

Project 1: Implementing an HTTP Server

Prerequisites

- 1) The project is to be completed **individually**. No late submissions.
- 2) You can use any programming language you want as long as it is your code.
- 3) If you must use an open-source code, make sure to provide a reference and indicate where you used it. **Note that you should be able to complete this project solely using your own code.**
- 4) Read and study <http://beej.us/guide/bgnet/>.

Submission

Create a single zip file that includes your code and sample output, name the file as your_lastname_project1.zip and upload it to LMS.

Demo

We will have the demos during the class.

Problem to Solve

Single-Threaded HTTP Server

Implement a simple HTTP server using the original HTTP as defined in 1991 (<http://www.w3.org/Protocols/HTTP/AsImplemented.html>) which is referred to as HTTP 0.9 and simply provides one object per request.

The server is to be configured via the console taking two parameters, i.e., the TCP port where the server will run on and the document root from which the objects are served. Note that:

- HTTP 0.9 does not include error codes. Signal any error (e.g., when the requested file does not exist) to the client by closing the connection.
- If the client requests a directory, the server responds with the index.html of the requested directory. If this file does not exist, the server closes the connection.
- HTTP 0.9 only supports text but modern browsers also support binary data such as images. Hence, also images can also be requested from the server.

Multi-Threaded HTTP Server

Extend the server above such that a separate thread handles each request. Additionally, keep a log of all requests (i.e., timestamp, request, IP address and port of the client) which is output to the console every five seconds by yet another thread.

Example output:

2017-11-01 10:40:10 GET /index.html 192.168.1.1:44239

2017-11-01 10:40:13 GET /images/logo.gif 192.168.1.2:44239