Lab 1: Bare metal communication

Objective:

An introduction to communication at a basic (socket) coding. Regardless of what software stack is being used, communication breaks down to this set of code. Your first objective is to explore our basic socket project to understand how communication is established between two processes (client-server, server-server), and how messages are encoded and decoded. Obviously, this represents a simplistic view of how processes communicate however, in understanding distributed computing, we must start with what is happening.

A second objective is to learn how to build and run projects OUTSIDE OF AN IDE. Yes. Editing and code exploration is easier in an IDE. However, we don't run servers in IDEs. Therefore, your second challenge is to understand scripting (bash) and how to build/run from the command line.

Extend your investigation to:

1. Measure latency of message composition and communication latency of your laptop/computer to determine peak performance.

2. Improvements? Review the supplied code for robust and health. Are there improvements [fix/enhance/refactor]? Why? Be prepared to discuss in this coming class.

Overview:

This lab servers two purposes 0) benchmarking your computer, 1) to look at and consider data movement at the lowest code level, and 2) Server-side coding practices.

Data – When we build multi-server (or multi-tier) systems, we must be aware of the latency or cost of separating functionality across multiple process [servers]. Exploring this small code example you should look at what is involved in moving data between two points (A,B).

Questions you should consider are:

1. Memory overhead – entropy of memory allocation/deallocation
2. Representation of data elements – strengths/weaknesses
3. Robustness and scaling – what are the faults and assumptions of the code

When refactoring the code, how do you measure the effects of your changes?

Practices – Often we think of coding cast as building applications or interactions. However, server-side coding presents new/different focuses. Examine the code for coding (good/bad) practices. Identify and comment on your observations.

I have added [enhanced] the project with code fragments observed in prior project submissions, from your peers for what to watch for in your code.

Running the project:

This project doesn't use an IDE to build and run the project. It has been setup to use an older build tool, apache ant, and is ran outside of the IDE – scripts.

Running – The server and each client should be started in its own separate window.