

CSE3300/CSE5299: Computer Networking

Programming Assignment 1: Socket Programming

Due Date: March 8, 2021. Submission through HuskyCT.

1 Purpose of Assignment

In this assignment, you will practice TCP socket programming. In the first part of the assignment, you will write a client and a single-threaded server. In the second part of the assignment, you will extend the server to a multi-threaded server. Feel free to use any programming language that you are comfortable with. You are highly encouraged to use python since the sample code we covered in class is written in python. In addition, the sample multi-threaded server program that you are given in this assignment is in Python (see Section 3).

2 Basic Assignment

In the basic assignment, you will implement a client that queries for English words that are stored in a word list at a server. The program needs to be done using TCP socket programming since we would like reliability that is guaranteed by TCP.

The client sends a wildcard query that you enter through the keyboard to the server. The only type of wildcard you need to support is the ones with one or multiple questions marks. Specifically, a query can be ‘a?t’, where ‘?’ means an arbitrary letter. When the client sends such a query to the server, it means to get all the words that contain three letters, starting with ‘a’, ending with ‘t’ and having an arbitrary letter in the middle (e.g., one matching word is ‘ant’). The server has a text file (`wordlist.txt`, enclosed in this assignment in HuskyCT), which has a list of English words. The

server's job is waiting for queries from clients to come in. Once receiving a query, the server looks up the wordlist and returns all the matching words back to the client. The client will then display the words and then terminate. The server only needs to serve one client at one time. The client only needs to send one query.

3 Multi-threaded Server

Now you are going to make your server to be multi-threaded so that it can serve multiple clients simultaneously. You will also extend your client to allow it to send multiple queries; each query is input by you through the keyboard. When you enter "quit", the client will terminate.

To help you write the multi-threaded server program, we provide a sample source code (called `thread-server.py`) to you. The code is taken from the book "Programming Python" (by Mark Lutz). It has detailed documentation. In particular, read `dispatcher` function, which delegates a client to a newly created thread that runs `handleClient` function. As a result, the server can serve multiple clients simultaneously.

Test your code and make sure the server can serve multiple players simultaneously. To do this, you can (i) run the server in one terminal, (ii) open another terminal to run one client (do not terminate the client), and (iii) open yet another terminal to run the second client.

4 What to Turn In

When you hand in your programming assignment, please include the following (please submit to HuskyCT):

- A program listing containing in-line documentation. Uncommented code will be heavily penalized.
- A separate (typed) document describing the overall program design, a verbal description of how it works", and design tradeoffs considered and made. Also describe possible improvements and extensions to your program (and sketch how they might be made). The format of the description file should be as follows (If your turn-in does not follow the above format, you'll get 10 points off automatically):
 - Description: describe the overall program design and "how it works"

- Tradeoffs: discuss the design tradeoffs you considered and made.
- Extensions: describe possible improvements and extensions to your program, and describe briefly how to achieve them.
- Test cases: describe the test cases that you ran to convince yourself (and us) that it is indeed correct. Also describe any cases for which your program is known not to work correctly. Please use screenshots to show the results.

5 Grading policy (Total: 100 points)

- Program Listing
works correctly: 50 points (shown by testing results)
in-line documentation: 10 points
quality of design: 10 points
- Design Document
description: 5 points
tradeoffs discussion: 5 points
extensions discussion: 5 points
- Thoroughness of test cases: 15 points

Notes: A full 10 points for quality of design will only be given to well-designed, thorough programs. A correctly working, documented and tested program will not necessarily receive these 10 points.

6 A Few Words About Borrowing Code...

As many of you probably already know, there is a wealth of sample socket programming code out on the Web and in books that you can use in your projects. Often learning from sample codes is the best way to learn. I have no problem with your borrowing code as long as you follow some guidelines:

- You may not borrow code written by a fellow student for the same project.
- You must acknowledge the source of any borrowed code. This means providing a reference to a book or URL where the code appears (you don't need to provide reference for the code that was given to you).

- In your design document, you must identify which portions of your code were borrowed and which were written by yourself. This means explaining any modifications and extension you made to the code you borrowed. You should also use comments in your source code to distinguish borrowed code from code you wrote yourself.
- You should only use code that is freely available in the public domain.