

CSI 32 LECTURE NOTES (Ojakian)

Topic 12: Networks

OUTLINE

(References: ch. 16)

1. Network background
 2. Client-Server examples
-

1. Networks: Background

- (a) Host name (or IP address)

***PROBLEM* 1.** *Find your IP Address.*

On windows: Go to command prompt. Type in "ipconfig". Then find the number corresponding to "IPv4 Address".

- (b) URL (Uniform Resource Location)
- (c) Protocols
- (d) Client-Server
- (e) Peer-to-Peer

2. Simple Client Example: Fetch time

PROBLEM 2. *Write a client program to get the official world time from the host `time.nist.gov`. using port 13 (no send required; just recv).*

- (a) Terminology: sockets, ports
- (b) Socket Methods:
 - i. `connect`: needs IP address and a port.
 - ii. `send`: typically needed for clients, to request info from server.
 - iii. `recv`: to receive input from server (input maximum number of bytes to receive)
- (c) String/Byte conversions.
 - i. `encode`: String to Byte
 - ii. `decode`: Byte to String

3. Server

(a) Module `socketserver`

- i. `TCPServer`: Instead of `socket`
- ii. `BaseRequestHandler`: To process requests from clients

(b) Using `BaseRequestHandler`

- i. Meant to be parent class of your child class
- ii. At least define `handle` method in your child class
- iii. `self.request`: socket created with client.

(c)

PROBLEM 3.

- i. Create a client that sends a message to the echo server, gets a response, and then prints out this response.*
- ii. Create a server on your own machine that waits for a message and then simply sends it back (i.e. an “echo server”).*

PROBLEM 4. *Modify the echo server and client to do something else that is interesting, and clearly state the sending protocol.*

4. Your Server!

***PROBLEM* 5.** *Create a server with your IP Address which sends back some kind of interesting response to our client. Let me know the IP Address and the sending protocol, then I'll send a message (for the class to see)!*