Write Up

Testing:

The networking portion was tested using curl with various different ip addresses and ports such as port 80, port 8080. Ip addresses and hostnames tested included localhost, my laptop hostname, 127.0.0.1, 127.1.2.3, 127.25.25.25.

The program was tested using curl with valid commands such as

GET: curl -v -s http://127.1.2.3:80 --request-target ABCDEFarqdeXYZxyzf012345-ab

PUT: curl -v --request PUT http://127.1.2.3:80 --request-target 07------ijklmnopqrst4567890

And Invalid command

GET: curl -v -s http://127.1.2.3:80 --request-target ABC

And scenarios when GET ask for a nonexistent file, or when PUT wants override an existing file.

Questions

1. What fraction of your design and code are there to handle errors properly?

Too few or too many arguments would print an error message. Every network function has an if statement to check the function returned -1. Roughly almost half of the code consist of error handling.

2. How much of your time was spent ensuring that the server behaves “reasonably” in the face of errors?

The majority of the time was spent adding situational error handling if statements. Errors are handled by writing an http error code whenever possible, so that the server can continue running.

3./4. List the “errors” in a request message that your server must handle. What response code are you returning for each error?

Errors in a request message includes:

a. filename not size 27: 403

b. invalid filename characters: 403

c. ‘/’ : 403

d. none PUT or GET requests: 403

e. GET for a file that don’t exist: 404

5. What happens in your implementation if, during a PUT with a Content-Length, the connection is closed, ending the communication early?

read() will stop at where the communication ended. The code will continue creating the response header. Finally at send(), the error if statement would fail and err() will print: send() failed.

6. Does endianness matter for the HTTP protocol? Why or why not?

Endianness matter since everything must be in network byte order, which is what htol() does. Certain functions expect network byte order port arguments.