Advanced Topics in Functional and Reactive Programming

Majeed Kassis

Syllabus

- Introduction to Concurrency
  - Introduction to Concurrency
  - Synchronization Mechanisms
  - Liveness, Deadlock Causes and Prevention
- Distributed Systems and Concurrency
  - Distributed Systems
  - Concurrency Control in Distributed Systems
  - Synchronization in Distributed Systems
- Lock-Free Concurrency
  - Synchronization Hardware
  - Immutability in Concurrency
- Concurrency Tools
  - Java Concurrency Tools
  - C++ Concurrency Tools
  - JavaScript Concurrency Tools
  - Python Concurrency Tools
  - Debugging Multithreaded Applications
- Asynchronous Programming
  - Actor Model and Event Driven Programming
  - Asynchronous Programming
  - Python Asynchronous Frameworks
  - JavaScript Asynchronous Frameworks
- Reactive Programming
  - Reactive Programming
  - Functional Reactive Programming
  - Reactive Frameworks
  - Reactive REST Case Studies

Introduction to Concurrency

- Introduction to Concurrency
  - Basic concepts
  - Types of concurrency
  - Difficulties in concurrency
- Synchronization Mechanisms
  - Locks
  - Mutual Exclusion
  - Semaphore
  - Monitors
- Liveness, Deadlock Causes and Prevention
  - Deadlock causes
  - Deadlock avoidance and solutions
  - Starvation – causes, prevention (scheduling)

Distributed Systems and Concurrency

- Distributed Systems
  - Basics
  - State and time
    - Physical clocks
    - Event ordering – logical clocks
- Concurrency in Distributed Systems
  - Causes
  - Deadlocks
- Synchronization in Distributed Systems
  - Distributed Mutual Exclusion
    - Token based
    - Centralized
    - Contention based
    - Election based

Lock-Free Concurrency

- Synchronization Hardware
  - Atomic Operations
  - CPU Instructions: Test-and-Set, compare-and-swap, fetch-and-add
  - Spin-locks: busy-wait locks
- Immutability in Concurrency
  - Immutable Collections – data structures
  - (that have) Mutative API – allows add/remove/update!
  - (which are) Persistent - preserves the previous version of itself when modified
  - (and implemented with) Maximum Sharability and Runtime Efficiency

Concurrency Tools

- Java Concurrency Tools
  - Executors, Futures, CompletionServices
- C++ Concurrency Tools
  - std::async
  - std::future
- JavaScript Concurrency Tools
  - Generators
  - async, await, promises
- Python Concurrency Tools
  - Generators
  - Coroutines
- Debugging Multithreaded Applications

Asynchronous Programming

- Actor Model and Event Driven Programming
  - Actor Model
    - Application components are defined as Actors who communicate through mail
  - Even Driven Programing
    - Program flow is determined by the occurrence of events.
- Asynchronous Programming
  - Asynchronous Design Pattern
  - Event loop
  - Non-blocking IO
- Python Asynchronous Frameworks
  - AsyncIO
- JavaScript Asynchronous Frameworks
  - Node.js

Reactive Programming

- Reactive Programming
  - Streams, Stream Combinators, Reactive Streams
  - Observables – Sequences, Streams
- Functional Reactive Programming
  - Akka
- Reactive Frameworks
  - ReactiveX - RxJava, RxJS, RxCpp, RxPY
- Reactive REST Case Studies
  - Node Express – Reactive REST server
  - Java Spring - HTTP streaming over asynchronous servlet
  - ReactiveX use case - Reactive Client of multiple REST servers

Reactive Programming

- Reactive Programming
  - Streams, Stream Combinators, Reactive Streams
  - Observables – Sequences, Streams
- Functional Reactive Programming
  - Akka
- Reactive Frameworks
  - ReactiveX - RxJava, RxJS, RxCpp, RxPY
- Reactive REST Case Studies
  - Node Express – Reactive REST server
  - Java Spring - HTTP streaming over asynchronous servlet
  - ReactiveX use case - Reactive Client of multiple REST servers

Course Requirements

- Assignments – 60-70%
    - Question Answering – 15%
        - Small “research” assignments + questions to be answered
    - Programming – 15%
        - Small “research” assignments + programing assignments
    - Big Assignment – 30-40%
        - Incorporating reactive programming tools taught at class
- Exam – 30-40%
    - Yes, I know. But it’s in the syllabus.
- Late Submissions
    - You are allowed to be late (2 days maximum) **once** for free, beyond that
    - 3 days late will net you 6 points penalty (2 points loss per day)
        - Beyond 3 days, submission will not be accepted!
            - Only army reserve duty or hospitalization are accepted for extensions.

Research assignments?

- Each week you will receive links to guides for self reading
  - These links are guides for specific elements you need to research
- An assignment will be published asking specific questions, their answers can be found in these guides.
- No copypasta!
  - A block of text which has been copied and pasted from somewhere else.
  - An error in a software application caused by the copy-and-paste of erroneous code.
- Write the answers using your own words.

Timeline - March

Date	Class Material		Practical Session Material		Slides
Week 1	March 13	(1)Introduction to Course (2)Introduction to Concurrency	March 16	Synchronization in Java8 & C++14	15+31
	March 15	Synchronization Mechanisms			18
Week 2	March 20	Liveness, Deadlock Causes and Prevention	March 23	Distributed Mutual Exclusion	30
	March 22	(1)Introduction to Distributed Systems (2) MapReduce (3)Case Study – Google Cluster			10+?+ 11
Week 3	March 27	Distributed Systems – Clock Synchronization	March 30	Apache ZooKeeper vs Google Chubby	43
	March 29	*Distributed Systems – Global State			17

Timeline - April

Date	Class Material		Practical Session Material		Slides
Week 4	April 03	*Concurrency Control in Distributed Systems	April 06	Holiday	36
	April 05	Synchronization Mechanisms in Distributed Systems			32
Week 5	April 17	Holiday	April 20		
	April 19	Actor Model and Akka			
Week 6	April 24	More Akka	April 27		
	April 26	Event Driven Programming			

Timeline - May

Date	Class Material		Practical Session Material		Slides
Week 7	May 01	Holiday	May 04	Holiday	
	May 03	Functional Programming			
Week 8	May 08	Persistent Data Structures	May 11		
	May 10	Reactive Programming - RxJava			
Week 9	May 15		May 18		
	May 17				
Week 10	May 22		May 25		
	May 24				
Week 11	May 29		June 01		
	May 31	Holiday			

Timeline - June

Date	Class Material		Practical Session Material		Slides
Week 12	June 05		June 08		
	June 07	Student Day			
Week 13	June 12		June 15		
	June 14				
Week 14	June 20		June 23		
	June 22				
Week 15	June 26		June 29		
	June 28				