6.033 - Coping with Complexity
Lecture 1
Katrina LaCurts, lacurts@mit.edu

1. Introduction to Systems
   - What is a system?
   - Complexity makes building systems difficult

2. Why is Complexity Bad?
   - Limits what we can build
   - Causes problems
     - Emergent properties
     - Incommensurate scaling
     - Propagation of effects
  - Limiting what we can build affects how well we can get other good properties: scalability, fault-tolerance, security, performance, etc.

3. Mitigating Complexity
   - We mitigate complexity with modularity and abstraction
     - Modular systems are easier to reason about, manage, change, improve
     - Modularity reduces fate-sharing.
     - Abstraction lets us specify interfaces without specifying implementation
     - Good abstraction decreases the number of connections between modules
   - Common techniques for applying modularity and abstraction: layering, hierarchy

4. Enforced Modularity
   - Soft modularity isn't enough
   - One way to enforce is with a client/server model
     - Reduces fate-sharing
     - Important: remote procedure calls (RPCs) != procedure calls (PCs)
       - Have to deal with different types of failure