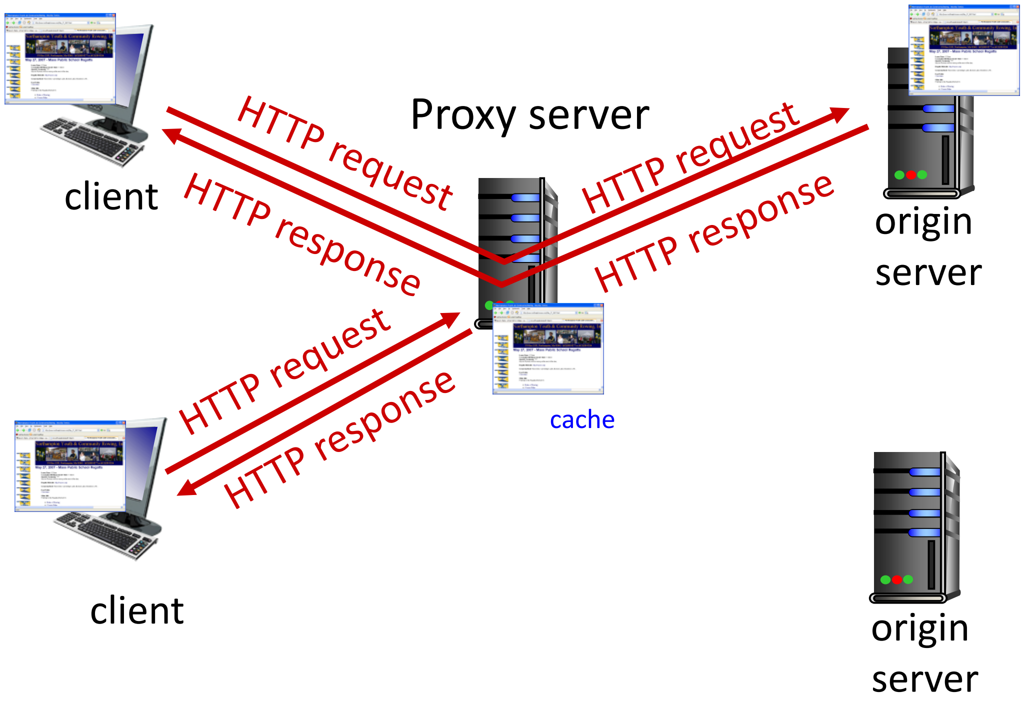
Homework #2

- □ Objective
  - ▪ Implementing an HTTP proxy

- □ user sets browser: Web accesses via cache
- □ browser sends all HTTP requests to cache
  - ▪ If object in cache: cache returns object
  - ▪ Else: cache requests the object from origin server, then returns it to client



Homework #2

□ Details

- Accept an absolute HTTP requests from clients
    - GET http://www.cs.princeton.edu/index.html HTTP/1.0
- Forward requests to origin servers (with relative URLs)
    - GET /index.html HTTP/1.0
      Host: www.cs.princeton.edu
      Connection: close (need to add this header or change to this if needed)
      (other headers can be just forwarded)
- Return response data to a client
- An invalid request from the client should be answered with an error code, i.e. "Bad Request" (400) or "Not Implemented" (501) for valid HTTP methods other than GET
- Should run using a command like “proxy <port number>”
- Languages : python, c, c++, Java

