Concurrency III

Share Nothing

Previously...

Definitions of Concurrency, Parallelism, and friends

Synchronization Primitives

Rule of Thumb

Methods, Classes, Modules, etc that require “state” are typically inferior to those that do not require “state”

Stateful code is hard to grok

Sharing state quickly becomes intractable

Idempotence

Def: A function is idempotent if it will always return the same value for a given input.

Let’s see code!

Immutability

Def: An object is immutable if its value can never be changed

Let’s see code!

Pool

Fact of (Programming) Life: Resource creation and allocation is expensive

Workaround: Create a “pool” of resources and reuse them as they become available

For concurrency: Replace “Resource” with “Thread”

Producer Consumer

Def: Learn this one and call it a day

Let’s see code!

Divide and Conquer

Def: Break up large problem into smaller sub-problems that share the *same* algorithm.

Let’s see code!

Map Reduce

Let’s See Code!

Reductions

(Super Confusing because I brought up Map Reduce above…)

Transformation of one problem (A) into another problem (B).