**Operating Systems**

**CS4348**

**Project #3: Network Communication Using Sockets**

**Due Date: Saturday, April 25, 2015**

## I. Project Organization

This project will utilize Sockets for communication between processes.

You should do the following pieces to complete your project. Each piece is explained below:

* Code 50 points
* Output 40 points
* Summary 10 points

# Code

Your code should be nicely formatted with plenty of comments. The code should be easy to read, properly indented, employ good naming standards, good structure, etc.

# Output

Output will be graded by running your program.

# Summary

The summary section should discuss your project experience. You should discuss how you approached the project, including anything difficult or interesting, what was learned, and the end result. This should be at least 1 page in length.

**Teams**

This project may be performed alone or in a team of 2 people. In the case of a team, the work should be performed collaboratively with equal participation or divided evenly. The summary should include discussion of what each team member did. Each team member should be prepared to explain the code if asked to do so.

## II. Project Description

**Language/Platform**

This project must target a Unix platform and execute properly on our CS1 server.

The project must be written in C, C++, or Java.

If using C or C++, you must use the Sockets API with stream sockets.

If using Java, you must use java.net.Socket and java.net.ServerSocket.

Your approach should be similar to the examples given in class.

For C/C++, you must only use read and write system calls to communicate over the socket.

For C/C++, you must account for possible partial messages as discussed in class. This means using a loop to read or write all expected bytes. You should not make a separate call for each byte.

### Web Browser

This project will utilize Sockets for communication between processes.

Your task is to write a simple web browser.

The browser will:

1. Accept a URL as a command line argument or typed into the GUI address box.
2. Connect to the web server (default port is 80 unless specified in the URL).
3. Send an HTTP GET request to the server for the requested page.
4. Parse the returned HTML or display returned error message.
5. For any images in the HTML, send additional HTTP GET requests for each one.
6. Output a page consisting of the text found in the page and the images.

Parsing:

1. Most tags are ignored.
2. The image tag (<img src=…) should be recognized and processed.
3. Though there are software libraries that can parse HTML, do not use these. Just use the normal language features to do parsing.

Rendering:

For a GUI browser:

1. Display the page content vertically.
2. Display images where they should appear within the page content.

For a non-graphical (command-line based) browser:

1. Display the page content vertically.
2. For images, display “Image: filename” where filename is the name of the image file which has been retrieved and stored in the current directory.

## Sample output

Assume www.utdallas.edu/os.html (this is not a real page) contains the following HTML:

<html>

<head>

<title>Operating Systems</title>

</head>

<body>

<p>Welcome to Operating Systems</p>

<img src="os.jpg">

</body>

</html>

For command line browser:

## {cslinux1:~ } java WebBrowser www.utdallas.edu/os.html

Operating Systems

Welcome to Operating Systems

Image: os.jpg

## For GUI browser:

## {cslinux1:~ } java WebBrowser

## III. Project Guidelines

### Submitting

Submit your project on eLearning. Include in your submission the following files:

1. readme.txt. A readme file describing how to compile your project
2. summary.doc A Word document for the summary
3. Your source files

### Academic Honesty

All work must be your own. If cheating is suspected, you will be referred to the Judicial Affairs Office for further discussion. Copying may be detected in a number of ways, including by software which compares your code with all other students’ source code, by comparison with code on the Internet, or by a visual inspection of your source code.

### Resources

The web has many articles on Sockets. There are also books available on Sockets. The course website also contains an example of Socket source code.