

# Design Document: Simple Web Server

Robert Hu

CruzID: ryhu

## 1 Goals

The goal of this programming assignment is to create and implement a single threaded simple HTTP server. The server will respond to GET and PUT commands to read and write “files”. The server will store files in a directory on the server so it can be restarted or run on a directory with pre-existing files.

## 2 Design

The design is all encapsulated in the original main function. The handling is split between between cases based on the arg\_count

**INPUT** : Argument count: arg\_count

**INPUT** : Array of arguments: arguments

**OUTPUT** : Standard Output

1. Create buffer of size BUFFER\_SIZE
2. Implement Socket creation code given to us in Sections
3. Check for errors in socket creation and connection code
4. Parse message
5. Iterate through each token in returned array
  - a. **For** all tokens i in returned array
    - i. **If** third token is -T flag then
      1. Curl response is Put request
      2. Read fourth token of data
      3. Open --request-target file with create or truncate flags
      4. Write data to target file
    - ii. **End**
    - iii. **Else**
      1. Send requested file
    - iv. **End**
  - b. **End**

**Algorithm 1:** Handling socket creation and send/recv

**INPUT** : Message received

**OUTPUT** : Array of tokens from message

1. Create pch string and token array
2. Set memory of token array to null character
3. Initialize counter
4. Set pch to first token before delimiter
5. **While** pch is not null
  - a. Set token[j++] to string pch
  - b. Set pch to next token
6. Return token array

**Algorithm 2:** Parse message received into tokens