

Topic: client/server, concurrency

**Reading:** Abelson & Sussman, Section 3.4

### Homework:

These exercises use the Instant Message program, found in the following files:

`~cs61a/lib/im-client.scm`

`~cs61a/lib/im-server.scm`

`~cs61a/lib/im-common.scm`

1. Invent the capability to send a message to a list of clients as well as to a single client. Do this entirely in the client program, so what actually goes to the server is multiple requests.
  2. Invent the capability to broadcast a message to every client. Do this by inventing a `broadcast` command that the server understands.
  3. Could #1 have been done with the server doing part of the work? Could #2 have been done entirely in the client code? Compare the virtues of the two approaches.
  4. Invent the capability of refusing messages from specific people. The sender of a refused message should be notified of the refusal. Decide whether to do it entirely in the client or with the server's cooperation, and explain why.
  5. Why is the 3-way handshake necessary when connecting to the server?
- Abelson & Sussman, exercises 3.38, 3.39, 3.40, 3.41, 3.42, 3.44, 3.46, 3.48

### Extra for experts:

Using the Instant Message program as a starting point, write a mail server and client. The mail server should maintain a database of messages for all users. (This can just be a list; don't worry about efficient lookup.) The client should be able to do the following:

`(mail username message)`

`(get-mail)`

`Get-mail` should return a list of messages, which should be deleted from the server.

If you want, you can improve this in several ways: Make deletion from the server be explicitly requested by the client, invent a subject header (another argument to `mail`) and have the client show just headers in `get-mail` and provide another command to read the text of a specific message, and so on.

- Read Section 3.3.5 and do exercises 3.33–3.37.

---

Unix feature of the week: `&`, `^Z`, `fg`, `bg`, `jobs`, `kill`

Emacs feature of the week: `M-x abbrev-mode`, `M-x add-mode-abbrev`