Outline

- Socket review
- URL classes
- Server programming example
- Homework #4
Outline

- Homework #3 solution
- Homework #4
- Socket review
- HTTP Protocol
- URL and URLConnection
- Example
Sockets Review

- For applications to communicate, they need
  - a transport mechanism
  - an addressing mechanism
  - way to interact with these mechanisms

- For client/server applications
  - client = caller, originates requests
  - server = listener, handles requests
Sockets Review II

- Transport mechanism
  - TCP/IP
  - bi-directional data stream

- Addressing mechanism
  - hostname
  - port number
Sockets Review III

- Java classes
  - Socket for client
  - ServerSocket for server
- But
  - A call to accept returns with a regular Socket when a client connects
Socket Review IV

- Multi-threading
  - Extremely important for server applications
  - One thread listens continuously for connections
  - Other threads handle requests
Server Programming Example

- WeatherServer
- Protocol
  - one line commands: ADD and INFO
  - one line responses
Homework #4

- Networked version of FFGUIMode
- In other words
  - turn FFGUIMode into a server
  - client already exists
Objectives

- Add server thread to FFGUIMode to listen for connections
- Add worker threads to execute commands
- Add a Server menu to interface
- Synchronize database access
New Package Structure

- core
  - Command, CommandMap and basic commands
- cmd
  - command implementations
- db
  - database
- ui
  - user interface
- client
  - a text mode client
- server
  - to be added
Client

- TextModeClient
- Looks like old FFTextMode
- but
  - instead of creating and executing command objects
  - it sends messages to a server
- What should those messages look like?
Homework #4 Protocol

- **Request #1**
  - prompts
  - command: name

- **Response #1**
  - OK
  - Enter first name:
  - Enter last name:

- **Request #2**
  - parameters
  - command: name

- **Response #2**
  - OK
  - first
  - last

- **Request #3**
  - name
  - first: Marilyn
  - last: Monroe

- **Response #3**
  - OK
  - Members with name: Marilyn Monroe
  - 54045 Monroe Marilyn 01/01/2003 75500 3

- **Request #4**
  - parameters
  - command: nmae

- **Response #4**
  - ERROR
  - Unknown command: nmae
Stateless Protocol

- Each request is a single connection
- Worker thread
  - handles a single request
  - then terminates
- Simplifies implementation
Alternatives

- **Session**
  - login
  - make text mode requests
  - logout
  - worker handles whole interaction

- **Multi-request**
  - transmit name of command
  - subsequent requests relative to command
  - worker handles whole command
Example

- Program as it should run
- Using telnet to test
URL classes
HTTP Protocol

- Request
  - “I want something”

- Response
  - “Here it is”
  - or “Not found”

- Headers

- Body
HTTP Response Example

HTTP/1.1 200 OK
Date: Fri, 25 May 2001 22:51:05 GMT
Server: Apache/1.3.4 (Unix)
Set-Cookie: visit=1102002283923049566;
path=/;
expires=Sat, 26 May 2001 22:51:05 GMT
Last-Modified: Mon, 21 May 2001 14:17:23 GMT
Accept-Ranges:bytes
Content-Length: 2000
Content-Type text/html

<HTML>
<HEAD>
<TITLE>All About Servers</TITLE>
</HEAD>
<BODY>
...

</BODY>
</HTML>
Stateless

- HTTP is stateless
- Each request is treated independently of others

Problems
- authorization
- identity
- “sessions”
URL

- Example
  - http://www.cti.depaul.edu/people/

- Parts
  - Protocol
  - Host
  - Port
  - Path
In Java

- URL classes
  - new URL (http://www.cti.depaul.edu/”)
- Simplest case
  - url.getInputStream()
  - read from downloaded document
URLConnection

- `url.getConnection()`
- with URLConnection
  - initially "unconnected" state
  - can modify request properties/headers
- `connect()` or `getInputStream()`
  - "connected" state
  - can read response properties/headers
Example