

Introducing

Concurrency

Conclusion

raywenderlich.com

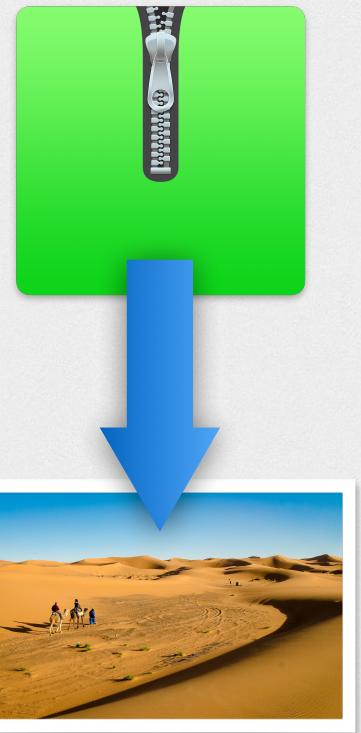
Act I: What have you learned?

⚙ NSOperation

- ⚙ Bundle up a unit of work
- ⚙ States: ready, executing, cancelled, finished
- ⚙ Either subclass or use NSBlockOperation

⚙ NSOperationQueue

- ⚙ Handles scheduling and running of NSOperations
- ⚙ Manages concurrency



raywenderlich.com

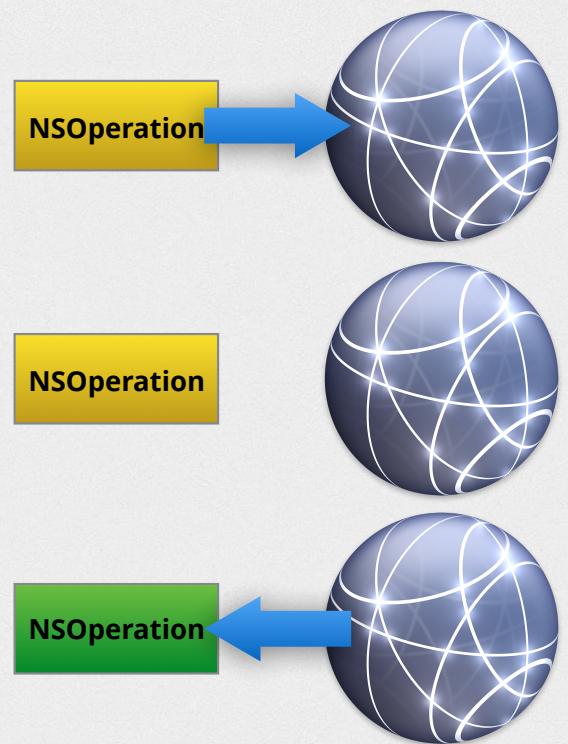
Act I: What have you learned?

⚙ Concurrent NSOperation

- ⚙ By default expects serial operation
- ⚙ Can wrap async function by subclassing
- ⚙ Have to manage states and KVO

⚙ Dependencies

- ⚙ Specify a dependency graph of operations
- ⚙ Pass data via the dependencies property



Act I: What have you learned?

⚙️ Cancelling NSOperation

- ⚙️ Must respect the cancelled property
- ⚙️ Can cancel an entire queue

⚙️ NSOperation in practice

- ⚙️ Wrap expensive operations in NSOperation
- ⚙️ Cancel when no longer needed



raywenderlich.com

Act II: What have you learned?

⚙️ Grand Central Dispatch

- ⚙️ Tasks & queues
- ⚙️ QoS
- ⚙️ Dispatching tasks



⚙️ Dispatch Groups

- ⚙️ Notify or wait for a set of tasks to complete



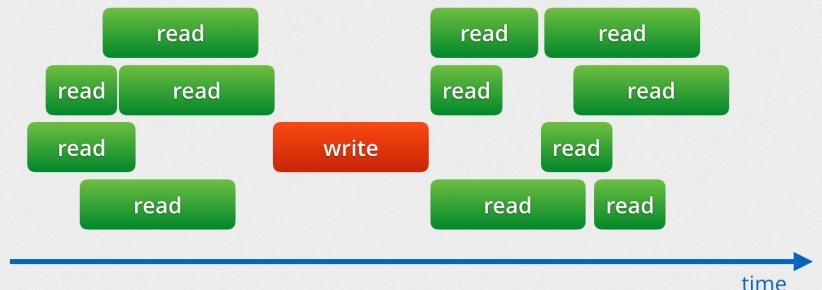
Act II: What have you learned?

