Mark Demore, 2d Lt

CSCE689 – HW1

CODE:

<https://github.com/mdemore2/AFIT-CSCE689-HW1-S/tree/0e4bd5d3b6f0733c19654171a2ac68e57df2b0d2>

QUESTIONS:

1. Based on what you've read so far, would you define this application as a distributed system? Why or why not? Use the design goals listed in Ch 1 to justify your position.

 According to the text, a distributed system is “a collection of autonomous computing elements that appear to its users as a single coherent system”. It goes on to elaborate the two key characteristics of this definition, “a collection of autonomous computing elements” and “a single coherent system”. In this regard, the client/server application could be considered a distributed system. The client and server occupy separate computing elements and are ‘autonomous’ outside of user input, and makeup a single coherent system in that they interact with each other and exchange information. The text also enumerates design goals for a distributed system: making resources easily accessible, hiding the fact that resources are distributed, open, and scalable. The client/server application also meets these criteria at a minimum. The server provides processing resources for user input from the client and the client is unaware that input is being processed by another application. The only limit of the system is the maximum number of clients allowed (which can be changed in the source code, limited by available resources to the server) making it open and scalable.

2. Based on your Chapter 2 reading, what architectural style(s) does your software leverage? Explain why it fits.

 My implementation of the client/server application leverages a pure-layered architecture. This architecture fits because all processing is done on the server side and the client simply takes user input. The client makes a downcall to the server and waits for a response.

3. Based on your Chapter 2 reading, what system architecture does your software leverage?

 Given that the assignment was a simple client/server application, my implementation leverages a simple client-server architecture, a physically two-tiered architecture. My specific architecture only has the user interface on the client side, with all other functionality on the server side.

4. What steps would you need to take to "evolve" your code into the following types of systems. List at least three major tasks each:

A. A three-tiered architecture pulling data from an SQL database.

- Means of distinguishing between SQL return and new client request on socket

- Translation from client request to SQL request

- Translation from SQL reply to meaningful data to display to user

B. A node of a peer-to-peer system (structured or unstructured).

- Combine server functionality with client (handling multiple connections, processing data)

- Search implementation (passing request to new nodes if data is unavailable)

- Implement horizontal distribution (share processing/data)

C. An edge-server system

- Add replication of data functionality

- Enable distribution of processing among servers

- Distinguish between clients/edge servers