Remote Procedure Call (RPC) and Middleware

Chapter 38, Tools that aid programmers in creating client-server systems.

Programming Clients and Servers

- Writing client-server software is difficult.
  - Complex issues of communication.
  - The sockets API causes the programmer to need to specify many low-level details.
  - Errors can go undetected until production.
- The structure of many client-server programs is very similar.
- A tool is a software program that itself generates all or part of another computer program.
  - Tools eliminate much of the “busy work” of writing client-server code.

Remote Procedure Call (RPC)

- Most programmers are familiar with procedures and calling them.
- Why not allow programmers to call procedures on other computers?

RPC Paradigm

- Once the programmer has built a “conventional” program to solve the problem, he considers how to divide it into two pieces.

Communication Stubs

- Two stubs handle the communication details.
- Neither the main program nor the called procedure need to be modified.

External Data Representation

- The client and the server must agree on how data values are represented.
- If they do not, they must agree on a common representation for communication purposes (an external data representation).

One machine.  Client-Server.
Object-Oriented Middleware

- Interface Definition Language (IDL) is a way of telling a tool the characteristics of procedure calls from client to server.
- These tools are called “middleware.” Most current middleware is object oriented.
  - Open Network Computing RPC - Sun
  - Distributed Computing Environment RPC
  - Microsoft RPC (derived from DCE RPC)
  - Common Object Request Broker Architecture (COBRA) - stubs (proxies) are made at run time.
  - Microsoft Object RPC - object version of MRPC.
  - Common Object Model and Distributed COM