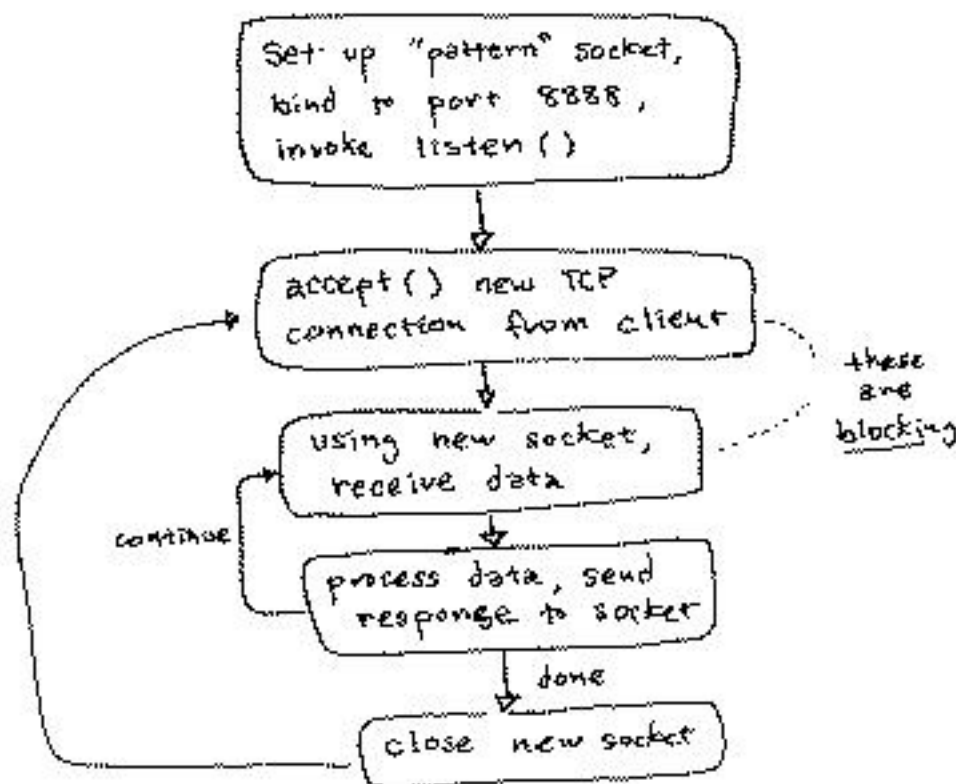


Recall TCP Server Logic



blocking calls can force server to wait

⇒ while server is waiting to receive data, no other client is able to connect to server

so we need a threading server
(each client gets new thread)

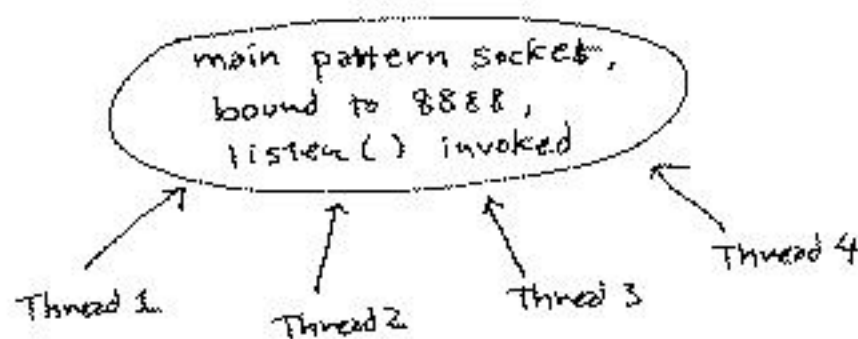
Idea of threaded TCP server

while one thread waits on `accept()`, other thread waits on `recv()`

(actually we generalize to many threads)

Implementations

1. "quick and easy"
(possibly unsafe)

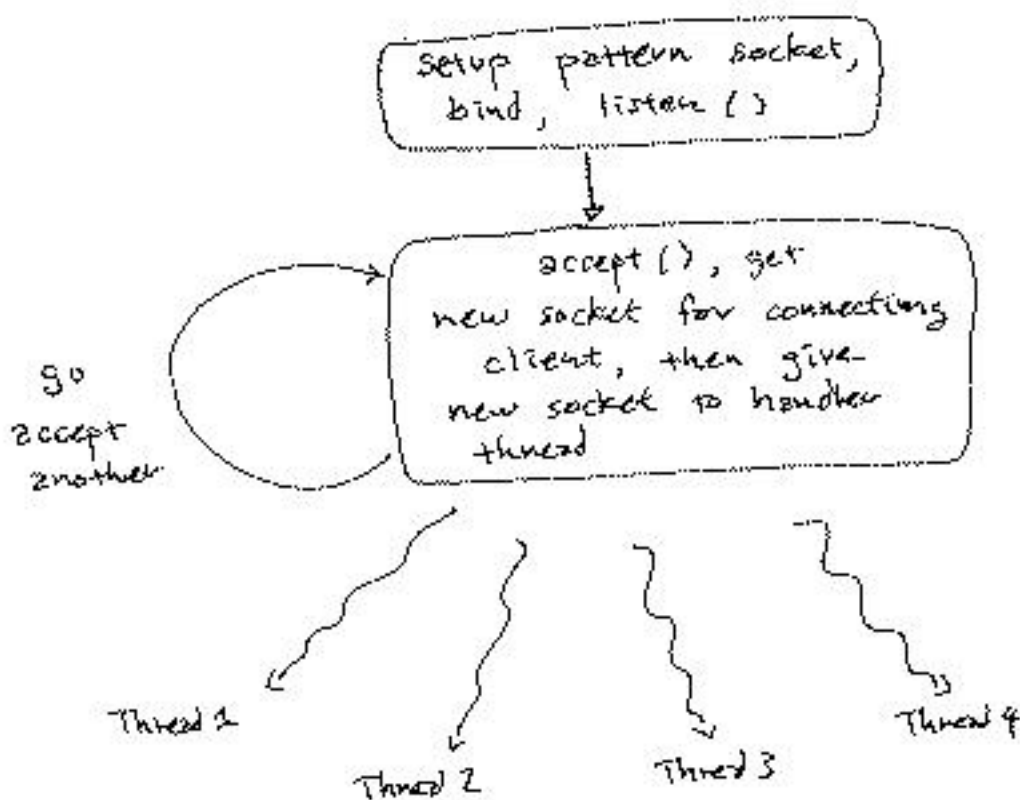


each is trying to `accept()` concurrently, then use the new socket

- will only work if OS does some kind of mutual exclusion (very likely true) so that two threads don't get same client

2. Controlled, single acceptor / multiple handler pattern

main thread:



these are "handler" threads that do the `recv()` and `send()` operations using the new sockets from main thread.

Tools and Techniques

- The queue class/object
- How to properly recv() and send()
- How to read from files
- How to get keyboard input
- How to extract messages from a stream
- telnet command, unix command execution