

EECS 22L: Software Engineering Project in C Language

Lecture 8

Rainer Dömer

doemer@uci.edu

The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine

Lecture 8: Overview

- Project 2 Technical Discussion and Advise
 - Application specification
 - Client-server software architecture
- Introduction to Socket Communication
 - Basic terms and concepts
 - Client-server example

Project 2: Software Specification

- Introduction
 - *Chat Tool*
 - Instant Text Messaging Application
 - A. User App
 - Register user's account name, password, IP address, etc.
 - Login, update user's status: available, idle, or offline
 - List available users, add/delete contacts
 - Chat with text messages, images, ...
 - B. Provider Service
 - Management of user accounts and contact information
 - Service users' login and requests
 - Maintain users' status, IP addresses, port numbers, ...

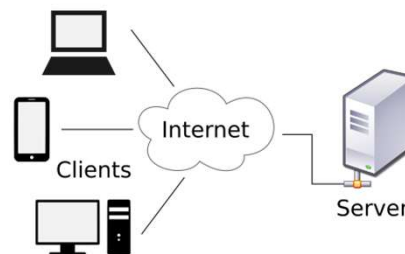
EECS22L: Software Engineering Project in C, Lecture 8

(c) 2018 R. Doemer

3

Project 2: Software Architecture

- Client-Server Software Architecture
 - Software applications communicating via the Internet



Source: David Vignoni (LGPL, wikipedia.org)

- Server: provides a service function to one or more clients
- Client: initiates requests for service
- Internet: communication network to exchange messages

EECS22L: Software Engineering Project in C, Lecture 8

(c) 2018 R. Doemer

4

Project 2: Client-Server Architecture

- Introduction to Internet Communication
 - Basic terms and concepts
 - Point-to-point communication, often designed as client-server model
 - *Client* initiates communication, sends a *request*
 - *Server* waits, services client request, sends a *response*
 - Client and server are software *processes* executing on *hosts*
 - Hosts are typically *distributed* (networked), but can also be identical
 - *Sockets* represent a *network connection* between two processes
 - Sockets operate *bidirectional* (both sides can *send* and *receive*)
 - *Stream sockets* implement *connection-oriented* semantics
 - Data is transported reliably in-order, without loss or duplication
 - *Transmission Control Protocol over Internet Protocol, TCP/IP*
 - Hosts have Internet Protocol (IP) *addresses* and *ports*
 - Host `crystalcove.eecs.uci.edu` has IP address `128.200.85.14`
 - Ports below 1024 are reserved (e.g. port 80 for web browsing)
 - Ports above 2000 are typically “free” for application-specific use

EECS22L: Software Engineering Project in C, Lecture 8

(c) 2018 R. Doemer

5

Project 2: Client-Server Architecture

- Introduction to Socket Communication
 - Sockets Tutorial
 - http://www.linuxhowtos.org/C_C++/socket.htm
 - <http://www.linuxhowtos.org/data/6/client.c>
 - <http://www.linuxhowtos.org/data/6/server.c>
 - Reference: Linux manual pages
 - `man socket`
 - `man select`
 - `man select_tut`
 - Extended client-server example:
 - `~eecs22/SocketTutorial.tar.gz`
 - `client2.c`
 - `server2.c`
 - `Makefile`
 - `README`
 - Online demonstration!

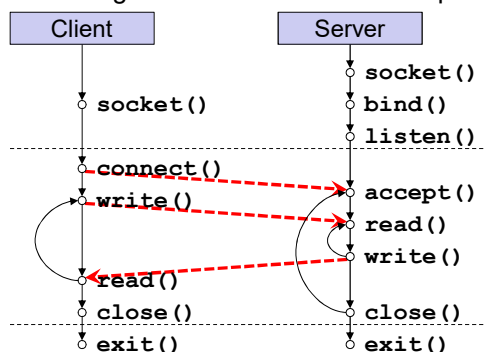
EECS22L: Software Engineering Project in C, Lecture 8

(c) 2018 R. Doemer

6

Project 2: Client-Server Architecture

- Discussion on Socket Communication
 - Sequence Diagram for client-server example



- This simple example can handle only one client at a time!
 - The `listen()` function keeps connecting clients in a back log where they need to wait until other clients have closed their connection

EECS22L: Software Engineering Project in C, Lecture 8

(c) 2018 R. Doemer

7

Project 2: Client-Server Architecture

- Communication Example: *Initial Chat App*
 - Client: `Hello!`
 - Server: `ERROR invalid message "Hello!"`

 - Client: `CREATE_ACCOUNT Albert Einstein AE`
 - Server: `OK AE = Albert Einstein`

 - Client: `GET_CONTACTS AE`
 - Server: `OK IN = Isaac Newton, NT = Nikola Tesla`

 - Client: `SEND_MESSAGE NT Hello Nikola, how are you?`
 - Server: `OK`

 - Client: `RECEIVE_MESSAGES`
 - Server: `OK NT="I'm fine, thanks!", END_OF_MESSAGES`

EECS22L: Software Engineering Project in C, Lecture 8

(c) 2018 R. Doemer

8