

# Design Document: Multi-Threaded HTTP Server with Logging

Julia Sales  
Cruz ID: jesales

---

## 1) Goals

The goal for Assignment 2 is to modify Assignment 1, our HTTP server and add a multi-threaded HTTP server with logging. We take in the commands ex: `./httpserver -N 8 -l my_log.txt localhost 8888`. Where we have “N”, which is the number of worker threads that the server uses. We also have “l”, which logs the data of the requests and writes them into the file specified in the arguments. For this assignment, we must use condition variables, mutexes, and/or semaphores when implementing multi-threading.

---

## 2) Design

For this assignment, there are four parts to this assignment. The first part is that we need to set up the socket and server for our code, which we have already done in Assignment 1. The second part is modifying our code for Assignment 1 where we have to modify the PUT header, where we also take in content length. The third part is that we have to implement multi-threading. Implementing multi-threading will be broken up into more sub parts as we have to take in the argument that counts the number of worker servers and use synchronization mechanisms in our code. The last part is that we have to also implement the logging of the requests. This will also include multiple sub parts as well.

---

### 2.1 Setting Up The Socket

Creating a `sockaddr_in`

1. if `argv[5]` is not NULL then make it `SERVER_NAME_STRING`
2. if `argv[6]` is not NULL then make it `PORT_NUMBER`
3. if `argv[5]` is NULL print the error "Request is missing required `Host` header"
4. if `argv[6]` is NULL print the error "Request is missing required `Port` header"
5. `struct hostent *hent = gethostbyname(SERVER_NAME_STRING /* eg "localhost" */);`
2. `struct sockaddr_in addr;`
3. `memcpy(&addr.sin_addr.s_addr, hent->h_addr, hent->h_length);`
4. `addr.sin_port = htons(PORT_NUMBER);`
5. `addr.sin_family = AF_INET;`

## Creating a Socket

1. `int sock = socket(AF_INET, SOCK_STREAM, 0);`
2. if no connection, when sock is 0
3.     | Error: In socket, no connection

## Socket Setup for Server

1. `int enable = 1;`
2. `setsockopt(sock, SOL_SOCKET, SO_REUSEADDR, &enable, sizeof(enable));`
3. `bind(sock, (struct sockaddr *)&addr, sizeof(addr));`
4.     | if not being able to bind - Error: cannot bind
5. `listen(sock, 0);`
6. `int cl = accept(sock, NULL, NULL);`
7. if cl is < 0
8.     | Error: Cannot accept

---

## **2.2 Modify the PUT Header**