

Agenda

- **Last time (Tues)**
 - Networks (chpt 3)
 - Start Interprocess Communication (chpt 4)
 - 1st HW/PA out (due Thur Feb 15 @ 12:30pm)
- **This time**
 - Interprocess Communication (chpt 4)
- **Next time (Thurs)**
 - RPC/RMI (chpt 4/5)
- **Next Next time (Thurs evening 5-6:15 or Fri 3-4:15)**
 - RPC/RMI (chpt 4/5)
- **Note: we are skipping chpt 6**

CS451: Distributed Systems (Spring 2007)

Before we start: Assignment #1

- Questions from the book + programming
- IM client, without graphics (a little primitive, yes)
- We'll start an "errata/clarifications" web page

- <http://www.cs.rpi.edu/courses/sysprog/sockets/sock.html>
- Cygwin (get gcc, too)
- Emacs (from the class web page)

- Server: sunfire1.cs.virginia.edu:5451

CS451: Distributed Systems (Spring 2007)

Interprocess communication – how?

- **Primitives: *send* and *receive***
- **Types:**
 - Synchronous: The sending and receiving processes synchronize at every message, the send and receive operations are blocking.
 - Asynchronous: The send operation is non-blocking, the receive operation may be either blocking or non-blocking.
- **Queue associated with each message destination.**
- **Senders cause messages to be added to remote queues.**
- **Receivers remove messages from local queues.**
- **Issues**
 - Reliability.
 - Ordering.

CS451: Distributed Systems (Spring 2007)

Demo

- Compiling the example code from RPI on cygwin

CS451: Distributed Systems (Spring 2007)