CS 4390/5390 Kresman Homework 4

Learning Outcomes

* Understand client-server communication; use TCP/IP primitives for client-server communication
* Explain and distinguish the various service types supported by internet applications – for example, best effort, streaming
* Understand underlying protocol format and interactions

ABET Outcomes

* CS 2 Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline
* CS 6 Apply computer science theory and software development fundamentals to produce computing-based solutions

**Task** 1: Implement a Web proxy

* A web proxy, typically maintained inside the enterprise, acts as an intermediary b/w the browser and web server caching web pages – requests served from cache (if available) and reduce server hits. An example below:
  + Suppose the proxy runs on port 12345, and web server on 23456, and server has the file abc/xyz. Launch Chrome w a url some like: http://localhost:12345/localhost:23456/abc/xyz. The proxy (@ 12345) receives a request: GET /localhost:23456/abc/xyz ...
  + The proxy parses the string and looks for the file abc/xyz locally. If found, send to client. Else, ask web server@ 23456 for the file (connect w server; send a request some like: GET http://" + abc/xyz + " HTTP/1.0\r\n\r\n”), receive it/cache locally before sending to client. Either case, client (Chrome) displays the file. Note also the possibility of file not found at server.
* Checkout USBWebserver (https://www.usbwebserver.net/webserver), a free web server one can download/install on the flash drive. Due to ITS security issues, the proxy and the web server run on the local host.

**Task 2**: Implement a SMTP client

* SMTP server is smtp.bgsu.edu port # 25/tcp. The client is possibly on another host on bgsu domain. The sender email address is yours (do not impersonating anyone else - may be liable for criminal prosecution, read BGSU student code of conduct).
* serverName, portNO, senderEmail, rcvEmail, emailText emailFileAttachement are all given as command line arguments.
* build a smtpClient class, some like below, and call its methods to send the email.

Class smtpClient { //the return value & parameters are just informational - you decide what they need to be.

public String smtpClient (String serverMachine, int portNumber)

public String senderAndReceiver (String sender, String receiver)

public String messageBody (message)

public int endTheSession ()

}

Notes

* (Thanks to Tom,) Group size is up to 4 - sorry, no brownie points if the group size is < 4. CS 5390 group size is 1.
* Use any language of your choice. Use standard library calls, not object files from others or ‘special/task-specific’ libraries. If you find internet resources, credit the source, and make significant functional modifications - you are graded for your work, not someone else's! The code cannot include any icing! For both tasks, cout what is sent/received so one can follow the sequence/progress, thanks.
* SMTP RFC 821 http://www.faqs.org/rfcs/rfc821.html - read it carefully first so you know the smtp interaction. [Command line client interaction - telnet smtp.bgsu.edu 25 may be of help in your comprehension, but NOT a substitute for the client you write.]
* Resources: Units 2-5. See also the class C++/python/Java code in “client-server code samples from class.”

Canvas submission

* One submission/group. A zip file w 3 items: smtpClient.txt, webProxy.txt, and a runSnapshot.doc word file (see sample).
* In addition, each student submits a ‘Self and Peer Eval’ (see sample), that reflects contribution of each group member on both tasks.

Grading Rubric (20 points each)

|  |  |
| --- | --- |
| Task 1 | Task 2 |
| \_\_\_/6 Command line parameters  \_\_\_/10 Correct implementation w email transmission/reception shown  \_\_\_/4 Individual contribution (based on peer eval)  Deduction: 8 points for missing source code listing | \_\_\_/6 2 each for correct reporting of at least 3 HTTP codes (400, 401, 403, 404)  \_\_\_/10 Correct implementation and browser display  \_\_\_/4 Individual contribution (based on peer eval)  Deduction: 8 points for missing source code listing; 3 points for missing documentation |

.