CSCI 315 Operating Systems Design
Fall 2014 - Prof. Felipe Perrone
Activity 24

Work on this activity with at least one partner in class.

1) Study the C code in files `send_udp.c` and `recv_udp.c` then tackle the following questions. You may need to poke around the web or the APUE textbook to find the answers you need.
    a) What is the function of a *port number*?
    b) Why does the port number of the sender for received messages keep changing every time you send a new message?
    c) Given that the port number to receive data is hardcoded in `recv_udp.c`, when multiple students run this program in the same computer, multiple programs will be listening on the same port. Coordinate with a few other students to create an experiment to determine whether this will cause problems for the reliable reception of messages.

2) Change both programs `send_udp.c` and `recv_udp.c` so that you don't have to hardwire the port number into the programs.

3) Having the structure of the message carried in the datagram defined in both programs `send_udp.c` and `recv_udp.c` is not ideal. Modify the programs so that you can define the structure of the message in one single file. The data structure of the message should contain: an `int` number, a `long int` number, a `double` number, and a C string with 80 characters.

4) Modify the two programs so they can both use:
    a) A function to initialize a message data structure with data typed in from `stdin`.
    b) A function to print to `stdout` a message data structure.

5) Change `send_udp.c` to run on an infinite loop, in which it requests the user to type up a message and then sends the message to the destination. Change `recv_udp.c` to print to `stdout` each message received. Make sure to use the `connect` system call to set up a default destination for each datagram.