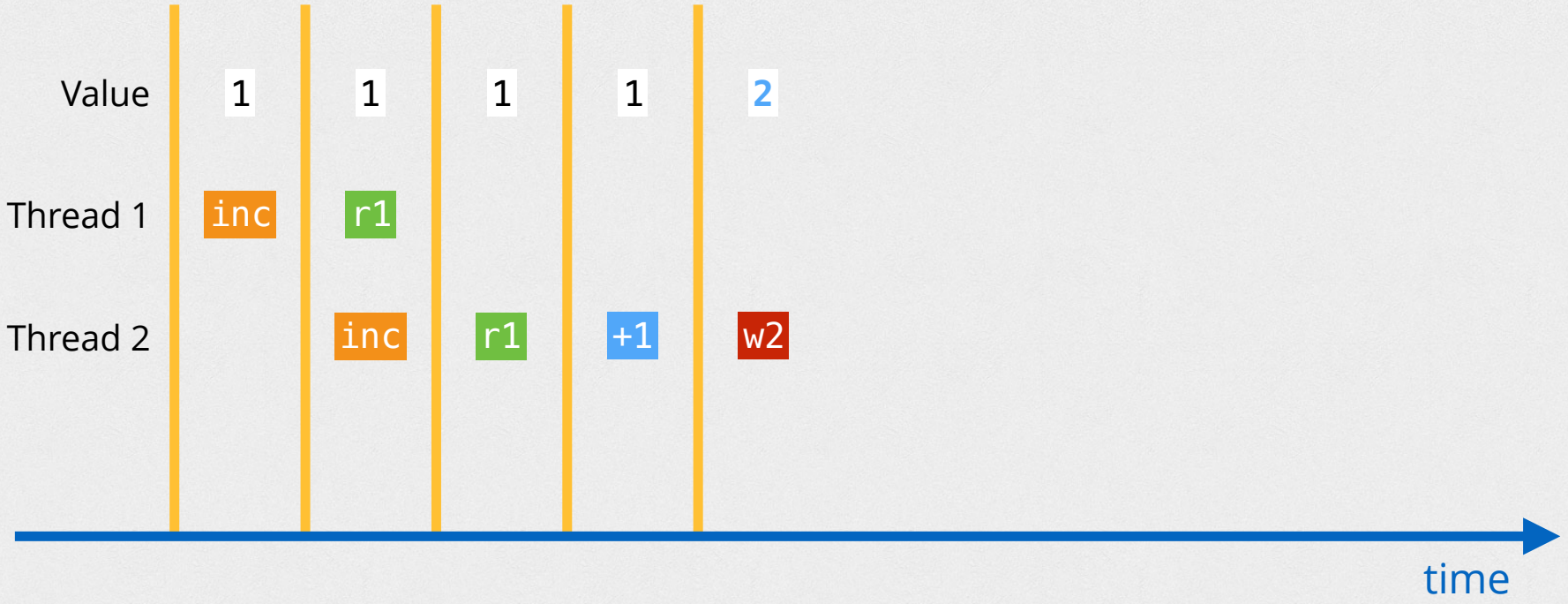
raywenderlich.com

Race Condition



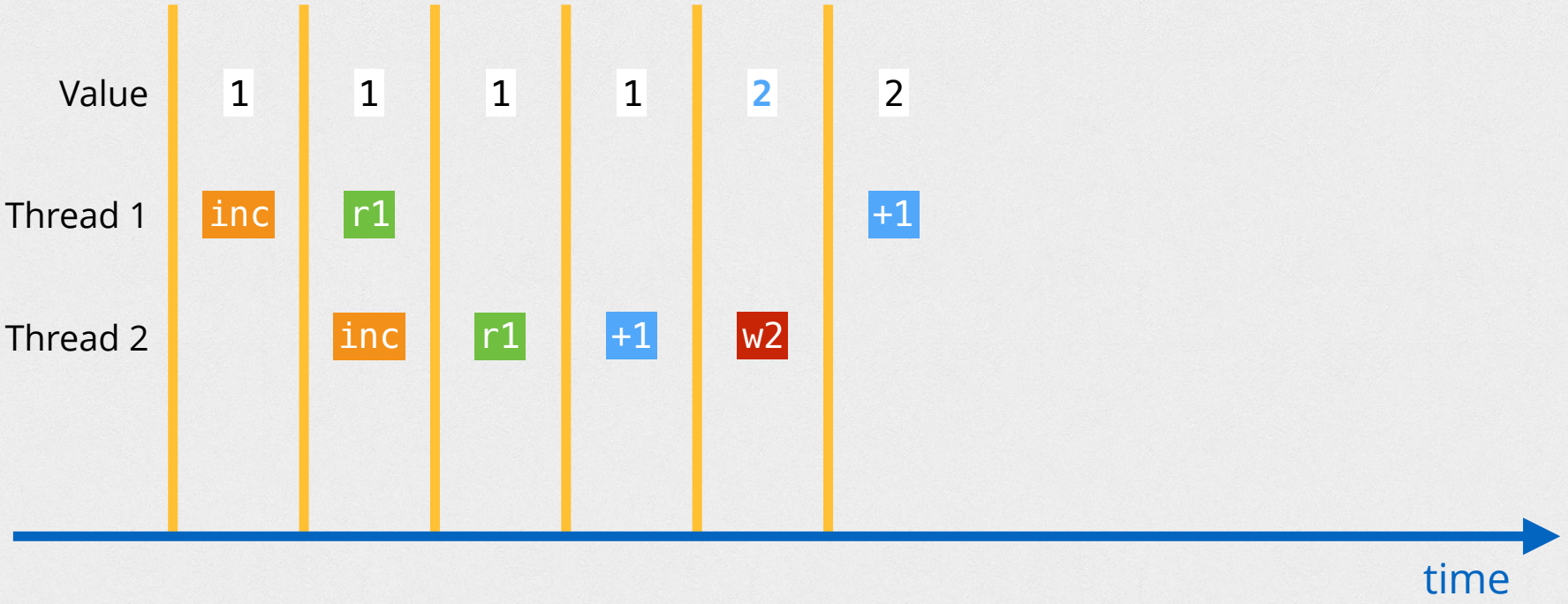
raywenderlich.com

Race Condition

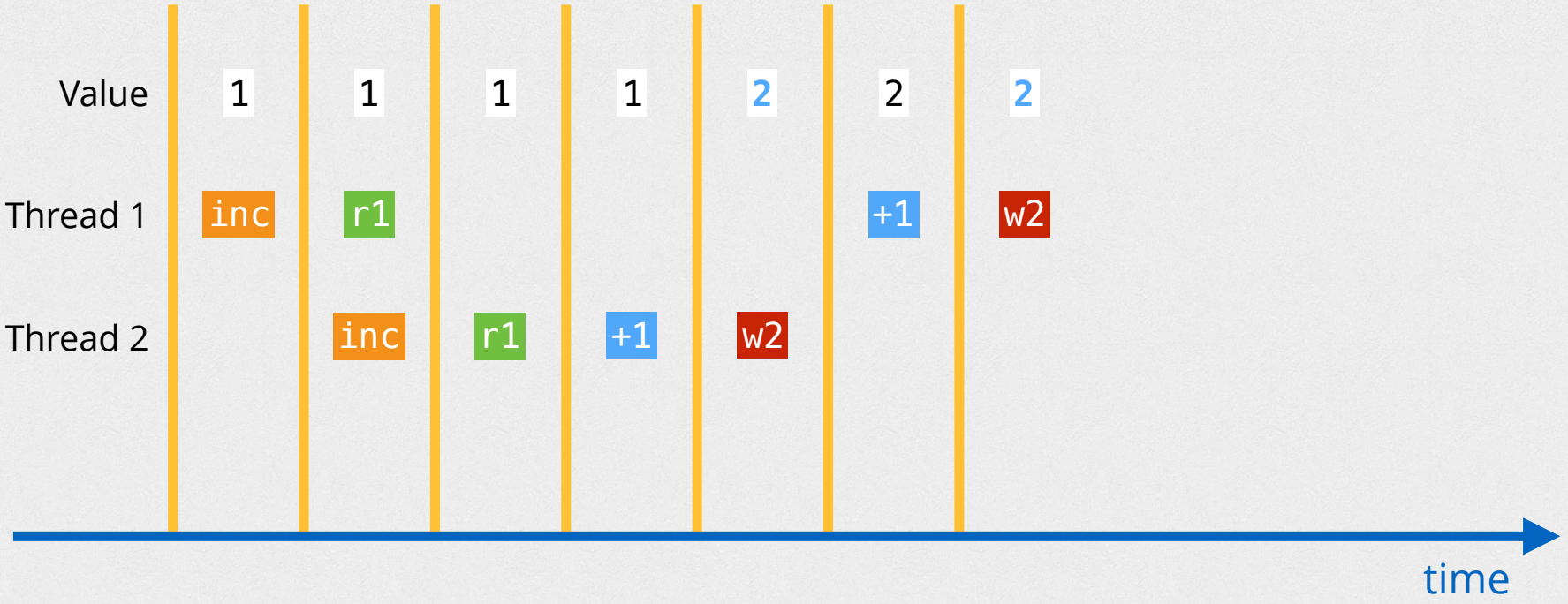


raywenderlich.com

