

Build a very simple HTTP 0.9 compatible web server. The only request method supported is GET. The only MIME types supported are text/html, text/css, application/js, image/jpeg, image/png, and image/gif. The web server will **not** support any type of CGI (slow or fast). The web server will **not** support folder indexing.

The server will listen on TCP port 8181. You **must** pass on command line parameter to your server, the document root (folder) for serving files.

The web server will implement a pre-standard version of HTTP as described by Sir Tim Berners-Lee in <http://www.w3.org/Protocols/HTTP/AsImplemented.html>

The following HTTP status codes are returned by the server:

- 200 OK
- 400 Bad Request
- 404 Not Found
- 414 Request-URI Too Long
- 501 Not Implemented

Code	When to return
200	Returned when the requested file is successfully served
400	Returned when the request is malformed and cannot be fulfilled
404	Returned when the requested file cannot be found
414	Returned when the request length exceeds the input buffer for requests
501	Returned for any type of request other than GET

Version 0.9 of HTTP was never documented as an IETF RFC and is not well supported in modern browsers. Therefore, your web server will lie and claim to be a HTTP 1.0 server in its response headers.

The following HTTP response headers should be returned with each page served:

```
HTTP/1.0 200 OK
Connection: close
Content-Type: text/html
Content-Length: xx
Server: cs360httpd/1.0 (Unix)
```

The status code in the HTTP headers (“200 OK” above) will be different depending on whether the server is able to successfully serve the file or not. The “xx” shown above in the “Content-Length” header should be replaced with the number of bytes the server is sending (not including the headers). The HTTP headers **must** be followed by one blank line to separate the headers from the content.

The simple web server in this assignment has ***no file security***. Security must be built into any version you plan to use on a production network. File ACLs, setuid(2), setgid(2), and chroot(2) may be used to restrict the files accessible via the web server.

Your programs (client and server) must be developed in C or C++ using the BSD socket API on either Linux or BSD. The programs must compile and run on the class server (BSD) with gcc or g++. The class server (cs-srv-01) is accessible from the campus network and campus wireless.

Due date:

Completed server - 04/08/15

Program evaluation:

See the evaluation sheet for programming assignments on the course web site. Your program must meet all of these criteria in order to receive full credit. Your programs (client and server) should be submitted in source form in either a single zip, tar, or gzipped tar file. Your submission must include **all** files (headers, source files, makefile, etc.) required to successfully build your programs. I will enable all warnings and diagnostics and expect your code to compile cleanly without warnings.

Create and include a Makefile with your submission. See `make(1)` for details.

Hints:

Test your server by using a browser such as `lynx(1)`, Firefox, or Chrome.

Testing your server on the same machine is possible using the `lynx(1)` browser installed on the class server. Use multiple terminal windows or a virtual terminal emulator to run your server and clients on separate terminals. `tmux(1)` is a terminal multiplexer that allows you to have multiple terminals in a single screen. Use `localhost` or `127.0.0.1` as the host. Don't forget to check your program over a network connection.

`tcpdump(8)` and `gdb(1)` are useful tools to track down unexpected problems.

Additional resources:

Sample code - Textbook pages 33-36

man pages - `socket(2)`, `accept(2)`, `bind(2)`, `listen(2)`, `send(2)`, `recv(2)`

man pages - `signal(3)`, `getaddrinfo(3)`

man pages - `select(2)`, `tmux(1)`, `lynx(1)`, `nc(1)`, `tcpdump(1)`, `gdb(1)`

TCP - RFC 793, RFC 1122

Domain names - RFC 1035

HTTP 1.0 - RFC 1945

<http://beej.us/guide/bgnet/>

—

<http://sourceware.org/gdb/current/onlinedocs/gdb.html>

<http://www.gratisoft.us/todd/papers/strncpy.html>

<https://buildsecurityin.us-cert.gov/daisy/bsi/articles/knowledge/coding.html>

<http://www.di.uniovi.es/~cernuda/noprogramming.html>