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
  - Application components
  - Application communication
- Introduction to Socket Communication
  - Basic terms and concepts
  - Client/server example

Project 2

- Introduction
  - Taxi Cab Management System
  - Story:

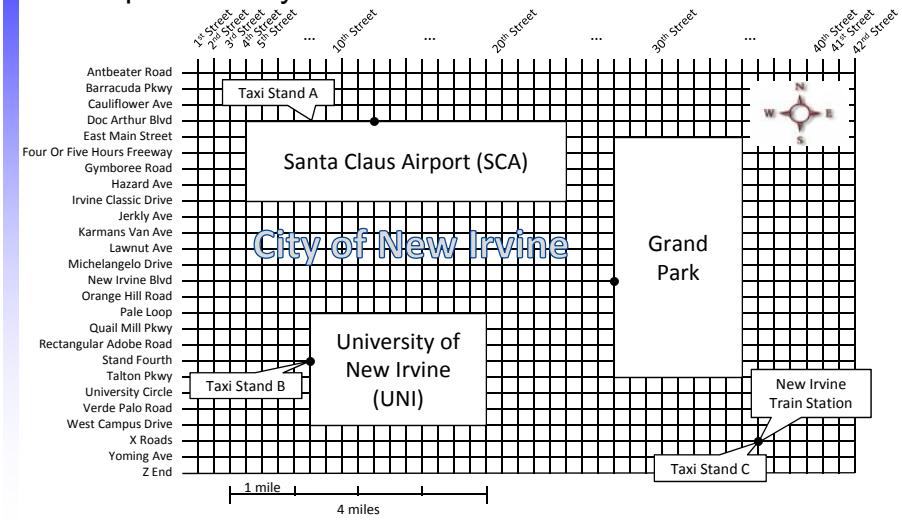
    *The **City of New Irvine** is soliciting proposals for the development of a new fully-automated taxi cab management service that can optimally coordinate a number of taxi cabs serving the individual public transportation needs of the population and visitors of the city.*

    The City of New Irvine expects transportation by taxi to be very flexible and efficient for individual people and small groups. Customers may request rides from and to every street corner within the city limits, to be served at the time of request or at a defined later time as specified by the customer.
  - Task:

    Design a networked computer server where incoming customer requests are automatically processed and taxi drivers can fairly compete for trips and compensation. At the same time, customers' expectations of punctuality and quick trips are to be met to the best extend possible.

Project 2

- Map of the City of New Irvine

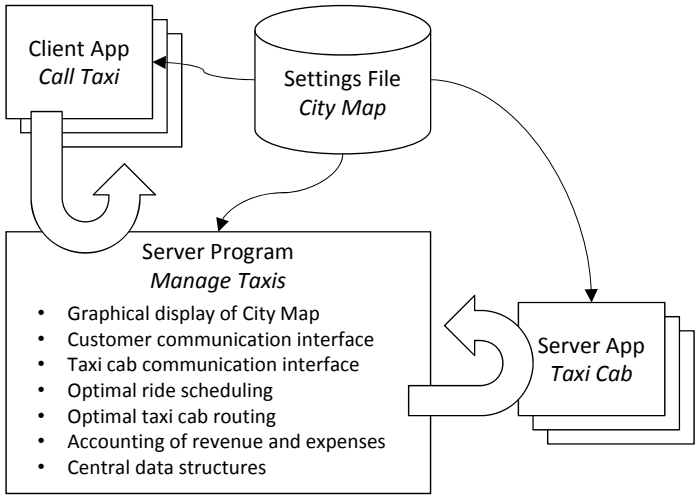


Project 2

- Goal and Requirements
  - Customer rides from/to every corner or landmark
  - Customer rides now or at specified later time
  - Up to about 20 minutes ride, obeying 45 MPH speed limit
  - Taxi stands
    - Up to 12 cab cars
    - Capacity limousine 3 passengers, van 6 passengers
  - Fair and easy transportation of individuals and small groups
  - Punctuality and quick trips
  - Taxi costs regulated
    - Fixed base fee $3.75 (for pickup and return)
    - Manhattan-distance dependent charge of $2.00 per mile
    - Taxi drivers earn $0.20 per block driven (incl. pickup and return)
    - Management system shall generate revenue!

