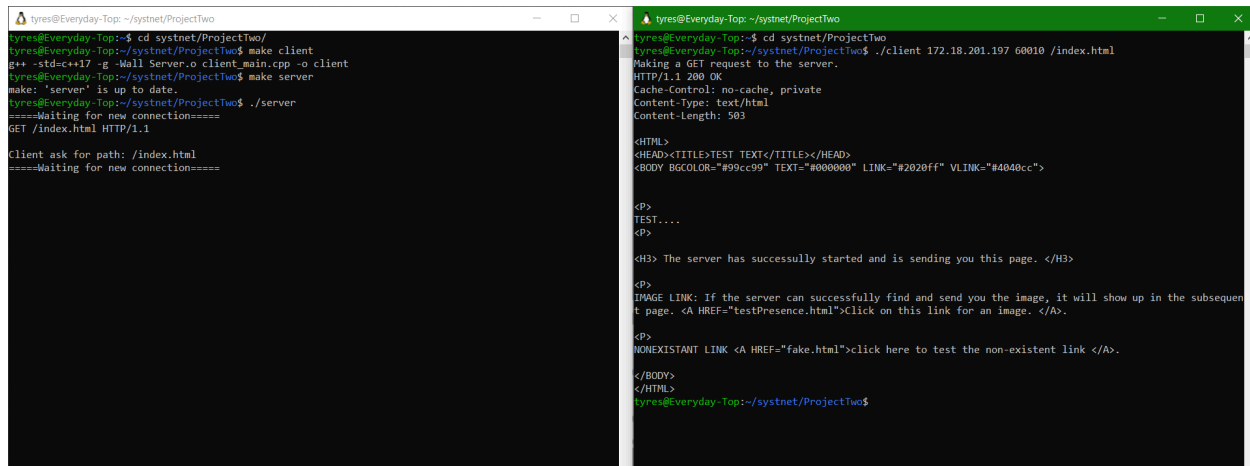


Team Members: Tyre Sheffield, Shep Harper

Screenshots:

1- server (left) and client (right) for index.html request



The left terminal window shows the server's perspective. It starts with the user running 'make client' and 'make server' in the 'systnet/ProjectTwo' directory. The server is then started with './server'. It receives a GET request for '/index.html' and responds with 'HTTP/1.1'. The client then asks for the path '/index.html' and the server responds with '====Waiting for new connection===='.

```
tyres@Everyday-Top: ~/systnet/ProjectTwo
tyres@Everyday-Top:~/systnet/ProjectTwo$ cd systnet/ProjectTwo/
tyres@Everyday-Top:~/systnet/ProjectTwo$ make client
g++ -std=c++17 -g -Wall Server.o client_main.cpp -o client
tyres@Everyday-Top:~/systnet/ProjectTwo$ make server
make: 'server' is up to date.
tyres@Everyday-Top:~/systnet/ProjectTwo$ ./server
====Waiting for new connection====
GET /index.html HTTP/1.1

Client ask for path: /index.html
====Waiting for new connection====
```

The right terminal window shows the client's perspective. It runs './client 172.18.201.197 60010 /index.html'. It makes a GET request to the server and receives a response with status 'HTTP/1.1 200 OK'. The response headers include 'Cache-Control: no-cache, private', 'Content-Type: text/html', and 'Content-Length: 503'. The response body is an HTML document with a title 'TEST TEXT', a body with a background color and text, and a link to 'testPresence.html'.

```
tyres@Everyday-Top:~/systnet/ProjectTwo$ ./client 172.18.201.197 60010 /index.html
Making a GET request to the server.
HTTP/1.1 200 OK
Cache-Control: no-cache, private
Content-Type: text/html
Content-Length: 503

<HTML>
<HEAD><TITLE>TEST TEXT</TITLE></HEAD>
<BODY BGCOLOR="#99cc99" TEXT="#000000" LINK="#2020ff" VLINK="#4040cc">

<P>
TEST....
</P>

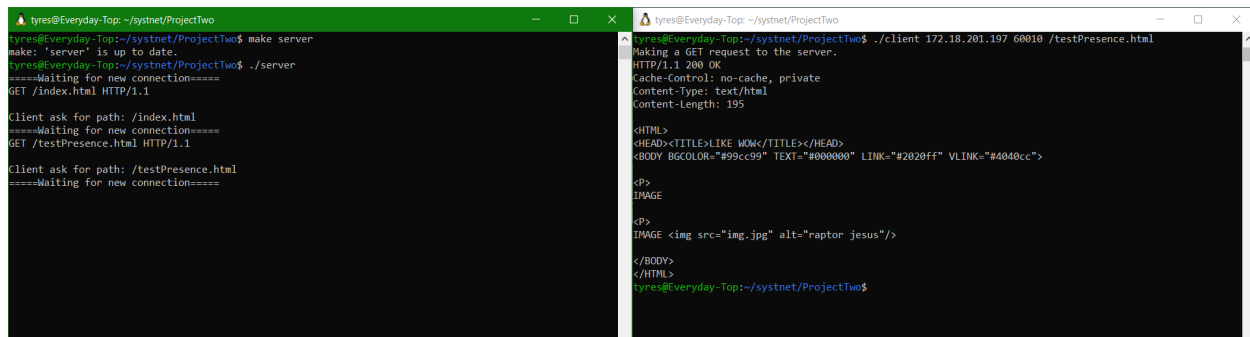
<H3> The server has successfully started and is sending you this page. </H3>

<P>
IMAGE LINK: If the server can successfully find and send you the image, it will show up in the subsequent page. <A HREF="testPresence.html">Click on this link for an image. </A>.

<P>
NONEXISTANT LINK <A HREF="fake.html">click here to test the non-existent link </A>.

</BODY>
</HTML>
tyres@Everyday-Top:~/systnet/ProjectTwo$
```