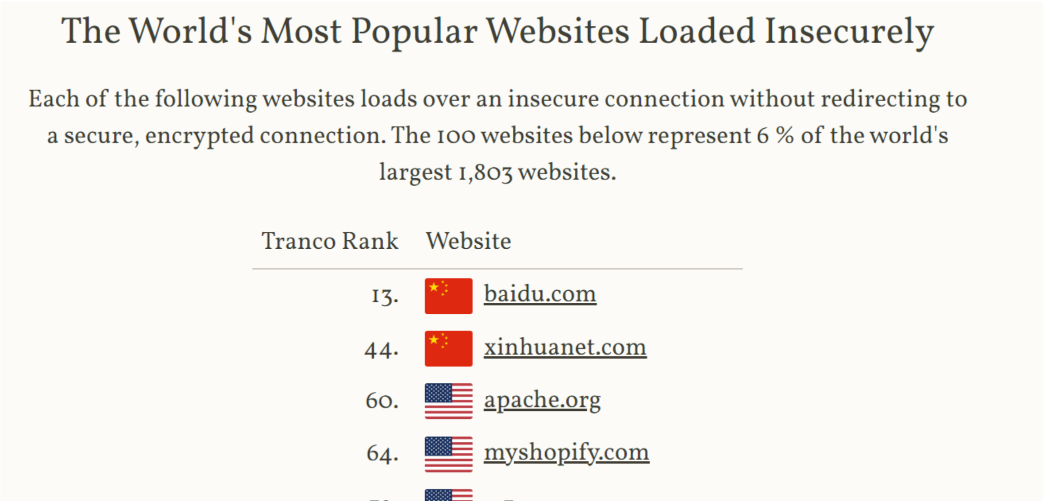
Homework #2

- Testing
  - Using Telnet
    - telnet localhost <port>
    - Trying 127.0.0.1...
    - Connected to localhost.localdomain (127.0.0.1).
    - Escape character is '^]'.
    - GET http://www.gnu.org / HTTP/1.0
    - (HTTP response message forwarded by the proxy should be printed in text)

Homework #2

□ Other details

- You don’t need to implement HTTPS
- Many sites provide only HTTPS, so for testing refer to https://whynohttps.com/



The World's Most Popular Websites Loaded Insecurely

Each of the following websites loads over an insecure connection without redirecting to a secure, encrypted connection. The 100 websites below represent 6 % of the world's largest 1,803 websites.

Tranco Rank	Website
13.	🇨🇳 baidu.com
44.	🇨🇳 xinhuanet.com
60.	🇺🇸 apache.org
64.	🇺🇸 myshopify.com

Homework #2

- □ Submission
  - ▪ Deadline : Dec 12 (midnight)
  - ▪ By Blackboard
  - ▪ Submit a zip file with the file name of your name
    - A report
      - ✓ Environment (hardware, operating system, used libraries)
      - ✓ Compilation commands, if needed
      - ✓ Implementation details
    - Source codes
- □ Policy
  - ▪ If any plagiarism is found, you will get 0 and more penalty
  - ▪ Evaluation
    - Will be based on the correctness (using telnet)
    - If you submit late, you will get -20% penalty per day

Homework #3

- □ Objective
  - ▪ Improving the HTTP proxy
- □ Details
  - ▪ Save the objects received from the origin server into files, and serve future requests using them
  - ▪ I/O multiplexing
    - Making your proxy server handle multiple requests simultaneously
    - You can use select that we covered in class or other mechanisms
  - ▪ Should run using a command like “proxy <port number>”
  - ▪ Languages : python, c, c++, Java

Homework #3

- □ Testing
  - ▪ Using browser
    - Setup the proxy server in browser by referring to https://www.digitalcitizen.life/how-set-proxy-server-all-major-internet-browsers-windows/ or other web sites
    - If you run the proxy server locally, then set the server address to 127.0.0.1

Homework #3

- □ Submission
  - ▪ Deadline : Dec 19 (midnight)
  - ▪ By Blackboard
  - ▪ Submit a zip file with the file name of your name
    - A report
      - ✓ Environment (hardware, operating system, used libraries)
      - ✓ Compilation commands, if needed
      - ✓ Implementation details
    - Source codes
- □ Policy
  - ▪ If any plagiarism is found, you will get 0 and more penalty
  - ▪ Evaluation
    - Will be based on the correctness (multiplexing/browser operation)
    - If you submit late, you will get -20% penalty per day