Project 2

- Overall System Specification



Project 2

- System Management Settings
  - Settings file (extendable)

```
STREETS EAST-WEST:
Antbeater Road, Barracuda Pkwy, Cauliflower Ave, Doc Arthur Blvd, East Main Street,
Four Or Five Hours Freeway, Gymboree Road, Hazard Ave, Irvine Classic Drive, Jerkly
Ave, Karmans Van Ave, Lawnut Ave, Michelangelo Drive, New Irvine Blvd, Orange Hill
Road, Pale Loop, Quail Mill Pkwy, Rectangular Adobe Road, Stand Fourth, Talton Pkwy,
University Circle, Verde Palo Road, West Campus Drive, X Roads, Yoming Ave, Z End.

STREETS NORTH-SOUTH:
1st Street, 2nd Street, 3rd Street, […street names omitted for brevity…], 42nd Street.

LANDMARKS:
University of New Irvine (UNI) AT Stand Fourth AND 8th Street BOUNDARIES NORTH Pale
Loop EAST 19th Street SOUTH West Campus Drive WEST 8th Street,
 Santa Claus Airport (SCA) AT Doc Arthur Blvd AND 12th Street BOUNDARIES NORTH Doc
Arthur Blvd EAST 24th Street SOUTH Irvine Classic Drive WEST 4th Street,
Grand Park AT New Irvine Blvd AND 27th Street BOUNDARIES NORTH East Main Street SOUTH
Talton Pkwy EAST 35th Street WEST 27th Street,
New Irvine Train Station AT X Roads AND 36th Street,
Taxi Stand A AT Doc Arthur Blvd AND 8th Street,
Taxi Stand B AT Stand Fourth AND 8th Street,
Taxi Stand C AT X Roads AND 36th Street.
```


Project 2

- Introduction to Socket 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 requests, sends a *response*
    - Client and server are software *processes* executing on *hosts*
    - Hosts are typically *distributed* (different), but may also be the same
    - *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 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. 80 for web browsing)
      - Ports above 2000 are typically “free” for application use

Project 2

- Introduction to Socket Communication
  - Simple client/server example
    - **http://www.linuxhowtos.org/C_C++/socket.htm**
    - **http://www.linuxhowtos.org/data/6/client.c**
    - **http://www.linuxhowtos.org/data/6/server.c**
  - Extended client/server example
    ~eecs22/SocketTutorial.tar.gz
    - **client2.c**
    - **server2.c**
    - **Makefile**
    - **README**
  - ➢ Online demonstration!

Project 2

- Protocol *"Call Taxi"*
  - Client Request (in Backus-Naur Form, BNF)

```
<request> ::= REQUEST RIDE FROM <location> TO <location> <special>*
<location> ::= CORNER <street name> AND <street name>
        | <landmark name>
<special> ::= [<time>]
        | [FOR <number> PERSONS]
        | [FIRST CLASS]
<time> ::= AT <hours>:<minutes>
        | NOW
```

  - Server Response

```
<response> ::= OK PICKUP AT <estimated time> COSTS <amount>
        | DECLINED <reason>
        | ERROR <message>
<amount> ::= $<dollars>.<cents>
```

Project 2

- Protocol *"Taxi Cab"* (revised)
  - Client Request (in Backus-Naur Form, BNF)

```
<message> ::= <request>
            | <instruction>
<request> ::= REQUEST POSITION
<instruction> ::= GO <direction> <direction>*
            | PICKUP <number> PERSONS
            | DROPOFF <number> PERSONS
<direction> ::= NORTH | EAST | SOUTH | WEST
```

  - Server Response

```
<response> ::= OK POSITION AT <location>
            | OK DRIVING <direction> <direction>*
            | OK PICKUP | OK DROPOFF
            | ERROR POSITION <message>
            | ERROR DIRECTIONS <message>
            | ERROR PICKUP <message>
            | ERROR DROPOFF <message>
            | ERROR <message>
```