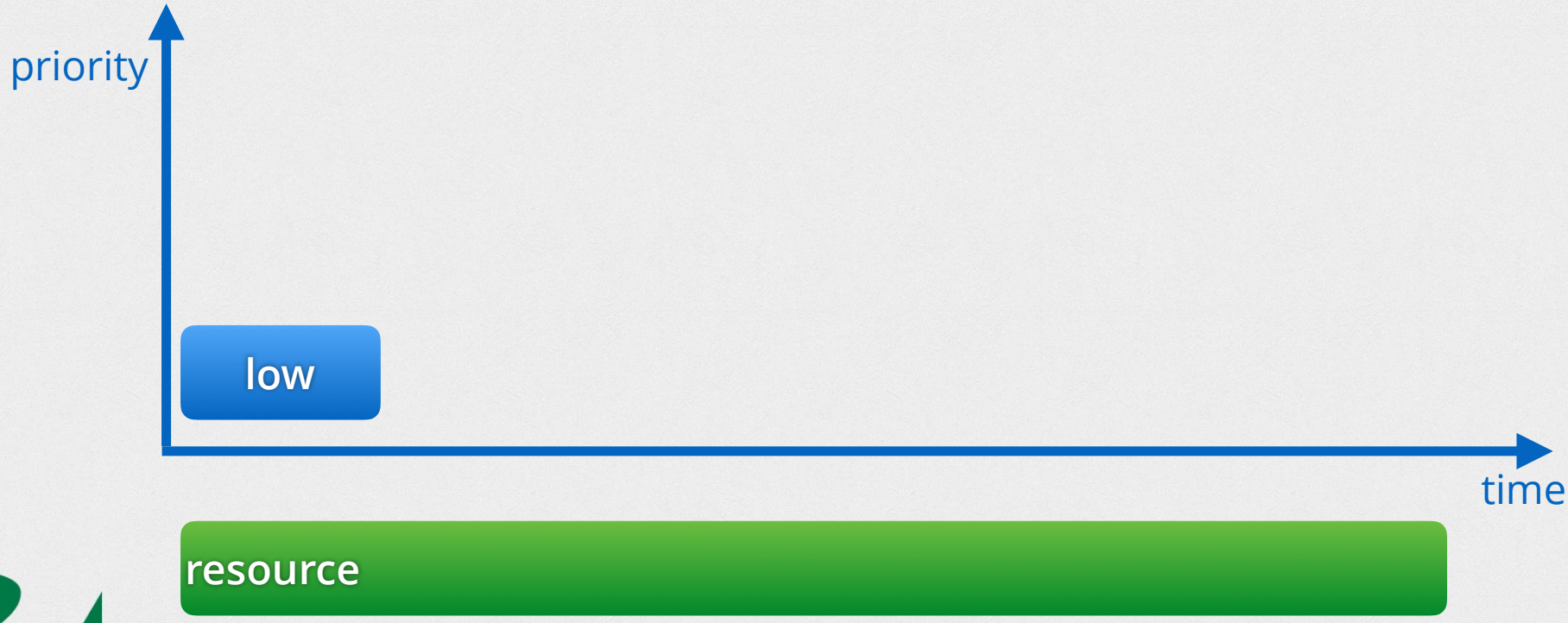
Race Condition



Race Condition

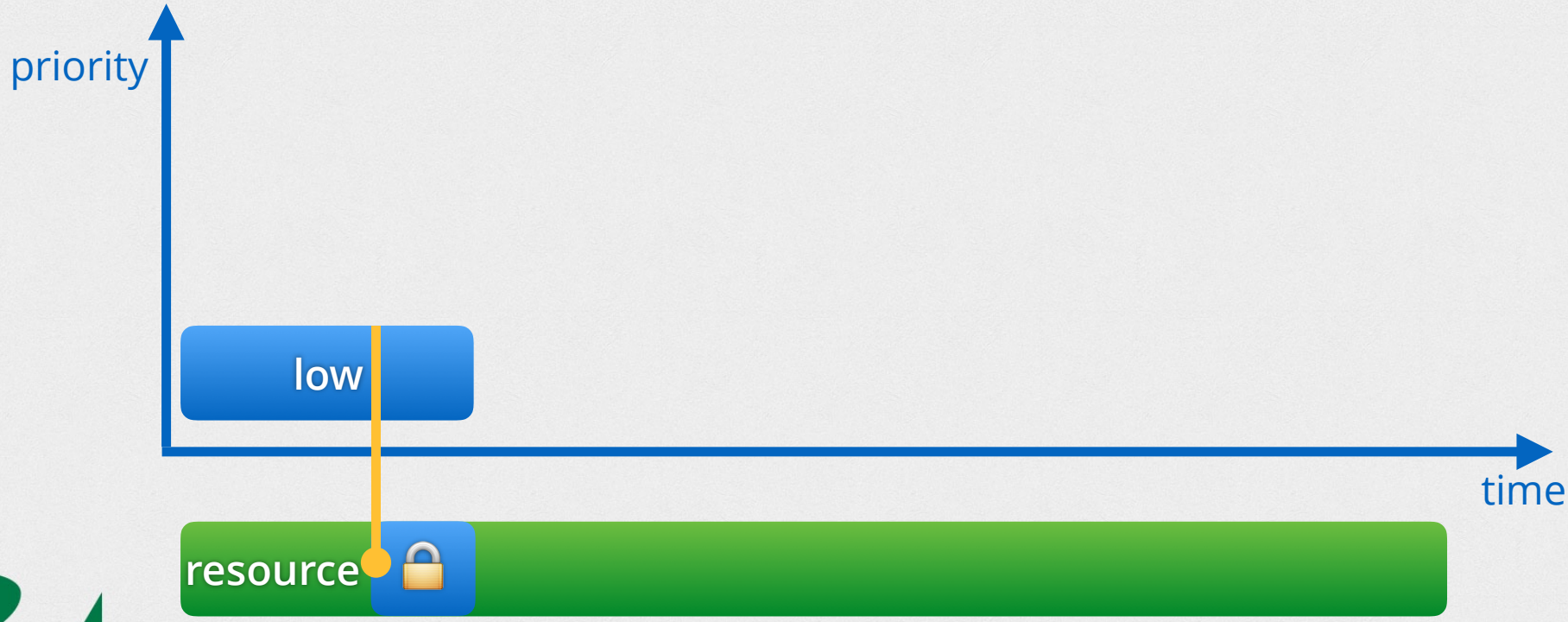


Priority Inversion



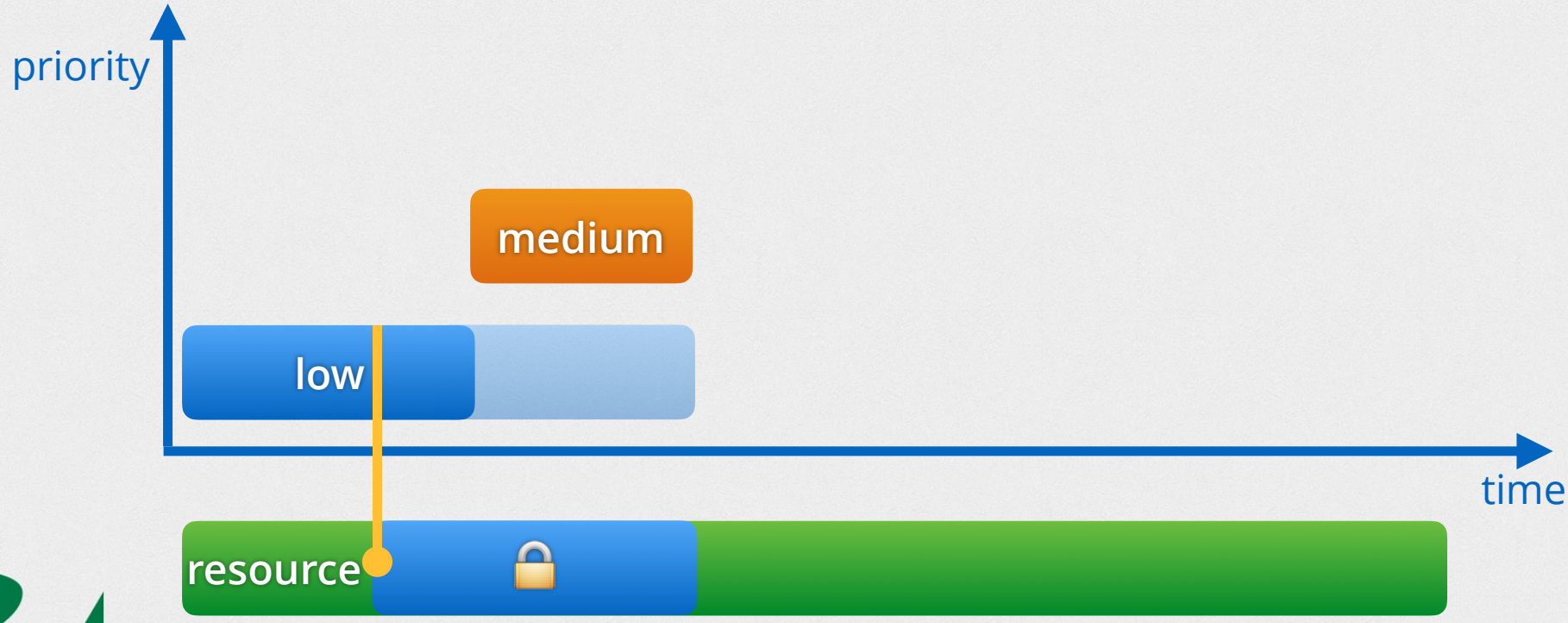
raywenderlich.com

Priority Inversion



