

CS450: Introduction to Networking

Assignment 1

Prof. Jon Solworth

due dates:

Part I: 11 October 2004

Part II: 18 October 2004

Introduction

The assignment is about developing a rudimentary http server based on HTTP version 1.0 specified in RFC 1945. The server when complete should be a concurrent one servicing multiple requests simultaneously.

The assignment is split into two parts with different requirements and deadlines for each part. The first part deals with the http engine and the second part deals with the networking, as detailed below.

Your server should handle the operations `GET`, `HEAD`, and `PUT`. It should handle the header-fields: `Content-Length`, `Content-Type`, `Date`, `Expires`, `If-Modified-Since`, `Last-Modified`, `Server`, and `User-Agent`.

Part I

The HTTP server works on a request-response model. The requests come from the clients, which the server parses and responds appropriately. This part of the assignment is about building this HTTP Engine which given a request produces the appropriate response.

The request from the client, for this part of the assignment, is stored in a file. The name of the file with the client request is passed to the server as command line arguments. The server reads through the file and displays the appropriate HTTP response on the `stdout`.

The server does not have to deal with networking at this point. The server can quit once it displays the response.

You should turn in the listing, a test suite which demonstrates that your program works properly, and a description of any shortcoming of your program—that is, what doesn't work.

Note: In the hardcopy submitted for this part, please list out the set of protocol features (like message types, header fields etc.) implemented and the test cases and output to show they're working.

Part II

In this, the networking part of the server is added to the HTTP engine developed in Part I.

The HTTP server is a concurrent one. It receives requests from the client over a network connection, creates a new process/thread to service the latest request and kills the new process/thread once the response is sent to the client. That is the server uses a separate connection for each request.

The server keeps running till it is explicitly killed by the user. The server has to accept the host IP address and port number as command line arguments.

Whats due and when??

The turnin of the code is due before class on the dates mentioned above. And a listing of your code along with any test output and/or assumption etc., is due in class on the same day.

turnin

The turnin command is:

```
$ turnin -c cs450 -p a1.part# <cs-login>
```

The `part#` refers to the part number of the assignment(part1 or part2). To turnin your source and README files, store them in a directory with the same name as your cs login and turnin the entire directory. Also make sure you include a makefile that creates an executable named `server_part#` (`#` is either 1 or 2). (Hopefully by doing all this, I can automate the testing and give your grades back sooner than the end of the semester!!).(And please do not turnin any binaries or object code).