ECE435: Network Engineering – Homework 2
http, HTML, webserver

Due: Thursday, 14 September 2017, 12:30pm

1. **Write a simple webserver** (7 pts)

   The first part of this homework is to create a sockets program that acts as a simple webserver.

   (a) Download the code from:
   `http://web.eece.maine.edu/~vweaver/classes/ece435/ece435_hw2_code.tar.gz`

   (b) Unpack the files:
   `tar -xzvf ece435_hw2_code.tar.gz`

   (c) Build the C files:
   ```
   cd ece435_hw2_code
   make
   ```

   (d) You can use your code from HW#1 as a basis, or you can use the provided `webserver.c` file. If you use your own code, just be sure it is copied over `webserver.c`

   (e) Modify the code so it listens on port 8080.

   (f) When a connection comes in, read the entire request from the file descriptor into a char buffer. You can ignore most of the request; the important thing to look for is the `GET` command.

   (g) Find the line with `GET` in it and obtain the filename that is being requested (it follows immediately after GET). The `strstr()` function may be of use here.

   (h) Open the requested file. If it exists, open the file and write it out to the file descriptor. You can use `open()` and `read()`. Or you can use the buffered `fopen()` and such, but in that case you may want to use the `fdopen()` function.

   You will also need to send proper http headers. Note: to get the date/time for the Date: field you can look into the `time()`, `gmtime()`, and `strftime()` functions. You can find the last-modified time for a file as well as the size of the file (for `Content-Length:`) using the `stat()` function. For Content-Type you can look at the extension on the requested file. Some common content MIME types: .html = "text/html" .txt = "text/plain" .png = "image/png" .jpg = "image/jpeg"

   To create the buffer that you `write()` to the file descriptor you might find the `sprintf()` function to be useful.

   After the header, just write the contents of the file.

   Note that the http specification calls for carriage-returns and linefeeds (\r\n), not just linefeeds (\n).

   A sample header is shown below:

   ```
   HTTP/1.1 200 OK\r\n
   Date: Fri, 08 Sep 2017 04:56:25 GMT\r\n
   Server: ECE435\r\n
   ```

```
Last-Modified: Fri, 08 Sep 2017 04:31:47 GMT\r\n
Content-Length: 100\r\n
Content-Type: text/html\r\n
\r\n
```

Followed by the file contents.

(i) If the requested filename is not found, return a 404 not found error.

```
HTTP/1.1 404 Not Found\r\n
Date: Fri, 08 Sep 2017 04:56:25 GMT\r\n
Server: ECE435\r\n
Content-Length: 100\r\n
Connection: close\r\n
Content-Type: text/html; charset=iso-8859-1\r\n
\r\n
<html><head>\r\n
<title>404 Not Found</title>\r\n
</head><body>\r\n
<h1>Not Found</h1>\r\n
<p>The requested URL was not found on the server.<br />\r\n
</p>\r\n
</body></html>\r\n
```

(j) Log all requests received to stdout (just use `printf()`). Print the name of each file requested.

(k) Be sure to comment your code!

(l) To test, start the server and connect a web browser to
`http://localhost:8080/test.html`
and the webpage should appear.

You can use any browser and it should work. If you're remotely logging into a Pi or similar you can install and use the text-mode `lynx` browser.

If you have your test system on the network you can possibly connect from a remote machine if you know the IP address. Sometimes things like firewalls can get in the way though.

2. **Answer the following questions in the README file** (2pts)

(a) You use a browser to try to connect to a webserver.

i. The webserver returns error code 404. What is wrong?

ii. The webserver returns error code 418. What is wrong?

iii. The webserver returns error code 451. What is wrong?

(b) You use telnet to connect to a web server, and this is the top part of the results you get:

```
telnet www.maine.edu 80
Trying 130.111.33.23...
Connected to umsweb2013lb.its.maine.edu.
Escape character is '^]'.
GET / HTTP/1.1
Host: www.maine.edu
```

```
HTTP/1.1 200 OK
Date: Fri, 18 Nov 2016 21:50:22 GMT
Server: Apache/2.2.22 (@RELEASE@)
X-Powered-By: W3 Total Cache/0.9.4.1
Last-Modified: Fri, 18 Nov 2016 21:10:14 GMT
Expires: Fri, 18 Nov 2016 22:10:14 GMT
Pragma: public
Cache-Control: max-age=1192, public
Etag: 997a240b2b6a53d6c5e8b3bdb5dc782f
X-Pingback: http://www.maine.edu/xmlrpc.php
Vary: Accept-Encoding
Content-Type: text/html; charset=UTF-8
Set-Cookie: avr_3786590994_0_0_4294901760_400191362_0=2028572619_28159379;
max-age=300;path=/
Transfer-Encoding: chunked

4000

<!DOCTYPE html>
<!--[if lt IE 7 ]>            <html lang="en" class="no-js ie6"> <![endif]-->
<!--[if IE 7 ]>               <html lang="en" class="no-js ie7"> <![endif]-->
<!--[if IE 8 ]>               <html lang="en" class="no-js ie8"> <![endif]-->
<!--[if IE 9 ]>               <html lang="en" class="no-js ie9"> <![endif]-->
<!--[if (gt IE 9)|!(IE)]><!--> <html lang="en" class="no-js"> <!--<![endif]-->
<head>
<meta charset="utf-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
...
```

i. What web-server is the maine.edu website running?

ii. Can you directly connect to a HTTP/2 server with telnet as we did here with a HTTP/1.1 server? Explain.

3. **Something cool** (1pt)

Modify the provided test.html to be more interesting. Do one or or more of the following:

(a) Change the title to "ECE435 Homework 2"

(b) Add a size-1 header, centered, saying "ECE435 Homework 2"

(c) Put your name, as a size-2 header, centered.

(d) Have a horizontal rule (horizontal line)

(e) Have a clickable link to a website of your choice (although, keep it safe for work please). Have some surrounding text that briefly describes it, and bolds or italicizes at least one word.

(f) Add colors

(g) Add a table

(h) Add an ordered list

(i) Go overboard and have Javascript

4. **Submit your work**

- Please edit the README file to include your name.
  Also put your answers to the questions there.
- Run `make submit` which will create a `hw2_submit.tar.gz` file containing `README`, `Makefile`, `webserver.c` and `test.html`.
  You can verify the contents with `tar -tzvf hw2_submit.tar.gz`
- e-mail the `hw2_submit.tar.gz` file to me (vincent.weaver@maine.edu) by the homework deadline. Be sure to send the proper file!