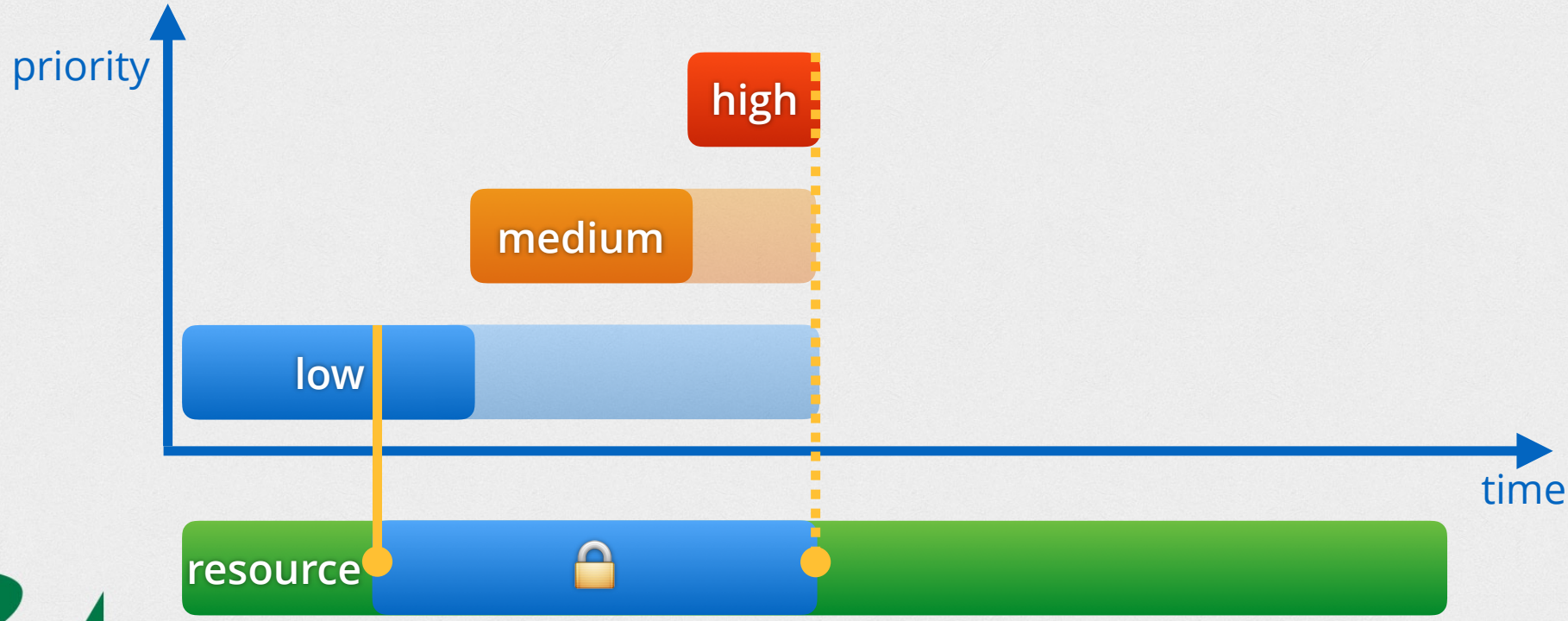
raywenderlich.com

Priority Inversion



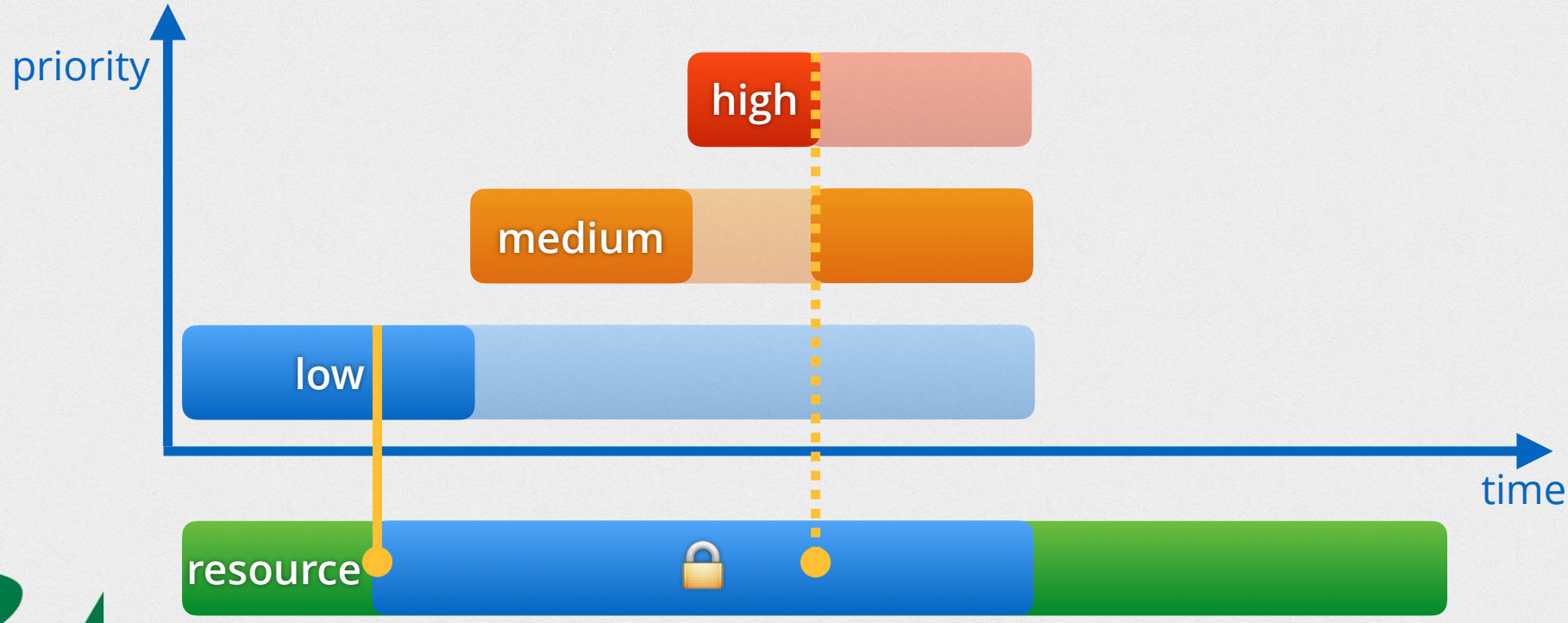
raywenderlich.com

Priority Inversion



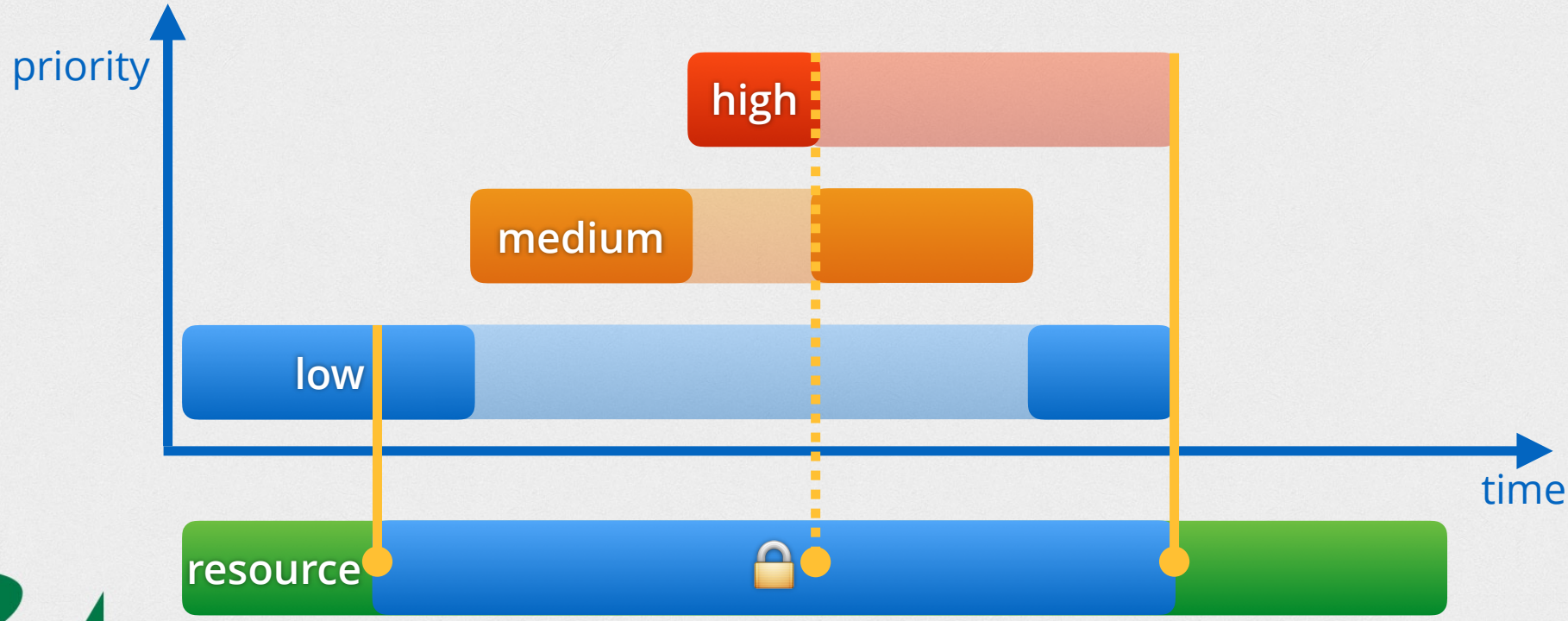
raywenderlich.com

