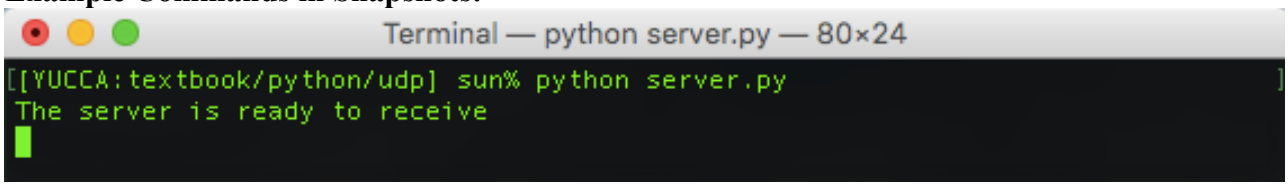
Worcester Polytechnic Institute
Department of Computer Science
CS4516– Computer Networks

Sample Socket Programming– Lecture Example

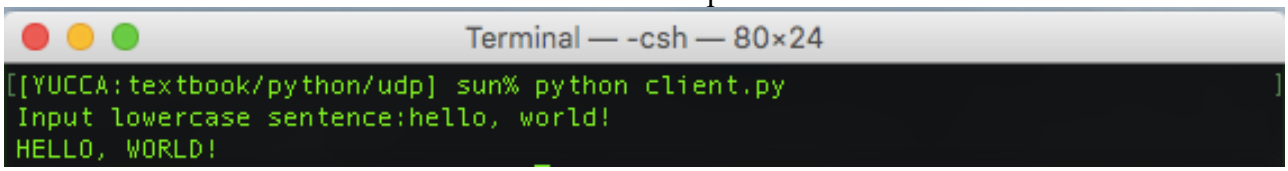
Goal: Get familiar with socket programming assignments by repeating the demo in the lecture. The program has to run without errors in linux.wpi.edu server.

Instructions: Please repeat what's done in the lecture about implementing the **UDP** and **TCP** client/server interactions with Python. If you prefer C or Java implementation, that's OK, but Python is encouraged in this assignment.

Example Commands in Snapshots:



Server side snapshot



Client side snapshot

Notes:

1. For Windows users, you may use a tool Thonny to do python programming. https://thonny.org/
2. For Mac users, you can directly do programming by starting the Terminal, as shown in the demo.
3. If you cannot install Python on your own machine, you can connect to ***linux.wpi.edu*** **server** on campus network to do the assignment. If you are off campus and ***you try to run the programs on the server***, you need to use VPN to connect back to campus network and use ssh to connect to the server. You don't need VPN or ssh if you run all your programs on your local machine.
   Please see more instruction here:
   https://hub.wpi.edu/article/786/connect-to-linux-cluster-using-a-terminal-program
4. When you connect to linux.wpi.edu, you may get hostnames other than `linux.wpi.edu` (as given the examples above). Please use the command `hostname` to find out where you are connected. Be cautious that `linux.wpi.edu` is made up of currently four systems. Please make sure that you are connecting to the correct one that you ran your code on. This will be displayed on the default prompt (e.g. `jkingsley@ccc-app-p-u15`), or by running the `hostname` command.

5. You must choose a server port number larger that 1023 (to be safe, choose a server port number larger than 5000). If you are using linux.wpi.edu for testing, WPI has made ports in the range 18000-19000 available for use.
6. If you need to kill a background process after you have started it, you can use the UNIX *kill* command. Use the UNIX *ps* command to find the process id of your server.
7. Make sure you close unused sockets that you use in your program.
8. If you abort your program, the socket may still hang around and the next time you try and bind a new socket to the port ID you previously used (but never closed), you may get an error.
9. On UNIX systems, you can run the command "netstat" to see which port numbers are currently assigned.