CS118

(Computer Network Fundamentals)

Project 1: Concurrent Web Server
Using BSD Sockets (Fall 2015)

Derek Nguyen (ID: 304275956, derek)
Sung Hyun Yoon (ID: 904303999, sungy)

High Level Description

First, we open a socket at the specified port number (greater than 1024), and listens for a client connection request. Once we receive a request, we establish a connection with a new socket. When the client sends request, we fork() and handle the request in the child process; we read in the HTTP request and print it out to the console. The parent process, on the other hand, continues to listen for more incoming requests. We then parse the requested file name, and construct an appropriate response accordingly, and return it to the client.

Difficulties and Solutions

One of the main difficulties that we faced in this project was understanding the somewhat vague project specifications. We asked one of the TAs (SeungBae) for clarifications, and with his help, we then made adequate assumptions regarding fine details.

Another difficulty was the lack of helpful development tools in the provided environment. We had difficulty installing tools such as git, curl, IDE, and we resorted environment (collabedit.com).

Also, handling errors regarding requested files was challenging. We were initially unsure of how to react to this situation.

Finding a way to verify that our headers were properly added was somewhat challenging as well. We ended up utilizing a browser developer tool. Deciding which headers should be added required some discussion between us too.

Manual

How to compile: In the console, put webserver.cpp in the working directory, and please run the following command:

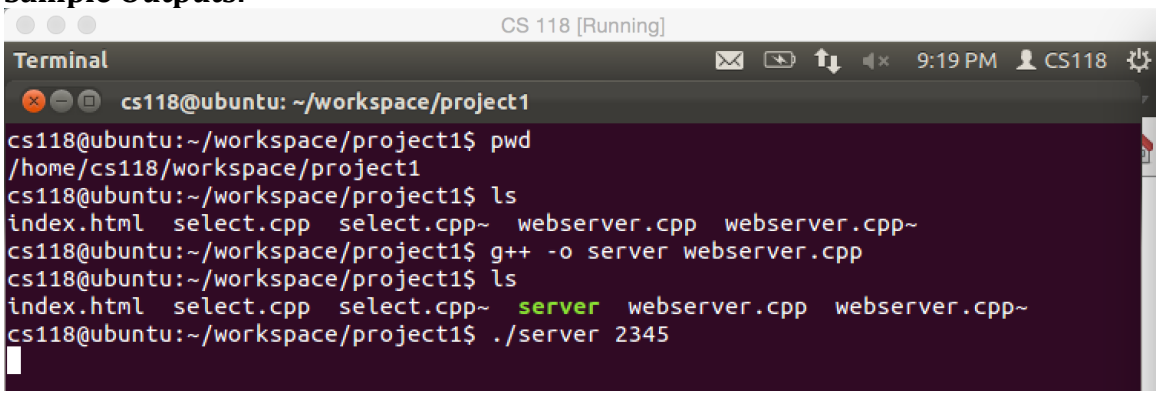
```
g++ -o [executable-name] webserver.cpp
```

How to run the web server:

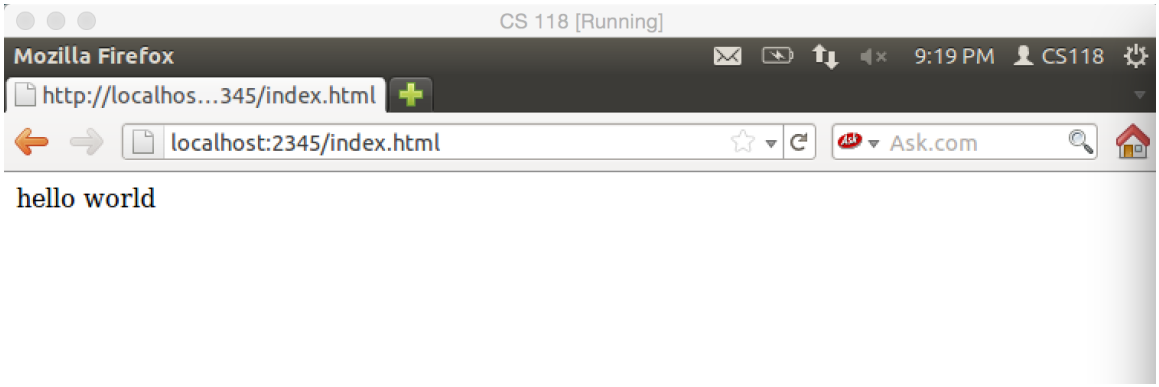
```
./[executable-name] [port-number (> 1024)]
```

Once the server is up and running, interact with the server via your favorite web browser.

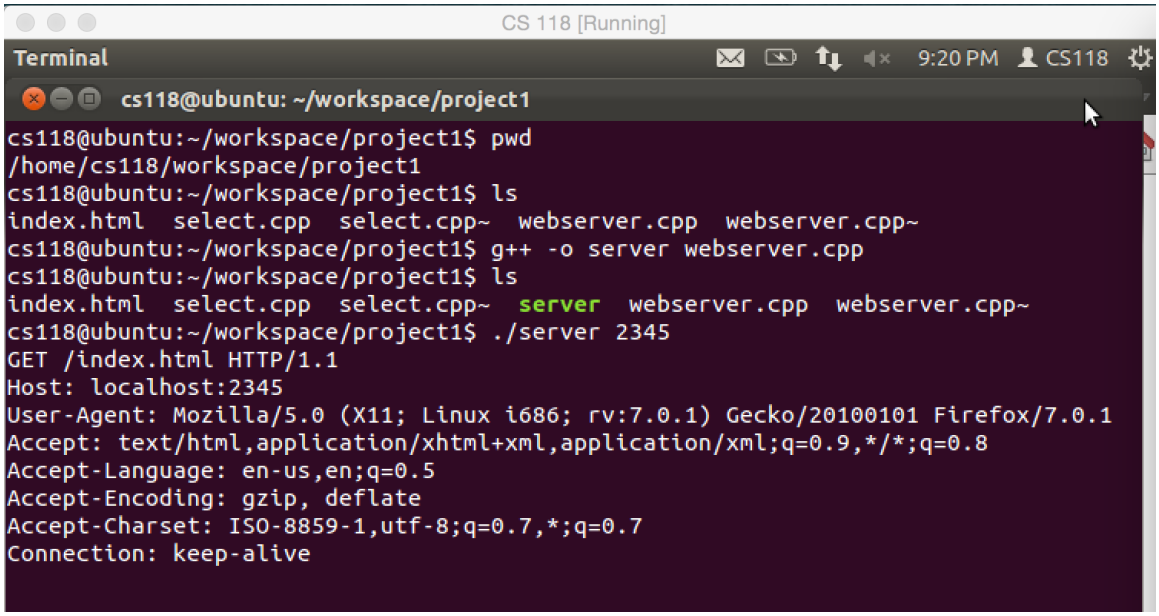
Sample Outputs:



Compiling and executing the web server



Server's response to the client request via Firefox browser



Client browser's HTTP request printed in console