Priority Inversion



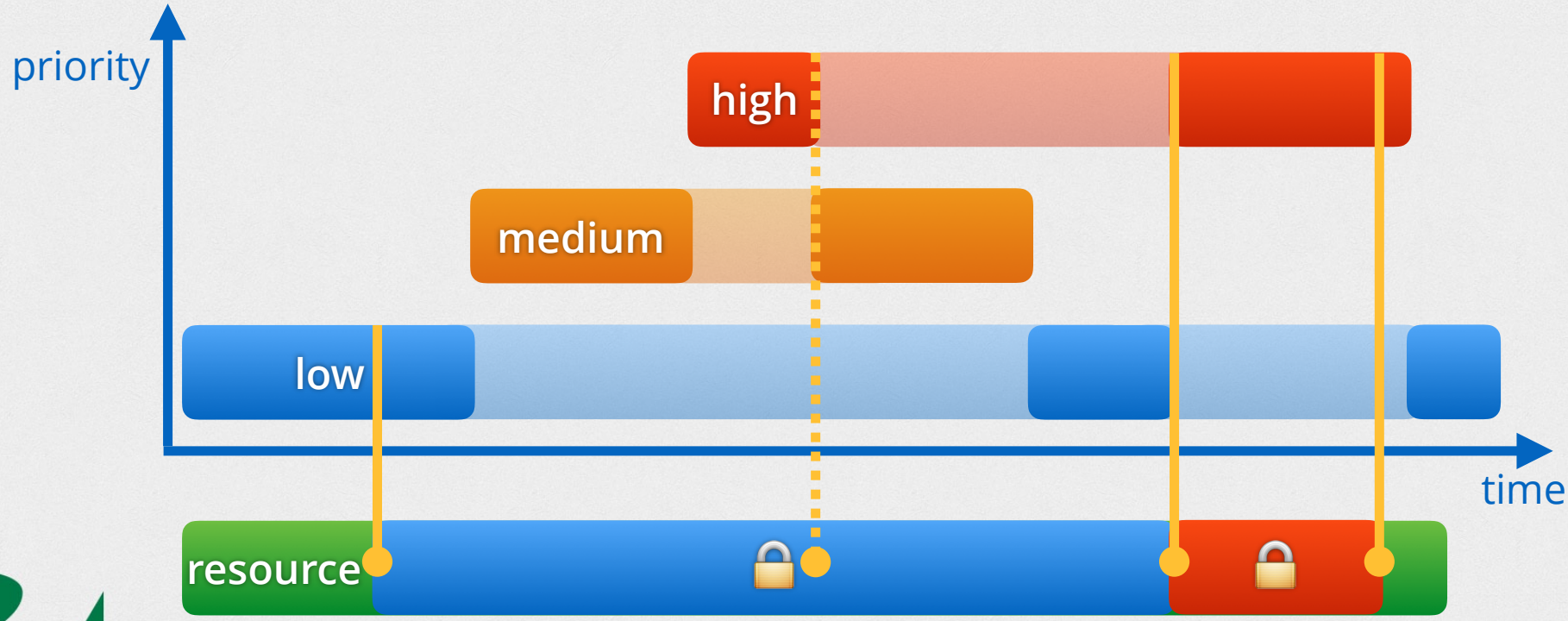
raywenderlich.com

Priority Inversion



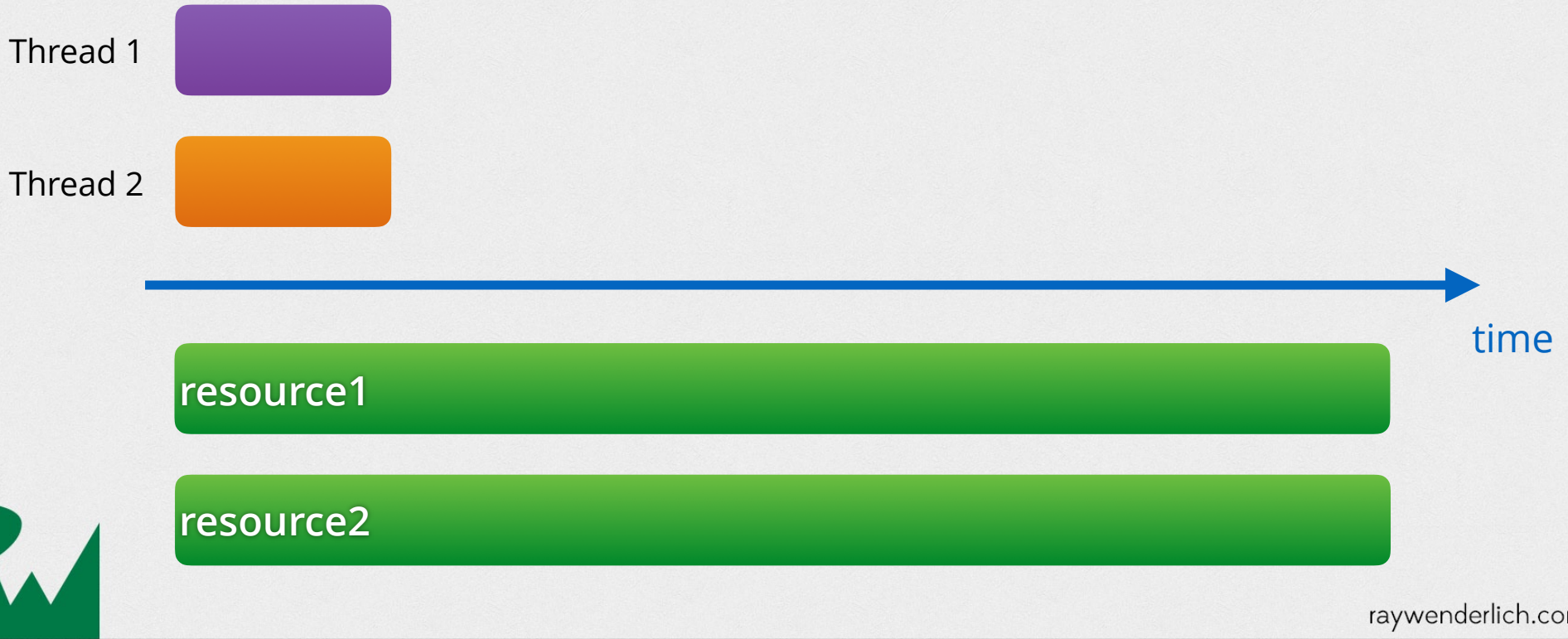
raywenderlich.com

Priority Inversion

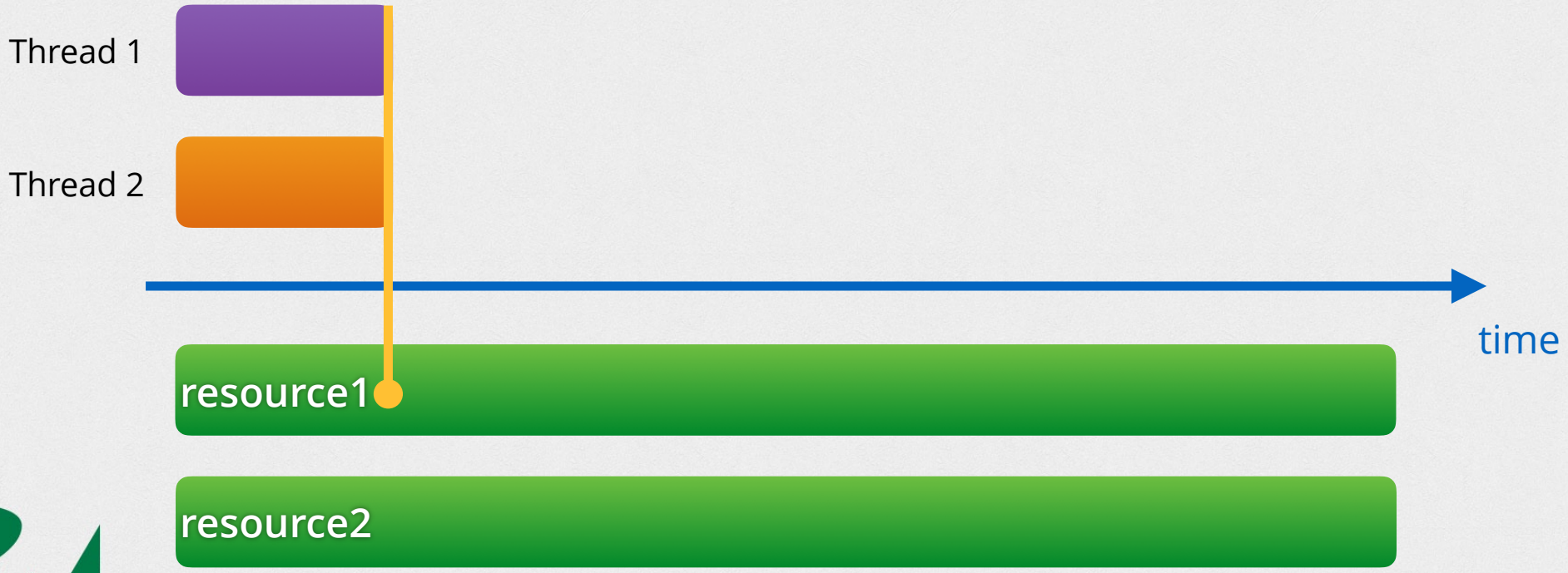


