

# CS 145 Problem Set 2-3

## Objectives

- Formulate an application-layer protocol that will satisfy a certain service.
- Implement an application/user agent that will use the formulated protocol
- Apply socket programming in practice.

## Introduction

For this problem set, you have to create a multiplayer game with a pure client-server architecture. Your application must run on multiple end systems, and the clients must be able to interact with other clients. To enable the hosts/processes to communicate to each other, you must use the socket API in developing your application.

## Protocol

Before creating the multiplayer game, you have to formulate an application-layer protocol that will at least satisfy the following services:

- A connected user must have an identifier. Users that are connected to the same host must have distinct identifiers.
- A connected user must have the option to provide an alias/username. Users that are connected to the same host may have the same alias/username.
- A connected user must be able to receive messages from other users that are connected on the same host.
- A user must be able to send messages to all users, or to selected users.
- Aside from aliases and messages, a user must also have the option to share other kinds of information (e.g. status messages).
- A user must have the option to voluntarily disconnect.
- A user must be notified when another user disconnects.

The number of commands (e.g. in HTTP, you have GET, POST, etc.), status codes (e.g. in HTTP, you have 404, 500, 200, etc.), and actions that you have to provide is up to you. Just make sure that your protocol will satisfy the requirements stated above. Indicate the port number and the underlying transport layer protocol. It is up to you if you will create a protocol that maintains the state of communication between the client and the server.

## Application

Your multiplayer game must run on multiple hosts, and to make this possible, you must use the socket API of the language that you will use. Use the protocol that you have formulated to facilitate the communication of the clients. You may extend your protocol so that it can provide some services that are specific to your game. Just make sure that your protocol will provide the services stated above.

## Mechanics

- You have the option to work on this problem set individually or work as a group. The maximum number of members per group is 3.
- You may use any programming language and tools (Graphics Libraries, XML, SQL) in implementation as long as you will use the socket interface that is provided by the language.
  - If you will implement your application in Python, use the socket module.
    - You may start with your code from the machine exercises.
  - If you will implement your application in Java, use the classes that are provided in the java.net package.
- You may use other tools (such as graphics libraries, XML and SQL for data, etc.) if you want.
- The minimum number of clients is 1.

## Submission, Presentation and Grading

This problem set will be graded as follows:

Protocol – 30%

Implementation – 50%

Those who have submitted on time must present their work to all CS145 students for the remaining 20%. Presentation is on March 28, 2014.

The deadline of submission is on 22 March, 2014. You may submit your work via email. Those who submitted late will get 0 in the presentation score even if they are able to present (you may still present if you are proud of your work hehe).

This problem set is 15% of your final grade (50% of the problem set score). Good luck!

## Deliverables

Aside from the source code, you have to submit the following:

1. Documentation of the protocol
  - Message Format
  - Actions Taken by the client/server
2. Documentation of the application
  - Instructions on how to build and run the application
  - Minimum requirements and dependencies

## Guidelines

- For the interaction to become possible, the messages must be relayed by the server from the sending client to the receiving clients (which may include the sender).
- The server must be able to track all connected users and their corresponding sockets.

- In selecting users, you have to use an identifier. The aliases/usernames only provide the end user an easier way of identifying and communicating with other end users.
- In responding to requests, the EASIEST implementation is to respond to one request at a time. Your knowledge in scheduling will be helpful for this problem.
- Your instructor is available for consultation.