

Networked Systems Programming (A.Y. 2023/24)

Lab n. 7 – November 13, 2023

Prof. Eugenio Zimeo

The objective of this lab is to implement, step by step, a simple concurrent Web server with the objective of learning HTTP 1.1.

The server must allow an existing browser to get (by using the GET method) text files (plain or HTML) and the related contents, by exchanging HTTP requests and responses, transported by TCP. These files must be available on the server by placing them in a folder on the machine where the server is running. To this end, the server must be configured by using the following options, passed to the server through a configuration file (e.g. *webserver.conf*):

port NNN

NNN is an ASCII string representing the port number the server will be listening on. If this option is not specified, the default number 80 is used.

root /path

/path is the path that specifies the mapping of the virtual file system (VFS) managed by the Web server and the hosting file system. If this option is not specified, the working directory of the server is the VFS root.

Notes on HTTP:

The format of a HTTP message request is:

```
Request = Request-Line
          *(( general-header
             | request-header
             | entity-header ) CRLF)
          CRLF
          [ message-body ]
```

Where the request line is organized as follows::

```
Request-Line = Method SP Request-URI SP HTTP-Version CRLF
```

The format of the response message is:

```
Response = Status-Line
           *(( general-header
              | response-header
              | entity-header ) CRLF)
           CRLF
           [ message-body ]
```

The status line is organized as follows:

```
Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF
```

Exercise 7.1

By using the default values for *port number* and *root path*, implement a simplified version of the Web server that can print on the standard output the content of the request message received from the client.

Exercise 7.2

Extend exercise 7.1 by sequentially processing the request messages to produce the related response messages containing the requested content.

Add also the code for reading the configuration options from a file (e.g. `webserver.conf`).

Exercise 7.3

Extend exercise 7.2 by introducing concurrent processing of the request messages.

References

HTTP/1.1 is defined by RFC 2616 (www.rfc-editor.org)

Upload the exercises of this lab into your shared folder before the end of today.