raywenderlich.com

Deadlock

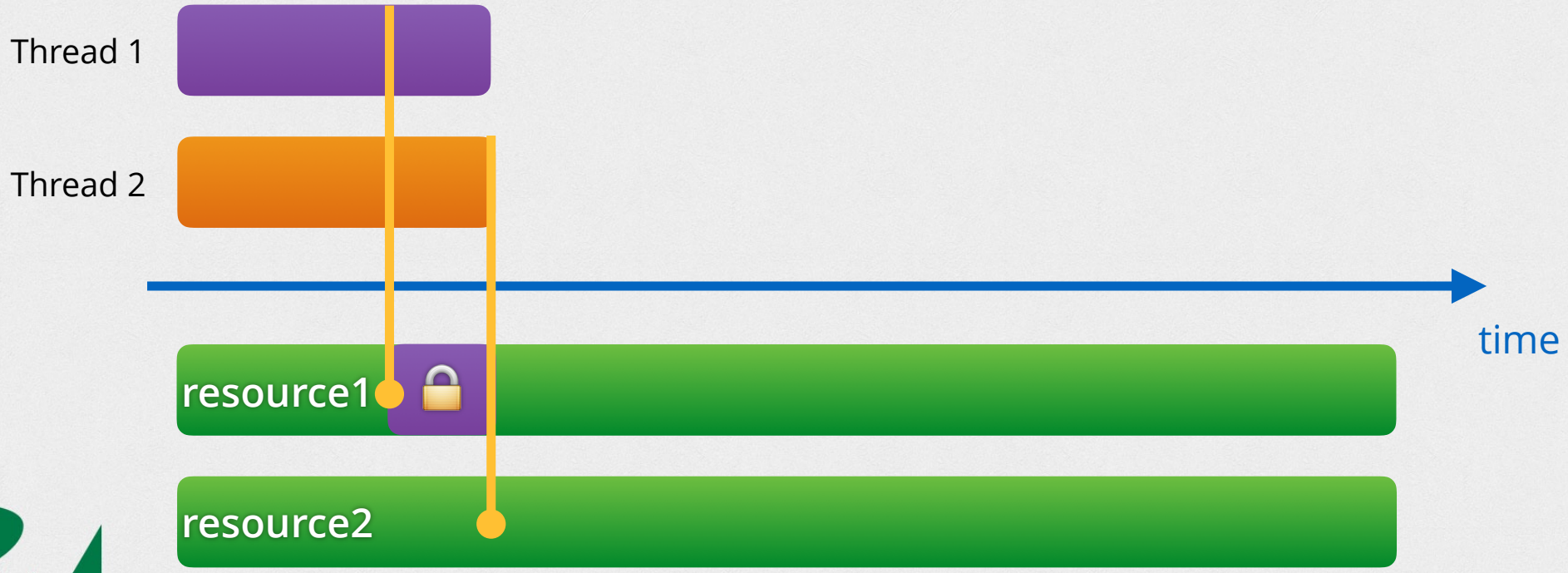


Deadlock



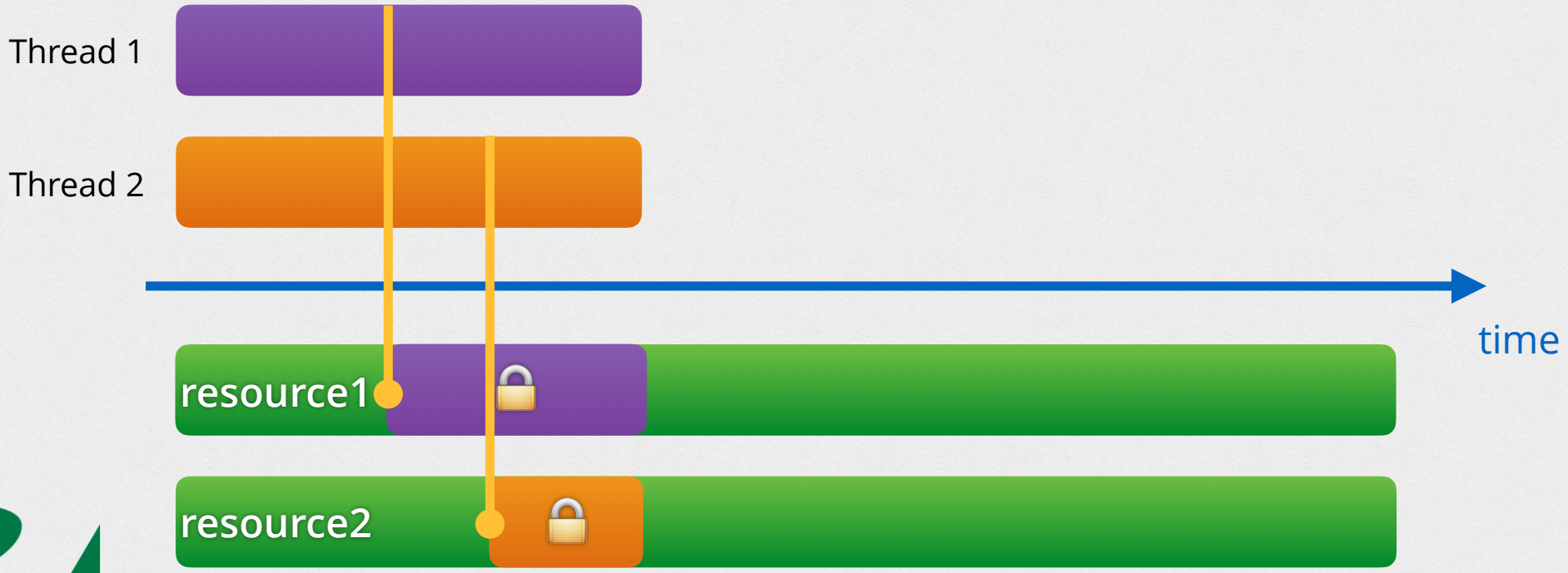
raywenderlich.com

Deadlock



raywenderlich.com

Deadlock



raywenderlich.com

Deadlock

raywenderlich.com

Deadlock

raywenderlich.com

Deadlock

raywenderlich.com

Deadlock

raywenderlich.com

Tasks & Queues

- Simpler abstraction
- Scales with system and conditions
- Lower overhead
- Serial queues include intrinsic locking

