

**ECE435: Embedded Systems – Homework 2**  
Sockets Programming

**Due: Monday, 19 September 2016, 3pm**

This Homework is meant to get you started with socket programming. It should run on any Linux machine but it's probably a good idea to try it on your Raspberry Pi.

**1. Download and Build the Code**

(a) Download the code from:

```
http://web.eece.maine.edu/~vweaver/classes/ece435/ece435_hw2_code.tar.gz
```

(b) Unpack the files:

```
tar -xzvf ece435_hw2_code.tar.gz
```

(c) Build the C files:

```
cd ece435_hw2_code  
make
```

**2. Test the Code**

(a) You'll need multiple shell consoles open.

(b) In one, first run `./server`

(c) In another, run `./client`

(d) Type a message on client, and it should travel over the network to server and appear on the server.

(e) Take a look at the code and see how it works.

(f) Note, if you try to re-run the code you might find you get an error such as `Error binding! Address already in use` This is because the client ends so suddenly the network connection is not shut down properly and the network connection enters `TIME_WAIT` state which lasts 60 seconds as the OS waits in case any lingering packets are still on their way. You can possibly avoid this by uncommenting the `sleep()` in the `server.c` code.

**3. Modify the server code (5pts total)**

(a) First modify the server code (`server.c`) so that instead of exiting after one transaction, it instead loops forever reading from the file descriptor and responding (1pts)

(b) Next, modify the client so that it also loops forever, waiting for a message to be typed then sending it. (1pts)

You can always use control-C to quit.

(c) Next modify the server code so that if the string `bye` is sent, it exits the server. (2pts)

You can use the `strcmp()` function for this, but beware the strange behavior of `strcmp()` (0 means a match)

Also note that `fgets()` is going to leave the trailing linefeed at the end of the string so take that into account.

(d) Finally, modify the server so that it receives the string, converts all lowercase characters to uppercase, and sends that as the response. (2pts)

- (e) Be sure to comment your code!

#### 4. **TCPdump** (1pt)

- (a) Install the program “tcpdump” on your Pi. If your pi is on a network and you are running raspbian you should be able to just `apt-get install tcpdump`
- (b) tcpdump can be used to grab raw network packets off the network and show their contents. This is very useful for debugging network problems.
- (c) Run tcpdump while your client/server is running. `sudo tcpdump -A -i lo`  
Note that last part is the lowercase letters LO (as in loopback)
- (d) Cut-and-paste the result of one round trip client/server communication into the README file.

#### 5. **Answer the following questions** (3pts total)

Short answers are fine. Put your answers in the README file using a text editor, it will be automatically included in the submission process.

- (a) What is one benefit satellite connections have over fiber?
- (b) What is one benefit fiber connections have over satellite?
- (c) You decide you want to build a transmitter that transmits at 4.3GHz. Can you? Who is likely to get upset about this?

#### 6. **Submit your work**

- Run `make submit` which will create a `hw2_submit.tar.gz` file containing README, Makefile, `server.c` and `client.c`.  
You can verify the contents with `tar -tzvf hw2_submit.tar.gz`
- e-mail the `hw2_submit.tar.gz` file to me by the homework deadline. Be sure to send the proper file!