

Network Protocols

A protocol is a specification of the "message behavior" between communicating programs

Example:

- The format of messages between a client and server
- The order of queries and responses, e.g.

1st { → "How are U?"
 ← "I am OK"

2nd { → "Do U process version 1.0?"
 ← "Yup"

 etc.

EXAMPLE: The HTTP protocol.

there is an `httplib` module in Python (see p. 419) — better than my `http-set.py`, but doesn't show what's going on.

Also, for server, there is a `BaseHTTPServer` module (see p. 440)

Concurrency Without Threads

The `select()` method:

`i, o, e = select.select(i, o, e)`

list of "input" sockets

"output" sockets

exceptions

`select()` asks and answers the question:
 which sockets are immediately ready for send, recv, or accept without blocking?

Programs can therefore wait until a client is trying to connect or another client ~~is accepted~~ has sent a message to be processed, whichever occurs first.

See: `select_top_server.py`