CS420 Computer Communication and Networks

Assignment 5
Assigned: 10/5/15
Due: 10/22/15 11:59 PM

In this assignment, you will complete the web server that you started in the last assignment. At this point, your program should be able to analyze the HTTP packets sent by the browser. You will need to implement the HTTP GET method alone for completing the assignment. At the end of this assignment, your web server should be able to deliver a simple HTML web page with an embedded image.

Web Server in Java: Part II

Your program should parse the HTTP packet sent by the client to obtain the filename requested by the client. Look at the **StringTokenizer** java class to see how you can use it to parse the request line in the HTTP packet. You can ignore the header lines, since we are not interested in these in the assignment. In case if the file is not found, you should generate a response with status **404 Not found** response. How can we generate this response? The HTTP response has three main parts – thes status line, header lines, and the Entity body. You would need to populate these values, and send these as a stream of bytes to the client. An example code snippet is given for your benefit.

```
if (fileExists)
{
    //code to send the file as a stream of bytes to the client
}
else
{
    //Create status line
    String statusLine = "HTTP/1.1 404 Not found" + CRLF;   // response line
    // header line. Only interested in the type of object we are
sending back
    String contentType = "Content-type: " + contentType( fileName ) +
CRLF;
    //Entity body. HTML file with message Not Found as the text
    String entityBody = "<HTML>" + "<HEAD><TITLE>Not
Found</TITLE></HEAD>" +
    "<BODY>Not Found</BODY></HTML>";
    writeBytes(statusLine);
    writeBytes(contentType);
    writeBytes(CRLF);
    writeBytes(entityBody);
}
```

`writeBytes()` is a method that sends a string as a stream of bytes. You will need to implement this method.

CRLF (Carriage Return and Line Feed - \r and \n respectively) indicate the end of line for both the status line and each header line. An additional CRLF after the header lines indicate the end of header lines.

Note the `contentType( fileName )` method above. This method specifies the type of object sent by the server. Below is the code snippet for this method. You will need to add code to the ae method to deal with image types.

```
private String contentType(String filename)
{
    if (filename.endsWith("html"))
    {
        return("text/html");
    }
    else if ()
    {
        // Left incomplete intentionally; add code for other types.
        // For e.g. "image/gif" and "image/jpeg" can be used for image
        // type files
    }
    else
        return("application/octet-stream");
}
```

If the file exists, then it should be sent as a stream of bytes with the response code **200 OK**.

How do you send a stream of bytes using java? Look at the **DataOutputStream** class, which allows you to send primitive types or objects as a set of bytes.

Specifications

The web server should be started with the following command

java HttpServer <port number>

where port number is user specified at run time.

What to submit

Upload your code in WesternOnline as well as toolman.wiu.edu. Use the *submit_cs420* program to submit your source code (HttpServer.java) only on toolman.

For this assignment, you should use the command

submit_cs420 –s hw5 HttpServer.java

to submit your program. To verify submission, use the command

submit_cs420 –l hw5