CN Programming Assignment- Simple Web Proxy Server

The purpose of this assignment is to learn about Web proxies and the HyperText Transfer Protocol (HTTP). This is one of your major assignments in the Computer Networks course. Form a group of two students of your choice and **post it on moodle**.

You will implement and test a simple Web proxy in C/C++/Java. This Web proxy performs only one role (proxying) but not caching. The most important HTTP command for your Web proxy to handle is the "GET" request, which specifies the URL for an object to be retrieved. In the basic operation of your proxy, it should be able to parse, understand, and forward to the Web server a (possibly modified) version of the client request. Similarly, the proxy should be able to parse, understand, and return to the client a (possibly modified) version of the response that the Web server provided to the proxy. Your proxy should be able to handle response codes such as 200 (OK) and 404 (Not Found) correctly, notifying the client as appropriate.

You will need at least one TCP (stream) socket for client-proxy communication, and at least one additional TCP (stream) socket for proxy-server communication. Your proxy should support multiple concurrent HTTP transactions, so you need to fork child processes or use threads for request handling. Each child process or thread will use its own socket instances for its communications with the client and with the server.

You should be able to compile and run your Web proxy on any lab machine, or even your personal machine. You should be able to use your proxy from any Web browser (e.g., Internet Explorer and Mozilla Firefox), and from any machine (either on campus or at home). To test the proxy, you will have to configure your Web browser to use your specific Web proxy (e.g., look for menu selections like Edit, Preferences, Advanced, Proxies).

As you design and build your Web proxy, give careful consideration to how you will debug and test it. For example, you may want to print out information about requests and responses received and processed. Once you become confident with the basic operation of your Web proxy, you can toggle off the verbose debugging output. You can also use tools like **wireshark** to collect network packet traces. By studying the HTTP/TCP packets going to and from your proxy, you can convince yourself that it is working properly.

Your proxy server should able to communicate with Multi-threaded web server that you had developed as part of Programming Assignment 1. While testing, you should have client (web browser or your own client application) running on PC1, proxy on PC2, and Multi-threaded web server on PC3.

In your testing of the proxy, you may want to go through incremental steps similar to the following:

- Download a small ASCII text file such as IETF RFC on HTTP 1.1 (http://www.ietf.org/rfc/rfc2616.txt)

CN Programming Assignment- Simple Web Proxy Server

- Visit simple HTML pages such as http://www.iith.ac.in/~tbr/teaching.html
- Visit a modest Web page with a few embedded objects such as www.iith.ac.in
- Visit a more complicated Web page such as http://www.espncricinfo.com/

The primary test of correctness for your proxy is a simple visual test. That is, the content displayed by your Web browser should look the same regardless of whether you are using your Web proxy or retrieving content directly from the Web server.

Documentation should include a README file explaining what is your project and how you are implementing various major modules of the projects, and names of various files (along with their purpose) created for the project.

50% of marks are allotted for Documentation.

Useful links:

- http://en.wikipedia.org/wiki/Proxy_server
- HTTP 1.1 RFC: http://tools.ietf.org/html/rfc2616
- http://web.mit.edu/6.033/2000/www/lab/webproxy.html
- Web Proxy: http://yuba.stanford.edu/vns/assignments/web-proxy/
- Sockets Tutorial: http://www.linuxhowtos.org/C_C++/socket.htm
- Beej's Guide to Network Programming Using Internet Sockets: http://www.beej.us/guide/bgnet/
- Programming UNIX Sockets in C - Frequently Asked Questions: http://www.softlab.ntua.gr/facilities/documentation/unix/unix-socket-faq/unix-socket-faq.html