USF UNIVERSITY OF SAN FRANCISCO

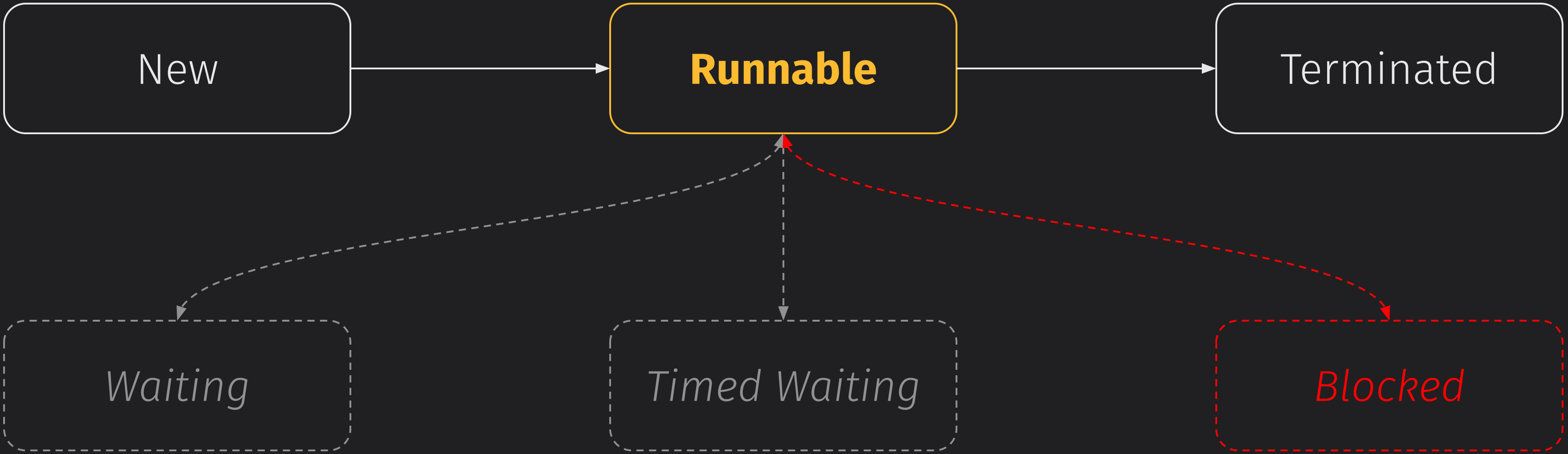
CHANGE THE WORLD FROM HERE

Thread Pools and Work Queues

CS 272 Software Development

Department of Computer Science
https://www.cs.usfca.edu/

Thread States



https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/lang/Thread.State.html

Motivation

- Goal: Web Server
  - Must handle multiple simultaneous requests
  - Must be **responsive** AND **efficient** (e.g. respond quickly, finish quickly)
- Implementation: Multithreading
  - One thread per request?

Problems

- Overhead cost to **creating objects**
  - Initialization in constructor (and `super()` calls)
- Overhead cost to **destroying objects**
  - Garbage collection
- Overhead cost to **excessive memory usage**
  - Causes thrashing

Solutions

- Keep Threads Around
  - Initialize a "wise" number of threads once
  - Reuse threads for other tasks instead of destroying
- Two-Part Approach
  - Thread pool (efficiency)
  - Work queue (responsiveness)

Thread Pools

- Create a fixed number of worker threads
- When have work to do...
  - Get available thread from pool and assign task
  - Thread runs assigned task
  - Thread returns to pool of available threads
- What if there are no available threads?

Work Queue

- Add a work queue to thread pool
- Threads check for available work in queue
    - Usually remove work in FIFO fashion
    - If no work, thread waits until queue is not empty
- When have work to do...
    - Add work to queue and return

Keeping Threads Around...

- Thread Pools
  - Basically an array of threads that sticks around
  - Simple, but causes blocking
- Work Queues
  - Adds a queue of "work" (runnable objects)
  - More complicated, but responsive

Resources

Java Theory and Practice: Thread Pools and Work Queues

Brian Goetz on IBM Developer (2002)

~~https://www.ibm.com/developerworks/library/j-jtp0730/~~

Introduction to Java Threads

Brian Goetz on IBM Developer (2002)

https://developer.ibm.com/tutorials/j-threads/

Package java.util.concurrent.*

- Includes thread pool and work queue implementations
- Includes thread-safe data structures
- Related packages also include:
    - Read/write lock implementations
    - Atomic versions of Boolean, Integer, etc.

https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/util/concurrent/package-summary.html

USF UNIVERSITY OF SAN FRANCISCO

CHANGE THE WORLD FROM HERE

Software Development
Department of Computer Science

Professor Sophie Engle
sjengle.cs.usfca.edu