Httpserver Write Up

Jake Armendariz

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

About 1/3 of my code is built around error handling. I use one function to send all error problems, but I check most things in the code, using errno to assign the type of problem.

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

Hard to say, most of the error handling was built in to the design, as when I wrote anything, I tend to follow by an if statement checking that the assertions are true. But I would assume 33% of my time was around error handling (I specifically had problems with checking the repsponse was a key:value pair.

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

1. Invalid request (not get, put, head): 400 BAD REQUEST
2. Invalid http. Must be HTTP/1,1 : 400 BAD REQUEST
3. Cannot open file: 404 FILE NOT FOUND || 403 FORBIDDEN ? errno
4. filename contains special characters || len(filename) > 27: 400 BAD REQUEST
5. Content length missing in PUT request: 400 BAD REQUEST
6. Error while reading file/put data: 500 INTERNAL SERVER ERRORS

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

A 500 Internal server error with a half written file. But if you run another PUT request it without a connection close, then it will complete the job.

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

Yes, I assumed http tcp would handle it, but it seems to be extra safe the client and server are using the same endiness programmers should add ntohl, htonl, ntohs, htons to their code. This insures that the binary data is structured in the same way