CS 4348

Project 3

Prof. Greg Ozbirn

Vineeth Soma

April 25, 2015

# Summary

Filename: WebServer.java

## Platform:

I have used programed my project in Java programming language and NetBeans platform. The code has be tested to run in Unix environment too.

## Implementation:

1. I used Sockets to establish connection with the host server.
2. After the connection is made the program sends an http request and receives response.
3. On the command line, the program displays unparsed http response as-is.
4. On GUI, the web page is display but the EditorKit library in JEditorPane is used to display the webpage.

## Functions:

1. WebServer(String url) constructor: extends JFrame class. Instantiates text editor pane, test field and go button in the frame. Listeners for enter on test field and go button are also instantiated.

The url argument is received from the main function and can be inputted as an argument in command line.

1. Connect( String url): After receiving host and port information from function calls, Socket to host is initiated. Input and output buffers are initiated here.
2. getHost(String url): parses url and returns host information
3. getPort(String url): parses url and returns port information. Returns default port as 80 if url has no port.
4. getPath(String url): parses url and returns path information.
5. getHTTP(): sends through output buffer the HTTP request using the path in the url. Receives response on input buffer.
6. Parse(): reads response on the input buffer line by line. Supposed to handle pursing but does not do much at the moment. Returns parsed text

## Issues:

1. I had issues parsing the http response received from the server.
2. I could not receive the image file and save it on the local folder.

## Things I learned:

1. I learned how to use Sockets to establish connection between client and server.
2. I learned how to use input and output buffers to receive and send data.
3. I learned how to use JFrame components to display GUI and create action listeners.
4. I learned about HTML tags and HTML parsers.

## Result:

1) URL can be accepted as a command line argument or typed into the GUI address box.

2) Web server can be connected to using default port as 80 unless specified in the URL

3) HTTP GET request is send to the server for the requested page.

4) Returned HTML is not parsed in the program but Java libraries are used to do so.

5) Additional HTTP GET request is not sent for images.

6) Page outputted consisting of the text found in the page and the images.