⚙ Thread Safety with Barriers

- ⚙ Protect critical section with a `dispatch_barrier`
- ⚙ Making a thread safe object
- ⚙ `dispatch_once`
 - ⚙ Blocks other threads for one-time setup

⚙ `dispatch_later`

- ⚙ Execute closure at some future point



NSOperation -vs- libdispatch

- ⚙ Start with highest level - NSOperation
 - ⚙ Easier to handle - dependencies & cancellation
 - ⚙ Built on top of libdispatch
 - ⚙ Small overhead
- ⚙ Drop down if needed
 - ⚙ Performance critical
 - ⚙ Features not available in NSOperation



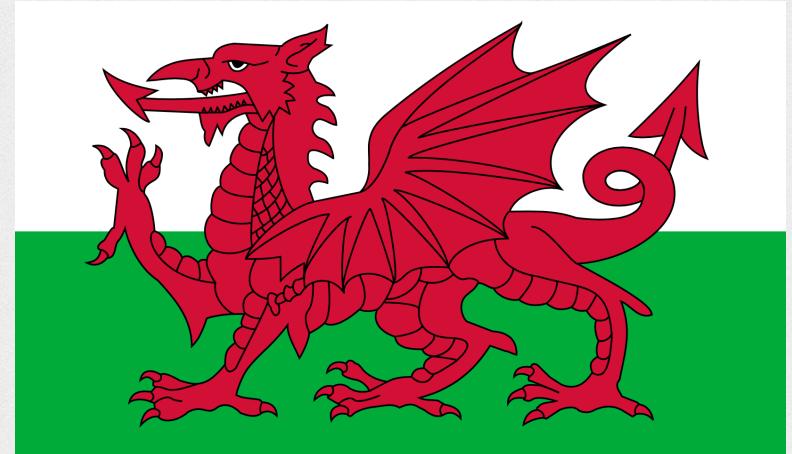
Here be dragons

- ⚙️ Deadlock

- ⚙️ Priority Inversion

- ⚙️ Thread safety

- ⚙️ UIKit is not thread safe



Where to go from here?

Grand Central Dispatch Tutorial for Swift: Part 1/2



Bjørn Ruud on January 7, 2015

[Tweet](#) [Share](#) 64

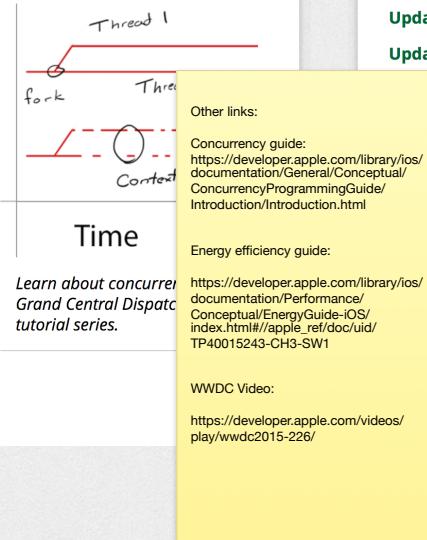
Update April 10, 2015: Updated for Xcode 6.3 / Swift 1.2

Update note: This tutorial was updated for iOS 8 and Swift by [Bjørn Ruud](#). [Original post](#) by Tutorial Team member [Derek Selander](#).

Although **Grand Central Dispatch** (or GCD for short) has been around for a while, not everyone knows how to get the most out of it. This is understandable; concurrency is tricky, and GCD's C-based API can seem like a set of pointy corners poking into the smooth world of Swift.

In this two-part series, you'll learn the ins and outs of GCD. This first part will explain what GCD does and showcase several of the more basic GCD functions. In the second part, you'll learn several of the more advanced functions GCD has to offer.

Getting Started



NSOperation and NSOperationQueue Tutorial in Swift



Richard Turton on October 7, 2014

[Tweet](#) [Share](#) 78

Update 17 April 2015: Updated for Xcode 6.3 and Swift 1.2

Update note: This tutorial was updated to iOS 8, Xcode 6.1 by Richard Turton. [Original post](#) by Tutorial Team member [Soheil Azarpour](#).

One has had the frustrating experience of tapping a button, entering some text in an iOS or Mac app, when all of a sudden – WHAM, the user interface stops being responsive. On a Mac, your users get to stare at the hourglass or the scroll wheel rotating for a while until they can interact with the UI again. In an iOS app, users expect apps to respond immediately to their touches. Unresponsive apps feel clunky now, and usually receive bad reviews.

Making your app responsive is easier said than done. Once your app needs to perform more than a few tasks, things get complicated quickly. There isn't much time to perform heavy work in the main loop and still provide a responsive UI.

