

Slide 1

Administrivia

- Reminders:
Homework 7 code due today. Okay to turn in through Thursday without penalty.
Homework 8 design due Friday, code same day and time as final. (Or — combine design/final steps?)
Extensions only under very unusual circumstances.
- Tentative review sheet for final on Web. (But see next slide.)
- Questions about grading? (Grades and comments on homework coming by e-mail soon.)
- Office hours this week to be announced by e-mail.

Slide 2

More Administrivia

- Original plan is for the final to be similar to the midterm, but twice as long.
- Alternative: Individual project presentations (about 10 minutes each, worth 50 points), then exam slightly longer than midterm and worth 150 points. Which should we do? (This one. Updated review sheet on Web soon.)

Networking Basics

Slide 3

- Inter-computer communication based on layered approach and “protocols”:
 - Application level — HTTP, FTP, telnet, SMTP, POP, IMAP, NTP, etc., etc.
 - Transport level — TCP (Transmission Control Protocol), UDP (User Datagram Protocol).
 - Network level — IP (Internet Protocol — addressing, routing of packets).
 - Link level — device drivers, etc.
- Messages are routed to
 - A machine (“host”), identified by IP or name.
 - A process, identified by “port number” (16 bits). 0 — 1023 are “well-known ports”, others available for applications.

Networking Basics — TCP and UDP

Slide 4

- UDP — independent messages, no guarantees about reliability or message order — analogous to (snailmail) letter.
- TCP — point-to-point channel, guarantees reliability and message order — analogous to phone call. Endpoints called “sockets”.

Networking in Java

Slide 5

- Classes for communicating at application level — e.g., URL (“show URL” example).
- Classes for communicating at network level:
 - TCP — `Socket`, `ServerSocket`.
 - UDP — `Datagram*`.
- RMI (Remote Method Invocation).

Networking in Java — Sockets

Slide 6

- Client/server model:
 - Server sets up “server socket” specifying port number, then waits to accept connections. Connection generates socket.
 - Client connects to server by giving name/IPA and port number — generates a socket.
 - On each side, get input/output streams for socket.
- Simple example in binary-I/O program from last week.

Networking in Java — RMI

Slide 7

- Motivation — for client/server applications, can be annoying to have to design your own protocol.
- Instead, idea is to define “remote objects” that can be treated (at program level) like any other objects — invoke methods.
- Typical use in client/server program:
 - Server creates some remote objects and “registers” them.
 - Clients look up server’s remote objects and invoke their methods.
 - Both sides can pass around references to other remote objects.
- Dynamic code loading possible too.

Networking in Java — RMI, Quick How-To

Slide 8

- Define a class for remote objects:
 - Define interface that extends `Remote`
 - Define class that implements that interface, extends a Java “remote object class. Can also include other methods, only available locally.
 - Write code using classes — if using as remote object, reference interface; otherwise can reference class.
- Compile and execute:
 - Compile as usual, *plus* run `rmi` to generate “stubs” to be used in communicating with remote objects as remote objects.
 - Make classes network-accessible.
 - Start `rmiregistry`.
 - Run server and clients as usual.

More Networking Examples

- Sockets versus RMI: Java master/worker example from parallel programming class.
- Chat program.

Slide 9

Course Recap — What Did We Do?

- Java basics.
- Object-oriented programming — polymorphism, inheritance, etc. Not stressed much in class, but game is a good example of a non-trivial o-o design.
- Basic ADTs — stacks, queues, trees (sorted and heaps); different implementations (arrays versus dynamic data structures using references).
- Recursion review.
- Tour of the Java libraries — GUIs, graphics, I/O; a very little about threads and networking.
- A fairly large programming project involving using someone else's code.
- To get a sense of what you learned — compare what you knew in August to what you know now.

Slide 10

Minute Essay

- How did the course compare to your expectations/goals? Did you learn what you hoped to learn?

Slide 11