Purpose: This assignment is to gain an in-depth understanding of a UDP client-server program.

You will build a connectionless time client and an iterative connectionless time server using the techniques found in Chapter 9 of the text. (You will use UDP.)

Submission: You must submit a print out of both the client and the server. In addition you must place the source code for both the client and the server in your home directory in files called `timec.c` and `timed.c`, respectively. I will access these files to check the correctness of your programs.

Port numbers: You must use your personal well-known-port number. Copy the file `~volper/classes/472/shells/get_port.c` into your home directory. The `get_port` procedure in this file returns your personal port number. Use an include directive to make the procedure available, and call this procedure in both your client and server at the point the *default* service is set up (see the lecture notes).

THE TIME SERVER

Your time server will listen on your well known UDP port. Whenever it receives a message from a client it will call `gettimeofday` and return the `tv_sec` and `tv_usec` values to the client. The call is similar to the `time` in Comer's example, except it returns a structure. It does have a second parameter which is unused, pass a `NULL` for that parameter. Don't forget to use the ampersand on the first parameter (see Comer's example).

You are returning 8 bytes; you will need an 8 byte array.

Sequence:

1) Get a request.

2) Use the `gettimeofday` call.

3) Print (to the screen) the values of `tv_sec` and `tv_usec` in hexidecimal.

4) `htonl` the `tv_sec` and `tv_usec`, go ahead and put them right back into the structure.

5) Use two calls to `memcpy`, to copy the sec into the array starting at position 0 and the usec into the array starting at position 4.

6) Send the array (8 bytes).

THE TIME CLIENT

Your time client will contact your server and get the time on the server's machine. The client will send a message and receive as a reply the time on the server's machine given as the two `gettimeofday` numbers (`tv_sec` and `tv_usec`).

You are receiving 8 bytes; you will need an 8 byte array.

You are receiving a time of day you are required to use a `timeval` structure to contain the final answer.

Sequence:

1) Send a request to the server.

2) Read the reply.

3) Use two calls to `memcpy` to copy the sec and usec into the (`tv_sec` and `tv_usec`) fields of a `timeval` structure.

4) `ntohl` the `tv_sec` and `tv_usec`, go ahead and put them right back into the structure.

5) Print (to the screen) the values of `tv_sec` and `tv_usec` in hexidecimal.

Recommendations: (not requirements)

Server) A good starting point is Comer's `UDPtimed.c`. Don't forget to change your default `service` to use `get_port`. Get rid of the Unix Epoch stuff, have both your client and your server use the `gettimeofday` time format.

Client) Start with your UDP client from your homework, it's most of the solution.