Network Exercise 1

I.G.Batten@bham.ac.uk

Summary

- Write a small, multi-threaded webserver, and use appropriate test tools to check it works correctly.

Documentation

- RFC 2616, HTTP/1.1

- You might like to look at 7230–7235 (still in draft, but shows how the world has become more complex in 17 years)

Languages

- Java or C, but the Java option involves doing more functionality

- OS: needs to run on school Linux machines

  - You can almost certainly develop on OSX or Linux and port at the last if you want: this is highly unlikely to throw up significant differences in environment

Details

- C or Java: write a webserver capable of serving multiple documents in parallel, which gives correct status responses for file found or not found

  - Probably a directory of documents, but a database would work as well and be cool

  - C: sockets

  - Java: Network Stream library, but…

Details

- If you write the server in Java using network streams, you should also write an HTTP/1.1 client which will fetch files and print them on the screen.

- You don't need to do this if you write the server in C

  - using sockets and pthreads is significantly harder than network streams and "implements Runnable"

Conformance

- The server should correctly serve documents when tested against "wget" and "curl"

- The client, where applicable, should be able to fetch documents from www.bbc.co.uk or similar.

- Servers should listen on port 8088 (note, three eights).

Functionality and Marking

- 65%: working as described

- 35%: extra functionality (authentication, compression, content types, encryption ~~if you are mad~~), cool stuff (serving out of a database), particularly clean or tidy implementation, (client) rendering some HTML, etc, etc.

  - Wow!

Labs

- I will be available by appointment for the next two weeks

- I have booked the upper ground lab 12–1 Mondays for the next two weeks and will be there

Submission

- Via Canvas

- A tar, zip or cpio archive, containing at some point in the hierarchy a makefile which, when typing "make all" will build the binaries or Java executables from source, "make run" will run the server and "make test" will run the client (where applicable)

- I will provide a test harness later this week which will check your submission (ie, what I'm going to use for marking)

- Please test from other than localhost to ensure your code works remotely

Dates

- On Canvas today

- ~~Submissions by 23:59 Friday 28th~~

- Submissions by 23:59 Tuesday 1st Nov

  - Moved to give one extra lab session and to avoid clash with SSM exercise