Twisha Sharma
CSE 156
Professor M.Parsa

Lab 1

Code Design

This code is designed to send HTTP requests, more specifically GET and HEAD requests, to a specified server and handle the consequential responses. This code is run through the command line, which allows users to input the hostname, URL path, and optionally, a port number and request type. The code constructs the HTTP request and sends it out based on the user given inputs. After it processes the response and manages the output. The structure of the code

Code Structure

The structure of the code is as follows; the header inclusions are the necessary ones for the networking and standard I/O operations. Additionally, the buffer size is defined so that it may be used for reading data from the server. An error handling function is implemented, that outputs error messages to stderr and exits the program. It takes a string message and a socket file descriptor as arguments. The main HTTP request function is “send_request” which is a function that is responsible for the main operation of sending the HTTP request, and processing a response. The function sets up a TCP socket, connects to the server, sends the HTTP request, reads the response, and prints to standard output. The parameters of the function include host, path, HTTP method, and port. The main function parses the command line arguments to extract the hostname, URL, and optionally the HTTP method. It additionally handles argument validation, and prints usage information if the arguments are incorrect.

Error Handling

The errors within this code are minimal but still make the code not fully functional. The code compiles and runs with no errors or loops, however; running any command with the -h will output the data to stdout.

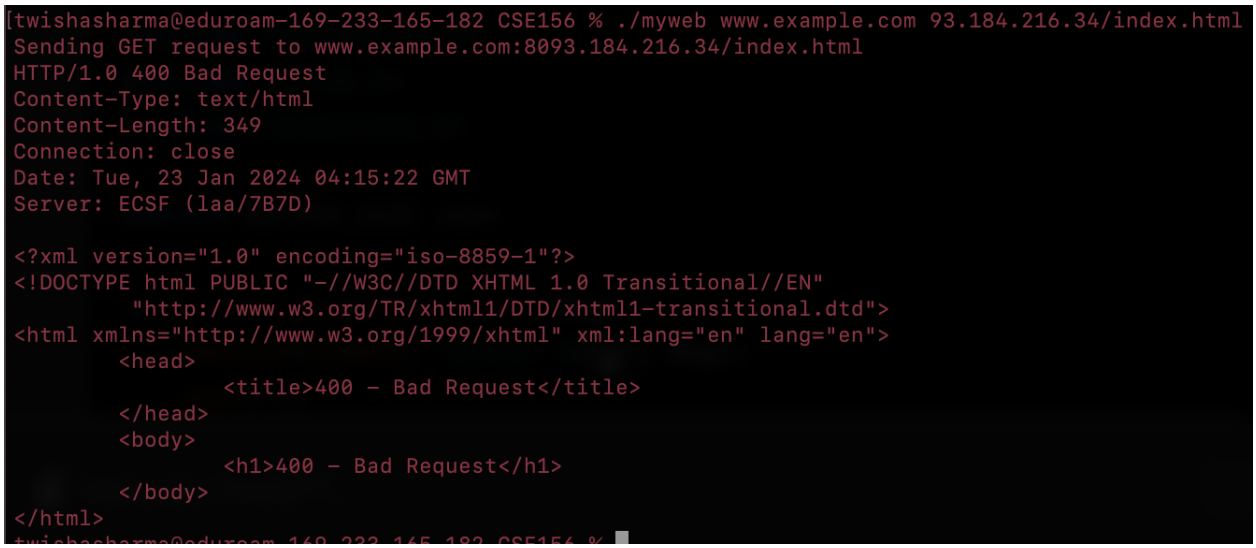
Test Cases

Case 0: Valid Command

Objective: Verify that the program functions as intended with a valid command.

Command: ./myweb www.example.com 93.184.216.34:80/index.html

Expected Outcome: The program should successfully process the command and output the expected results.

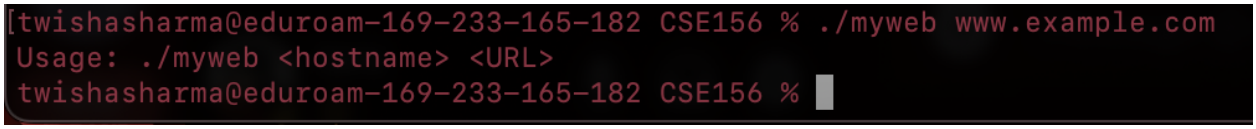


Case 1: Missing Argument - Invalid Port

Objective: Test the program's response to an incomplete command with missing port information.

Command: ./myweb www.example.com

Expected Outcome: The program should not attempt to connect to "www.google.com" as the command lacks sufficient arguments, and it should display an error message indicating incorrect usage.



Case 2: Valid IP Address and Formatted Output

Objective: Ensure that the program correctly processes valid IP addresses and produces properly formatted outputs.

Command: ./myweb alibaba.com 47.246.136.125/

Expected Outcome: The program should correctly process the command line arguments and execute as intended with the valid input.

```
[twishasharma@eduroam-169-233-165-182 CSE156 % ./myweb alibaba.com 47.246.136.125/
Sending GET request to alibaba.com:8047.246.136.125/
HTTP/1.1 302 Moved Temporarily
Server: Tengine
Date: Tue, 23 Jan 2024 04:19:57 GMT
Content-Type: text/html
Content-Length: 215
Connection: close
Location: http://err.taobao.com/error1.html
Via: scproxy033001227030.rg-us-east.us44[web,302]

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
<head><title>302 Found</title></head>
<body>
<center><h1>302 Found</h1></center>
<hr/>Powered by Tengine<hr><center>tengine</center>
</body>
</html>
twishasharma@eduroam-169-233-165-182 CSE156 %
```

Case 3: Valid IP Address with Non-Standard Port

Objective: Test the program's ability to handle valid IP addresses with non-standard ports.

Command: ./myweb info.cern.ch 188.184.21.108:443/index.html

Expected Outcome: The program should successfully connect to the specified host and handle the response.

```
[twishasharma@eduroam-169-233-165-182 CSE156 % ./myweb info.cern.ch 188.184.21.108:443/index.html
Sending GET request to info.cern.ch:443/index.html
HTTP/1.1 400 Bad Request
Date: Tue, 23 Jan 2024 04:20:35 GMT
Server: Apache
Content-Length: 362
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
Reason: You're speaking plain HTTP to an SSL-enabled server port.<br />
 Instead use the HTTPS scheme to access this URL, please.<br />
</p>
</body></html>
```

Case 4: Invalid Hostname

Objective: Examine the program's handling of an invalid hostname.

Command: ./myweb starBucks.com /index.html

Expected Outcome: The program should fail to resolve the hostname and output an appropriate error message.

Case 5: Invalid IP Formatting with -h Option

Objective: Test the program's response to invalid IP formatting combined with the -h option.

Command: ./myweb www.google.com 192.168.1.1.1:80/index.html -h

Expected Outcome: The program should display an error message due to invalid IP formatting, indicating the command is not executable.