IOMAXIS Technical Evaluation Problem

Introduction

This programming exercise is a distilled version of a real-world problem our team previously worked on, and is designed to help us understand your fundamental programming skills. It covers data structures, control structures (i.e., branches and loops), and networking.

Your time is valuable, and we do not expect you to spend more than 2 hours on the entire problem. If it is taking you significantly longer to solve the initial problem, consider reassessing your overall approach.

While crafting your solution, please keep in mind we value correctness, robustness, simplicity, and code cleanliness. Your code should be written as if it is going to be committed to your new team's repository!

Upon submittal of your solution, the interview team will review your work within 24-48 hours, and determine the next steps of the vetting process.

Time Limits

- Do not to exceed 2 hours on solving the problem.
- A solution must be returned within 24 hours of receiving the problem.

Prerequisites

- Evaluation interview problem (provided)

Instructions

1. Answer the following problem. Your solution should include a small set of unit tests verifying its correctness.
2. Submit your solution (as a zip) with run instructions using the "Attachments" button on the SmartRecruiters job page. Please include all files required to run your program, (e.g. language configuration files such as Cargo.toml for Rust, or .csproj for C#). We will use your instructions to review and run the solution.

Problem

Purpose

Exercise the candidate's skill networking, looping, data structures and solution design. The candidate should be able to discuss the pros and cons of their approach.

Problem Statement

A customer requested a simple file transfer mechanism between a client and server.

Client:
The client on the command line will take in arguments to specify the server (IP and Port) to connect to and a file to send. The client will connect to the server and send any necessary data in order to complete the transfer in a successful manner. This will include the file content and a file path to where to write the file on the server, i.e. filename. Other information maybe sent as desired. After sending the file, the client must receive a response from the server indicating if the the file transfer is complete.

Server:
The server on the command line will take in arguments necessary to start a listening connection. The server will listen for incoming connections and messages. When a message is received the server write the file and write to disk. If the server can write the file to disk, it will send a confirmation back to the client that the file was received successfully. If the server can not write the file or an error occurs, the servers responds to the client with a negative status indicating the transfer was incomplete. After handling a message the server will go back to a listening state.

Problem Statement Parameters

- It is your choice on how to model the client and server.
    - You may chose what protocols the client and server use.
- A “successful” file transfer means the file was written to disk by the server. It is your choice how to communicate success or failure.
- For simplicity, only one connection will be handled by the server at a time and only plain text files need to be transferred.
- The code should be composed in a way that engineers can reuse the functionality of the client/server.

Expected Behavior

- Server starts listening on a address:port
- Client connects to server on address:port
- Client reads file to send.

- Client sends file content and destination file path to server.
- Server processes information writes file to disk.
- Server sends a message to the Client with a success/unsuccessful status.
- Client receives success/unsuccessful message and prints result.
- Client closes connection
- Server returns to listening state.