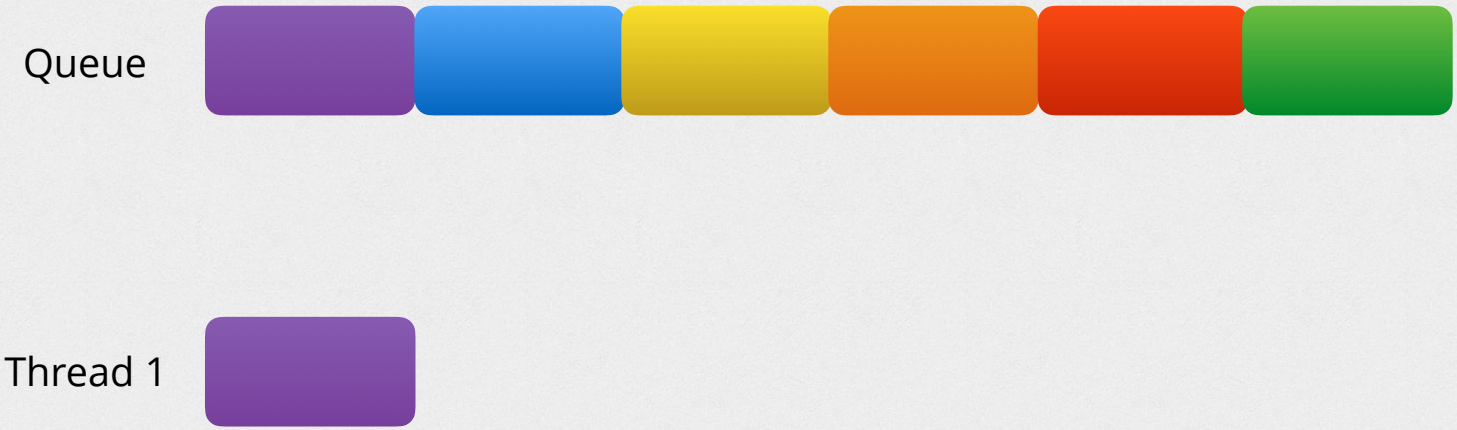
Tasks & Queues

Queue

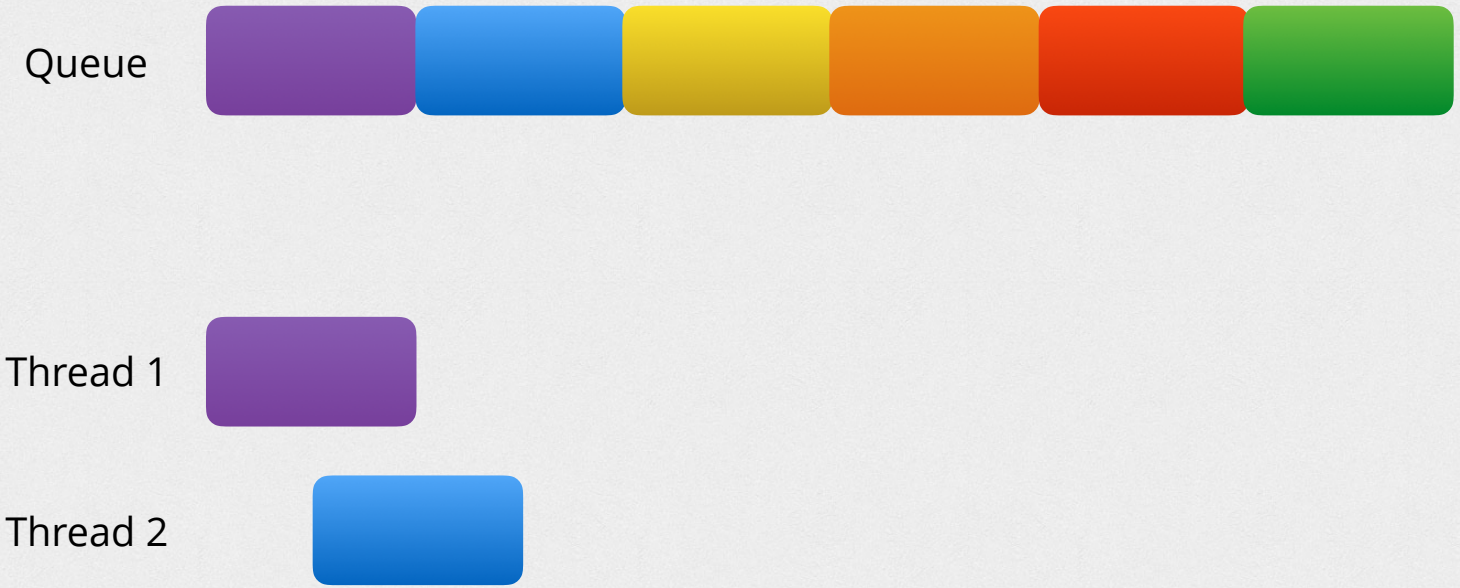
Thread 1

Thread 2

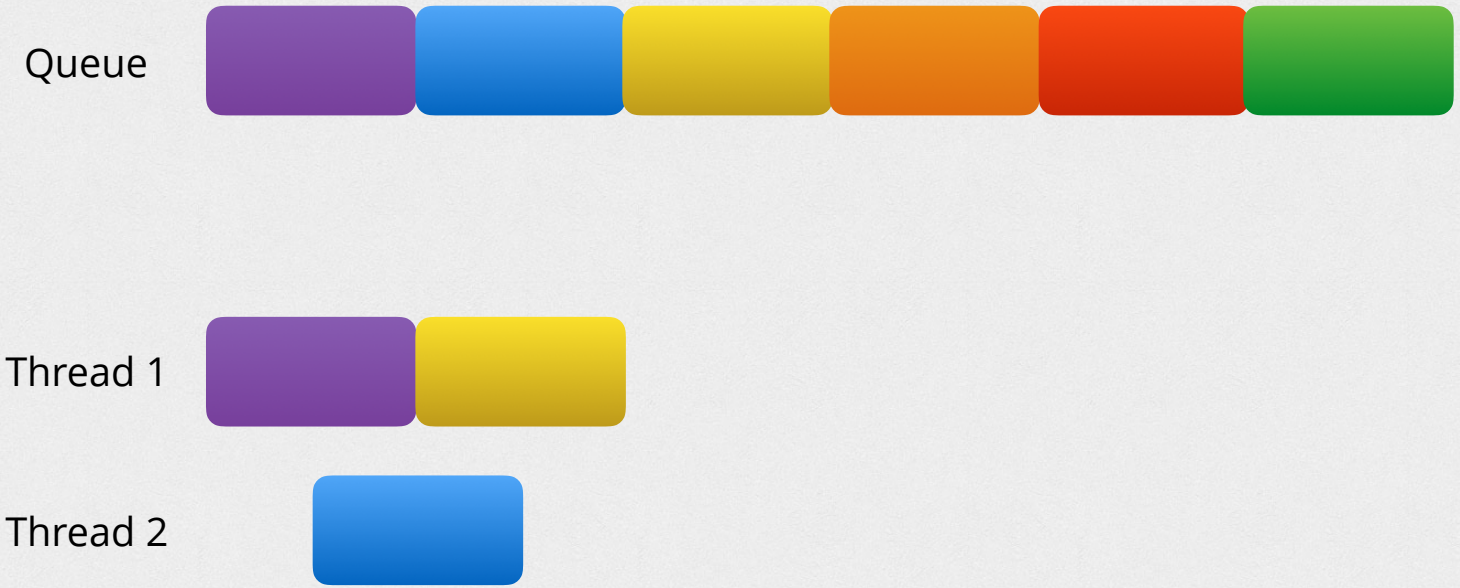
Tasks & Queues



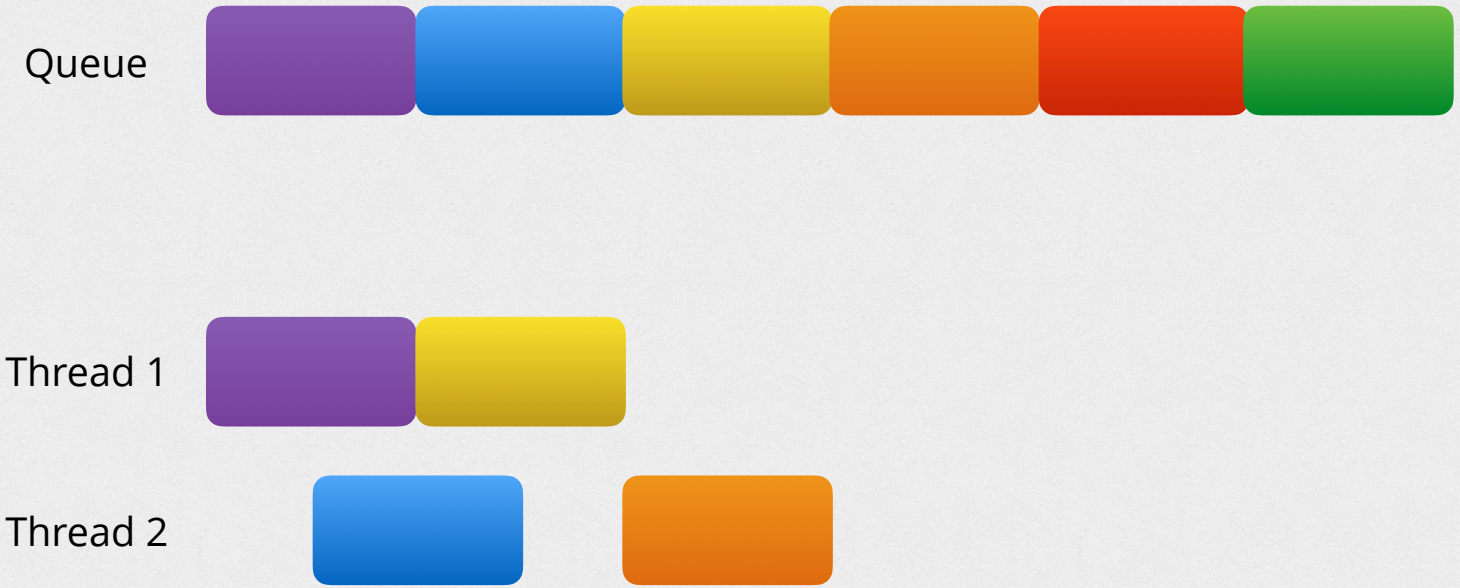
Tasks & Queues



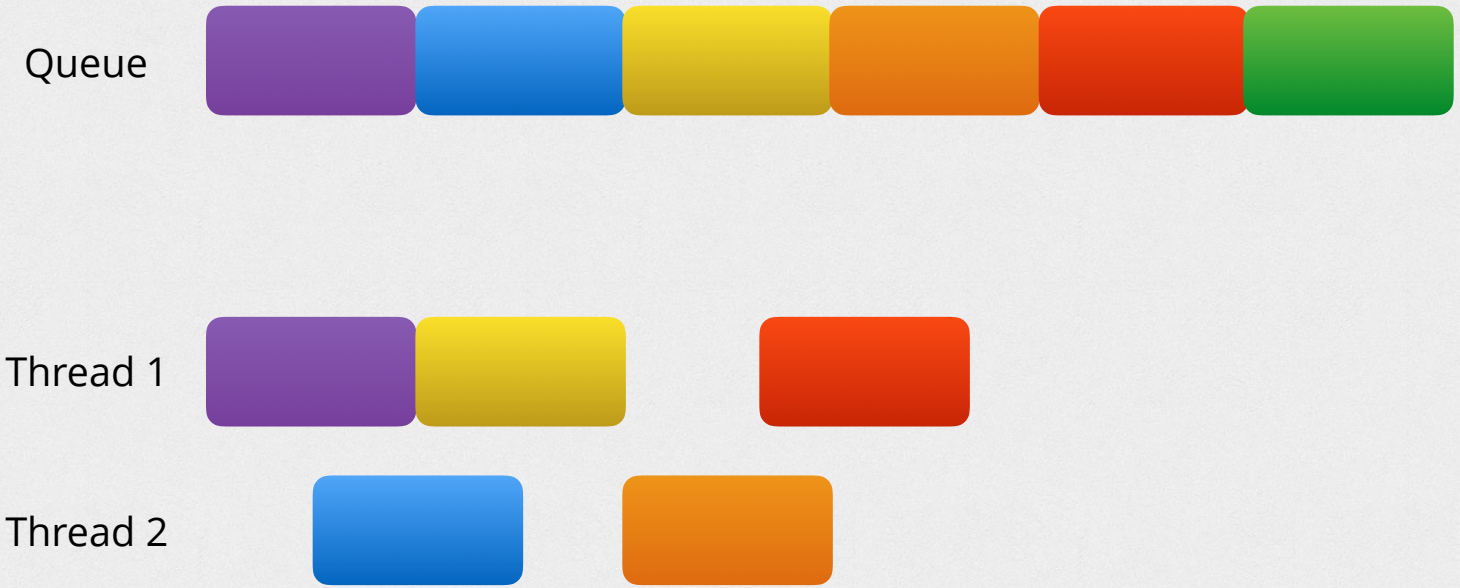
Tasks & Queues



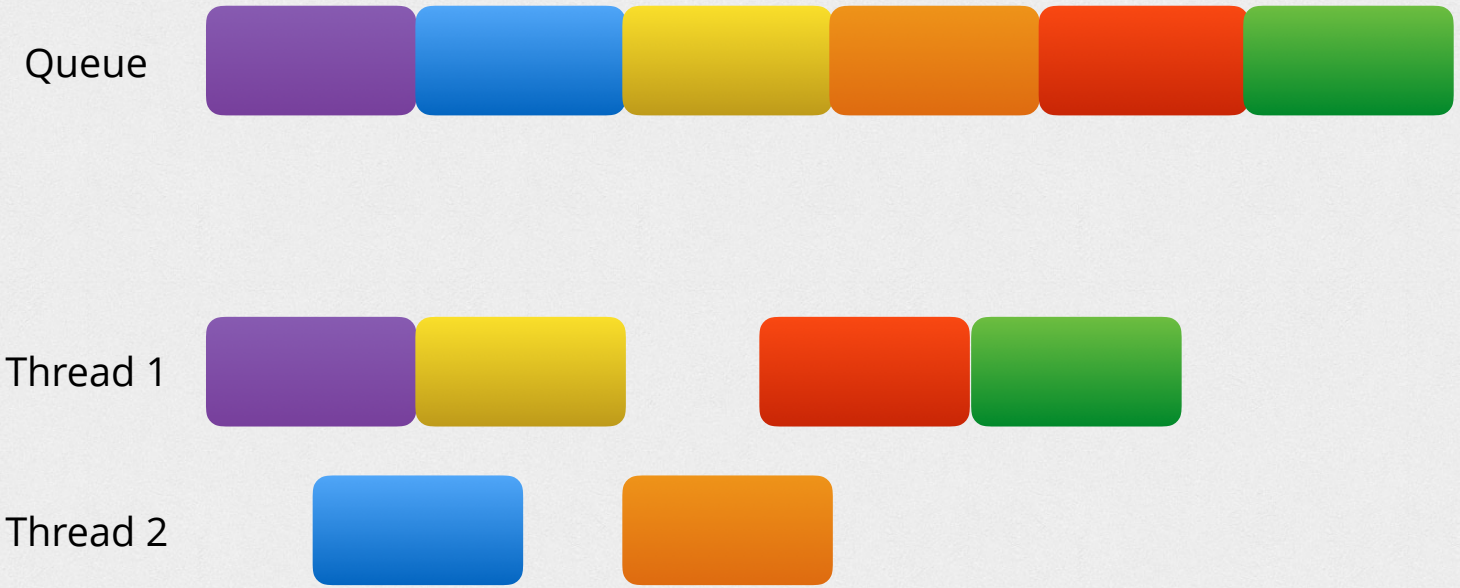
Tasks & Queues



