Lab 1 : Concurrent Web Server using BSD Sockets

Brendan Sio (SEASNET id: brendans) 804150223
Roland Zeng (SEASNET id: roland) 204150508

HIGH-LEVEL DESCRIPTION

All of our code was based on the example code given to us on CCLE. All of our code changes are done in function “dostuff.” Here is a high-level description of how we tackled this lab:

We found that **write(sock, msg, length of message)** wrote the given message onto the localhost browser. So, we then tried **write(sock,”<!doctype html><body><h1>hello</h1></body>”,[length-of-that-message])** which successfully wrote “hello” in bold large letters onto the browser. This meant we could serve an html file onto the browser simply by grabbing all the content of that file and using **write** to upload the content to the browser (which would render the html tags automatically).

Hence, we placed some html code inside “**index.html**” and hardcoded that into a string called **filename**. We then used fopen and fread to open the file, while using ftell to read the size of the contents. This let us populate **write** with the message and size of message.

Now, we are almost done. We have to assume the file can be any name (not just index.html) so we must search for the name of the file. Luckily, the name of the file is provided in “buffer”, so we just had to parse buffer after “**GET /**” and before “ **HTTP/1.1**” to find the name of the file. These three steps let us successfully render and display html, gif, and jpg files to the localhost browser.

DIFFICULTIES

We ran into some difficulties while working.

1. **Hard to parse string “buffer” in order to find the file name.**
   We solved this by using the function strstr to look for given substrings; it was just a pain lining up the char pointers.
2. **At first, we could only render html files but were unable to render gif and jpg files**
   This was because we were reading the file content into a buffer called “myBuf” which we defined to have a size of 256. Obviously, this is way too small to accommodate images and gifs, so we fixed this by declaring myBuf AFTER we found the size of the content. as shown below.

```
// Get size of file (http://stackoverflow.com/questions/238603/how-can-i-get-a-files-size-in-c)
fseek(myFile, 0L, SEEK_END);
int sz = ftell(myFile);
fseek(myFile, 0L, SEEK_SET);

// Read the file's contents and put into myBuf
char myBuf[sz+1];
fread(myBuf,1,sz,myFile);
write(sock,myBuf,sz);
```

HOW TO RUN

Open the submitted .zip or clone from our Github repo, https://github.com/rzeng95/CS118Lab1

Add files you wish to open (.html, .gif, .jpg) into the same directory. We have added 'index.html' as an example.

In console type:

make
./serverFork 3000

This creates a connection to localhost. Open up a web browser and type

Localhost:3000/index.html
Localhost:3000/test.gif
Localhost:3000/pic.jpg

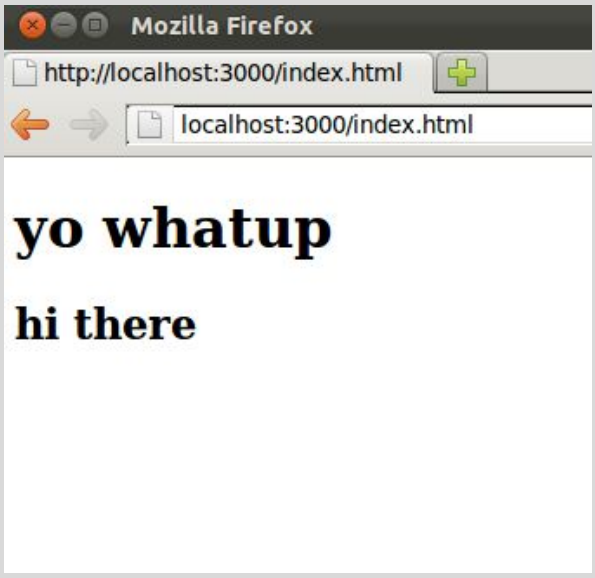
You should see the file rendered onto the browser!

SAMPLE OUTPUTS

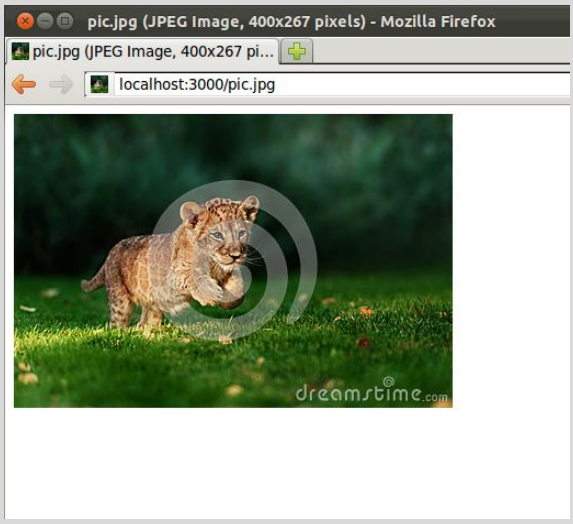
Console output for index.html

```
***BEGINNING OF MESSAGE***

Name of file : index.html
Size of file : 86
Contents of file: <!doctype html>
<html>

<body>
        <h1>yo whatup</h1>
        <h2>hi there</h2>
</body>
</html>
◆◆|t◆◆N0◆

Here is the response:
GET /index.html HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:7.0.1) Gecko/20100101 Firefox/7.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gz

***END OF MESSAGE***
```

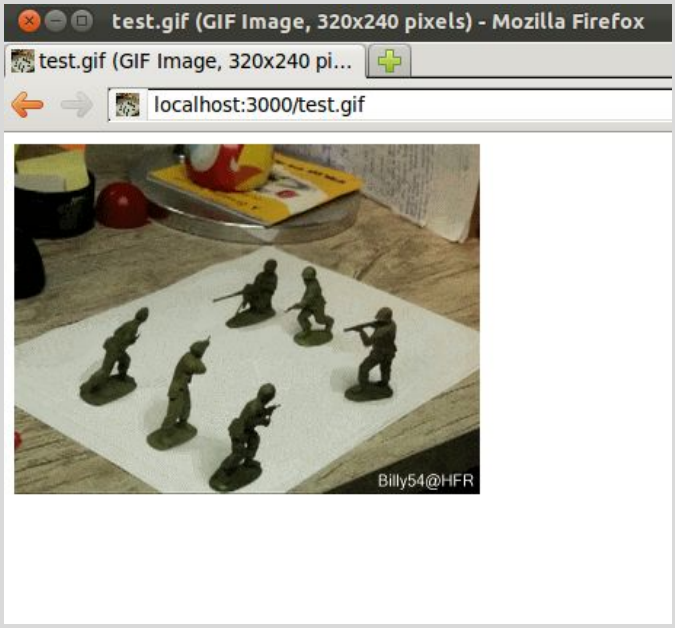
Browser output for index.html



Browser output for pic.jpg



Browser output for test.gif



Browser output for non-existent file blah.html