

CSCE 3600: Systems Programming Recitation Assignment 9

– Implementing Sockets

Available: Week 11 Due: Week 12

PROGRAM DESCRIPTION:

In this assignment, you are provided with almost-working code that establishes a TCP socket connection over the INET domain (though for this assignment you should run and test your client-server code on the same machine). What is missing are the actual parameters for each of the four connection establishment socket APIs on the server and the two connection establishment socket APIs on the client (you will see a ? where the function parameters should be). Here are some identifying characteristics about the sockets you will be creating:

- Use the INET domain for the socket type.
- Use TCP sockets.
- Use a backlog of 10 partially completed connections for the kernel to queue.

Your goal for the first part of this assignment is to fill in the socket APIs with their needed parameters, but you should not add any new lines of code (until told to do so) as all of the needed supporting information such as variable declarations and initializations has already been provided. Use what is given to you, but do not change the intent of the program.

If completed successfully, you will see the message “Server Message: SUCCESS” on the client side, but nothing on the server-side (although the server-side should still be running). The client-side only runs once while the server-side runs indefinitely, so you can run the client multiple times.

To quit the server-side program, you may use CTRL-C (i.e., ^C). Go ahead and stop the server-side socket program now and then attempt to run the server-side socket program again. Does it work? Or does it give you an error? Knowing why you are getting this error is important! Normally, we would call `unlink()` to delete the socket “file”, but this only works on UNIX domain sockets, not the INET sockets that we are using. For INET sockets, there is no file system token representing them, so we need to set a socket option to re-use the address as follows:

```
int on = 1;

setsockopt(listenfd, SOL_SOCKET, SO_REUSEADDR, &on, sizeof(on));
```

Now, enter the above two lines of code just before where the error is occurring. Then recompile and run your socket program again. Hopefully, you will no longer get these same errors as before. Make sure your client-server socket program works as expected and then submit both code files.

REQUIREMENTS:

- No comments are required for this recitation assignment, except for your name at the top of the program.
- Your program should have two components named `“rec09svr.c”` and `“rec09cli.c”`, without the quotes, for the server and client code, respectively.
- Your program will be graded based largely on whether it works correctly on the CSE machines (e.g., `cse01`, `cse02`, ..., `cse06`), so you should make sure that your program compiles and runs on a CSE machine.
- Although this assignment is to be submitted individually (i.e., each student will submit his/her own source code), you may receive assistance from your TA, IA (i.e., Grader), and even other classmates. Please remember that you are ultimately responsible for learning and comprehending this material as the recitation assignments are given in preparation for the minor assignments, which must be completed individually.

SUBMISSION:

- You will electronically submit your programs, `rec09svr.c` and `rec09cli.c`, to the **Recitation 9** dropbox in Canvas by the due date and time. No late recitation assignments will be accepted.