Simple TCP/IP web server

This program implements a (very) simple web server.

Programming Issues

The networking issues are actually fairly trivial. HTTP is essentially a text-based protocol which runs over TCP/IP (though it can run over any transport protocol). The essential network steps are:

- Create a listening socket
- Accept a connection with it
- Fork a child process to service the connection, whilst the parent process goes back to accept more connections.
- Read in the HTTP request
- Send the HTTP response
- Send the entity requested (e.g. an HTML document)

The bulk of the code involves interpreting the HTTP request and sending the HTTP response according to the protocol (though this server is by no means 100% compliant with any HTTP protocol). In order to effectively facilitate communication, HTTP requests and reponses must follow a strict convention.

HTTP requests can be *simple* or *full*. A simple request contains one line only, and looks like this:

```
GET /index.html
```

A full request can contain more than one line, and a blank line **must** be sent at the end to signify the end of the request:

```
GET /index.html HTTP/1.0
Host: www.paulgriffiths.net
User-Agent: Lynx/2.8.1rel.2 libwww-FM/2.14
Accept-Encoding: gzip, compress
Accept-Language: en
<BLANK LINE>
```

The number of headers following the *request line* is variable, which is why the blank line is needed to let the server know when they have all been transmitted.

The server must decipher this request, and make an appropriate response. Presuming the above file exists, the HTTP response generated may look like:

```
HTTP/1.0 200 OK
Server: PGWebServ v0.1
Content-Type: text/html
<BLANK LINE>
```

Again, a blank line must be sent to signify where the headers end.

Following the response, the *entity* (e.g. an HTML document, a JPEG file) is transmitted. Once this is complete, the connection is terminated. With a simple HTTP request, only the entity is sent; no HTTP

response is generated.

Usage

This program has been written and tested on Redhat Linux 6.0. It should port without modification to other UNIX systems (with the possible exception of the makefile), but will not generally work on other platforms.

To keep things simple, the feature list of this server is fairly restricted. For example:

- There are no configuration files; all options are hard-coded
- The server can only return HTML documents (actually the server will return any file, but the HTTP response will specify the content-length as "text/html"
- There is no support for CGI or SSI
- No log files are produced.
- Importantly, with the exception of basic C error checking, **there are no security measures included whatsoever!** This program was written for basic instructional and demonstration purposes. Do **not** run it on any machine connected to an external network.

Source and Downloads

- View the C source in HTML format
- Download the tarred, gzipped source
- Download the zipped source

Previous: Simple TCP/IP time client

Home > Computer programming > C examples > Sockets examples > Simple TCP/IP web server

Navigation menu

- Home
- Site map
- iPhone/iPad Apps
- Programming
- Web development
- About me
- Blog

Search this site

Search

Conformance

- Valid XHTML1.0
- Valid CSS3