Tiny Web Server

Team 4

Version 1

10/13/18

**1. Project Purpose:** The purpose of the Tiny Web Server (TWS) is to produce a miniature web server that can perform the basic functions of receiving requests and respond with basic html pages. This document assumes the audience has a basic understanding of web functions, principles, and related entities, such as HTTP and TCP.

**2. Definitions:**

* Server – The Tiny Web Server running on a computer
* Client – The browser on the user’s computer.
* Request – HTTP GET request. The server responds to other HTTP requests with an HTTP 400 Bad Request.
* Response – The HTTP response sent back from the server to the client in response to its HTTP request.
* Connection – TCP connection made between the client and the server initiated by the client.
* Server – Responding to a request.

**3. System Overview:**

Figure 1 is a basic diagram of the functions of the TWS. Client refers to a user’s web browser on their own device, and the server refers to the TWS program on a separate device. The client interacts with the TWS over the internet by sending an HTTP GET Request to the TWS. The server will then interpret the request, and return an error message if the request is not valid. If the request is valid then the server will retrieve the appropriate HTML files from the hard drive, and send them back to the client browser. The browser can then read the file and display the page appropriately.

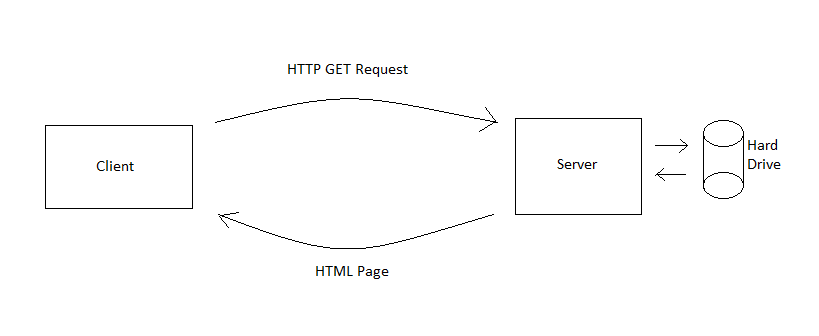
------------------------------------------------------------------------------------------------------------------------------------------

Figure 1 - Model of the Tiny Web Server System

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**4. Requirements:**

**4.1 Functional User Requirements:**

1. The user should be able to access the web server from any browser.
2. The user should be able to successfully follow links within html pages.
3. The user should be able to write the DNS of website that the server host and get the page.
4. The user should be able to receive HTTP respond after requesting the page from web client.
5. The user should be able to save HTML files and images on their device.
6. The user should be able to specify where to save HTML files and images on their device.
7. The user should be able to end the connection to the server before a file has fully loaded.
8. The user should be able to access multiple files before the connection ends to increase efficiency.
9. The user should be able to access the server from multiple different browsers.
10. The user should be able to the received HTML files and images accurately on any browser.

**4.2 Functional System Requirements**

1. The system must display an interface to manage server functions.
2. The system interface should allow the server administrator to view the amount of HTTP GET requests sent to the server.
3. The system interface should allow the server administrator to view the amount of dropped or unreceived HTTP GET requests sent to the server.
4. The system interface should allow the server administrator to view and change the online status of the server.
5. The system interface should allow the system administrator to view the daily, weekly, or monthly traffic of the server.
6. The system should implement TCP connections successfully.
7. The system should be able to handle multiple connections simultaneously.
8. The system must be able to receive HTTP GET requests for static HTML pages and images.
9. The system should display send back an error message if any other type of HTTP GET request is received.
10. The system should accurately send error files 200, 400, or 404 when needed according to the RFC 7231.
11. The system must be able to handle HTTP 1.1 persistent connections.
12. The system should allow the client to specify where any sent HTML pages or images are saved on their device.
13. The system should display all sent HTML Pages and Images in a manner legible to browsers.
14. The system should display all sent and received pages separately
15. The system must have open port (by default port 80)
16. The system must respond with HTTP 404 status if the server didn’t find the page requested.
17. The system should be able to respond to clients for 24/7.
18. The system should allow clients to save files in a designated folder on their device

**4.3. Non-Functional Requirements:**

1. The system computer should be connected to power source.

**6. References:**