

Programming for Concurrency

15-213/18-243: Introduction to Computer Systems

25th Lecture, 22 April 2010

Instructors:

Bill Nace and Gregory Kesden

Today

- Limitations of iterative servers
- Process-based concurrent servers
- Threads-based concurrent servers
- Event-based concurrent servers

Concurrent Programming is Hard!

- The human mind tends to be sequential
- The notion of time is often misleading
- Thinking about all possible sequences of events in a computer system is at least error prone and frequently impossible
- Classical problem classes of concurrent programs:
 - **Races**: outcome depends on arbitrary scheduling decisions elsewhere in the system
 - Example: who gets the last seat on the airplane?
 - **Deadlock**: improper resource allocation prevents forward progress
 - Example: traffic gridlock
 - **Livelock / Starvation / Fairness**: external events and/or system scheduling decisions can prevent sub-task progress
 - Example: people always jump in front of you in line
- Many aspects are beyond the scope of 15-213 / 18-243