

## CS78 Spring 2006

**Due Date: 28<sup>th</sup> April 2006**

### Programming Lab 2: Building a Web Server

For this assignment you will construct a multi-threaded web server. You will implement a web server capable of serving multiple requests from clients, in parallel, but with each request/response pair using a separate TCP connection (as was specified in HTTP/1.0, RFC 1945).

You are to implement this in java but just as the prior assignment you are not to use any packages that abstract away the details of socket programming. As an example do not use the `java.net.URL` package to assist in parsing the incoming URL request, use instead standard java string handling packages.

You have the freedom to construct the application in any manner of your choosing as long as it meets the requirements specified in document. We suggest the following structure. Construct a primary main thread that listens to incoming client connections. Then have this thread satisfy each request by creating a separate thread that handles a specific client request until completion of the request.

Your web server should accept two configuration parameters: one for the TCP port on which it listens, and the other for a directory, from which requests for files are satisfied. You are required to handle the following file types of: jpeg, gif, text, and html. Your web server should be compatible with one of the mainstream standard web browsers such as: Internet Explorer, Firefox, and so on.

You do not have to implement the full HTTP standard, for instance you can reply with a single standard error response of your choosing when an error occurs (e.g., the 404 error response). Note, that whichever response you select it must be from the HTTP specification and the response must be well formed.

Hand in via email ([niclane@cs.dartmouth.edu](mailto:niclane@cs.dartmouth.edu) prior to the assignment deadline) the following files:

1. Your source code. Please document your code.
2. A ethereal trace that demonstrates your server satisfies a web client requests coming from a mainstream standard web browser. The trace should contain:
  - I. A failed request where the URL is not available (for instance if the file is not available).
  - II. A successful request for a html page containing a hyper link to a text file and two embedded images one of type gif and one of type jpeg.

III. A successful request for the text file which is hyper linked from the html page used in section II.

3. A screen shot of the html page from 3.II rendered in the browser.

## Suggestions & Comments

1. You can debug your web server using a telnet client in the initial stages.
2. Don't use the standard HTTP port 80. Please make this a configurable setting for your server. This means you will need to specify this port when using a web client on your machine
3. Please review the HTTP ethereal lab and the K&R text book regarding HTTP
4. You do not need to implement cookie and other extended aspects of the basic HTTP standard.

## References

Java Network Programming

- <http://java.sun.com/docs/books/tutorial/networking/overview/index.html>
- <http://java.sun.com/docs/books/tutorial/essential/threads/>
- <http://java.sun.com/docs/books/tutorial/networking/sockets/index.html>

RFC 1945: HTTP 1.0

- <http://www.w3.org/Protocols/rfc1945/rfc1945>