Tasks & Queues

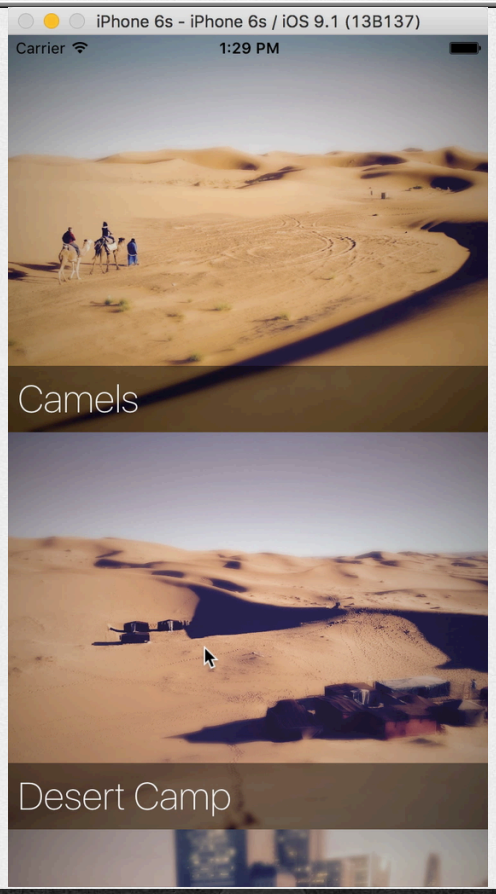


Tasks & Queues



Act I: NSOperation

- NSOperation
- NSOperationQueue
- Async Operations
- Dependencies
- Cancellation
- NSOperation in Practice



Act II: Grand Central Dispatch

- Fire & Forget with GCD
- Dispatch Groups
- Thread-safe resources
- GCD Delights

Where to go from here?