2- server (left) and client (right) for testPresence.html request



The left terminal window shows the server's perspective. It starts with the user running 'make server' and './server'. It receives a GET request for '/testPresence.html' and responds with 'HTTP/1.1'. The client then asks for the path '/testPresence.html' and the server responds with '====Waiting for new connection===='.

```
tyres@Everyday-Top:~/systnet/ProjectTwo$ make server
make: 'server' is up to date.
tyres@Everyday-Top:~/systnet/ProjectTwo$ ./server
====Waiting for new connection====
GET /index.html HTTP/1.1

Client ask for path: /index.html
====Waiting for new connection====
GET /testPresence.html HTTP/1.1

Client ask for path: /testPresence.html
====Waiting for new connection====
```

The right terminal window shows the client's perspective. It runs './client 172.18.201.197 60010 /testPresence.html'. It makes a GET request to the server and receives a response with status 'HTTP/1.1 200 OK'. The response headers include 'Cache-Control: no-cache, private', 'Content-Type: text/html', and 'Content-Length: 195'. The response body is an HTML document with a title 'LINK MOM', a body with a background color and text, and an image tag with a source 'img.jpg' and alt 'raptor jesus'.

```
tyres@Everyday-Top:~/systnet/ProjectTwo$ ./client 172.18.201.197 60010 /testPresence.html
Making a GET request to the server.
HTTP/1.1 200 OK
Cache-Control: no-cache, private
Content-Type: text/html
Content-Length: 195

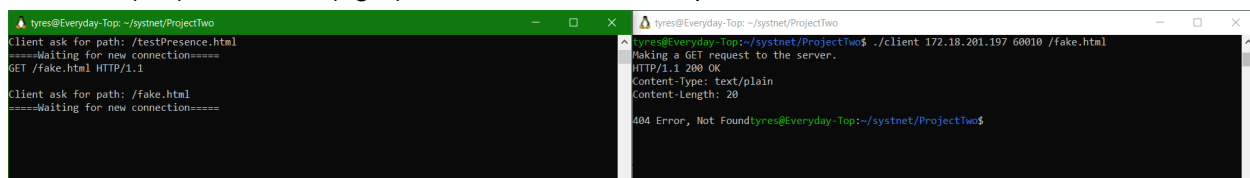
<HTML>
<HEAD><TITLE>LINK MOM</TITLE></HEAD>
<BODY BGCOLOR="#99cc99" TEXT="#000000" LINK="#2020ff" VLINK="#4040cc">

<P>
IMAGE

<P>
IMAGE 

</BODY>
</HTML>
tyres@Everyday-Top:~/systnet/ProjectTwo$
```

3- server (left) and client (right) for nonexistent file request



The left terminal window shows the server's perspective. It starts with the user running './server'. It receives a GET request for '/fake.html' and responds with 'HTTP/1.1'. The client then asks for the path '/fake.html' and the server responds with '====Waiting for new connection===='.

```
tyres@Everyday-Top:~/systnet/ProjectTwo$ ./server
====Waiting for new connection====
GET /fake.html HTTP/1.1

Client ask for path: /fake.html
====Waiting for new connection====
```

The right terminal window shows the client's perspective. It runs './client 172.18.201.197 60010 /fake.html'. It makes a GET request to the server and receives a response with status 'HTTP/1.1 200 OK'. The response headers include 'Content-type: text/plain' and 'Content-Length: 20'. The response body is a 404 error message.

```
tyres@Everyday-Top:~/systnet/ProjectTwo$ ./client 172.18.201.197 60010 /fake.html
Making a GET request to the server.
HTTP/1.1 200 OK
Content-type: text/plain
Content-Length: 20

404 Error, Not Foundtyres@Everyday-Top:~/systnet/ProjectTwo$
```

Protocol:

This server and client implementation uses the TCP protocol with HTTP/1.1. TCP is a reliable and connection oriented protocol that occurs in the Transport Layer. HTTP/1.1 helps with handling the requests in our Server class and the requests being sent from the client_main.cpp to the server.