Programming Assignment 4

TCP/IP Client-Server Interaction[1]

Assignment spec may change depending upon discussion in the class, this is a working doc

Release Time	Due Date
November 05, 2024	November 14, 2024

Objectives

You will design and implement a simple TCP/IP server and client program. The server should read data sent by the client, convert the bytes to integers, and print them. Your task is to create both the server and client programs.

Prerequisites:

- Basic understanding of Linux/Unix operating system.
- Familiarity with client/server programming in C/C++
- Familiarity with socket programing in C/C++

Tasks:

- **TCP Server Implementation:**
  - Write a C program to create a TCP server.
  - Initialize a TCP socket.
  - Bind the socket to a specific port.
  - Continuously listen for incoming TCP packets.
  - Read the data from the packets, convert to integers and print the integers to the console.
- **TCP Client Implementation:**
  - Write a C program to create a TCP client.
  - Initialize a TCP socket.
  - Read bytes from a file and send them to the server.
  - Implement a delay of 10 seconds between sending each packet.
- **Buffer Size Consistency:**
  - Ensure that the server and client use the same buffer size. Document the chosen buffer size.
- **Client-Server Interaction:**
  - The client should send multiple packets to the server.
  - The server should print the received data converted to integers for each packet.
  - The client and the server should also print the current system time before sending or after receiving a packet.
- **Error Handling:**
  - Implement error handling in both the server and client programs.
  - Test and document error conditions (e.g., server not available, client unable to send).

[1] Idea of this assignment courtesy Dr. Shameek Bhattacharjee

- **Testing:**
  - Create test cases to validate the functionality of your programs.
  - Test with different buffer sizes, file sizes, and network conditions.
  - Test with multiple cli8ents concurrently running (say at least 3 clients and one server)
- **Documentation:**
  - Document your code thoroughly, including comments explaining key sections.
  - Write a brief report summarizing your design choices, challenges faced, and testing results.

Additional Instructions:

- Use the provided sample input files (**hw4input.txt**) for initial testing and debugging.
- Each line in the file represents a set of hexadecimal bytes that the client program should read and send to the server.
- Implement your client program to read these files, and send them to the server.
- Implement your server program to receive these packets, convert the bytes to integers, and print the integers to the console.
- Test your programs with different input files to ensure they handle varied scenarios.

Submission Guidelines:

- Run your TCP client server application using various test cases and log them in a script file.
- Submit the source code for both the server and client programs along with your script file.
- Include the report detailing your design choices, challenges faced, and testing results. Include the listing of source and script file content in an appendix.
- Ensure that the code and report are well-documented and organized.

Notes:

- Test your scripts in a safe and controlled environment. Do not run scripts on critical systems without proper testing and permissions.
- You may use any programming environment that supports C.
- Test your programs on different machines or virtual environments to ensure compatibility.
- Pay attention to error handling and robustness in your code.
- Creativity in identifying and handling various error conditions will be appreciated.
- Creativity in designing test cases and robustness of your code will be rewarded.
- The assignment is designed to assess your understanding of UDP, socket programming, error handling, and your ability to design and implement solutions independently.

Congratulations! You're well on your way to writing programs with inter-process communications! 🔍

References

- Gen-AI tools
- …

If you have creative ideas for extensions or making it more interesting, run them by the course staff, and we'd be happy to give you guidance!

Design Requirements

Code Documentation

For this assignment, you must properly document your code and use good software development practices.

Github

Use github to store your repository. Use good revision-control-system practices as you develop various pieces of the search engine.

Testing

Make sure you test your application with several different values capturing different cases, to make sure it works.

Assignment Submission

- Generate a .zip file that bundles all your files, including:
  - Readme file
  - Source code files
  - any input or output files
  - sample test runs in a script file
  - signed plagiarism statement
- Don't forget to follow the naming convention specified for submitting assignments
- You will also show execution of your application to grader / instructor. They may give you a test case or two on the spot.