Questions

Sebastian Carbonero (sicarbon)

Implementation

1) How did you decide which type of socket to use? Why?

During the process of my implementation, I chose the stream socket becuase of it's reliability and use of TCP. Since this type of socket is reliable, I do not want any duplicates or lost packets since the content I want is loss intolerant. I want to recieve all bytes in order to obtain the full HTML page.

---

2) How did you choose the destination ports?

I chose the destination ports by default or by user input. If the user wanted to use a specific port, I would open up a stream socket with that specific port. I chose port 80 as the default port since webservers mostly use port 80 for HTTP requests.

---

3) What error handling cases did you implement?

These are the following error handling cases I implemented:

- Check if the URL starts with http and not https, if the URL starts with anything else, then It will print an error message.
- Checks if the URL has either a ip or a hostname in the url. If the URL has a IP, then the program would check if a hostname was provided. If no hostname was provided, then the program will print out a error message. If no IP or hostname in the URL was provided, then it prints out a error message
- Checks if IP address provided by the user was valid with the hostname that the user provided.
- On connection, checks if connection was successful, else the program outputs the error.
- Checks if HTTP request was successful.
- On revieve if no data was transferred over 10 seconds, timeout error will happen.
- While reading the HTTP header, checks if there is a `HTTP/1.1 200 OK` was recieved, if not, the program prints out the error code message.
- While reading the HTTP header, checks if chunk encoding is enabled, and prints out a error message stating that chunk encoding is not supported.

---

4) How does your program terminate? What happens to the TCP connection?

- On error, the program closes the socket and quits.
- On Success, the program closes the socket and quits
- If there is a time out, the program closes the socket and quits.

Overall, the TCP connection is closed before termination and leaves no sockets open.

---

5) For the unsuccessful URLs, why were they unsuccessful?

Reason for unsuccessful URLs:

- URL has https in it
- Domain name specified is not valid
- IP that was specified is not valid
- URL does not start with http
- port number is not valid

---

6) What happens if you try to access a site using HTTPS?

The program will print out an error message stating `https is not supported.` But lets assume that my program did not terminate when it identified https in the URL. Then my request would simply get rejected as a result of a HTTPS connection. Since HTTPS connections use port 443 rather than port 80 which requires a SSL certification. As a result, SSL is a type of encryption and decryption protocol used for messages that use HTTPS, and without SSL, our request would be rejected by the server.