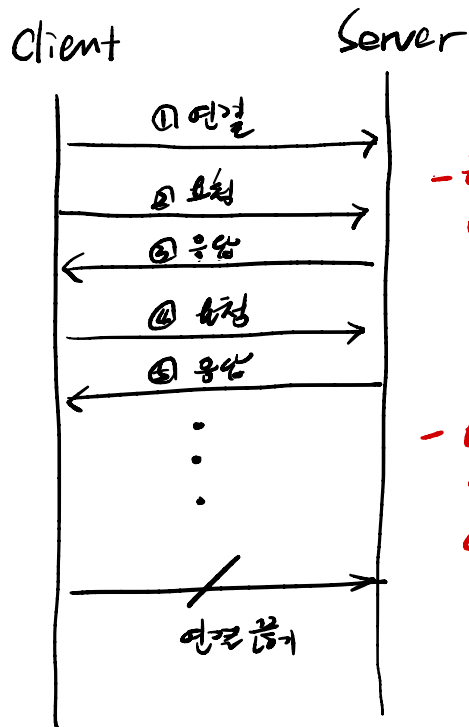


43. Stateless 통신 방식

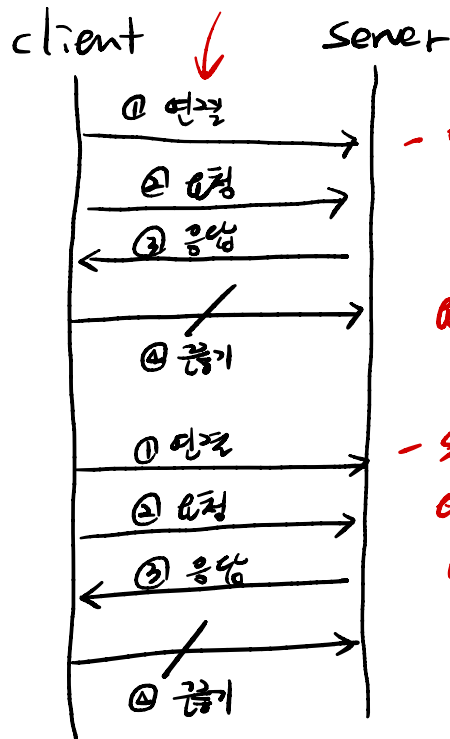
① Stateful



- 한번 연결하면 요청/응답을 연속해서 여러번 수행할 수 있다
- 요청/응답을 수행하며 많은 시간에도 연결 상태를 계속 유지
- ↓
- 대량의 클라이언트 연결을 지원하지 못한다

② Stateless

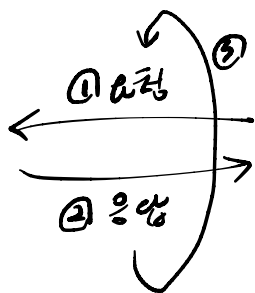
★ 연결하는데 시간이 많이 걸린다



- 요청한 amount 마다 매번 연결 수행
- ↓
- 요청 처리시간이 길어진다
- Stateful 방식보다 더 많은 클라이언트 요청을 처리

* Stateful vs Stateless

① Stateful



클라이언트1

클라이언트2

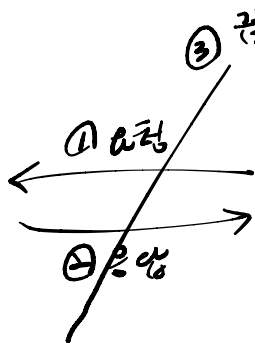
클라이언트3

...

예) 은행 입출금현물의

↳ 먼저 연결된 클라이언트가 연결을 끊을 때까지
다음 클라이언트는 기다려야 한다.

① Stateless



클라이언트1

클라이언트2

클라이언트3

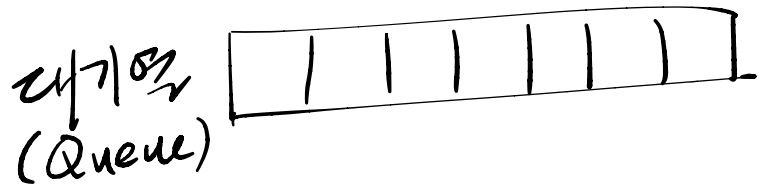
...

예) 114 안내

↳ 한번 요청/응답이 이루어지면 꼭시 연결을 끊는다!
다음 클라이언트의 기다리는 시간은 짧아진다.

* 서버 만들기

① `new ServerSocket(8888)` → { - 랜카드에 연결한, 정보 생성
- 대기열을 준비시킨다



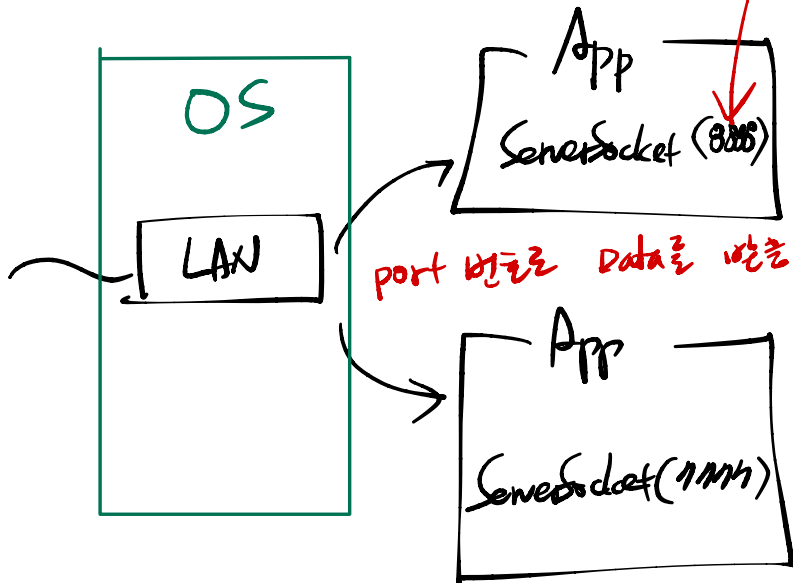
↓
② `accept()` → { - 대기열에서 연결한 클라이언트 정보를 꺼낸다
- Socket 객체 생성 및 반환
- 대기열이 비어있으면 리턴하지 않고 기다린다.

* port

↳ port 번호를
중복된 것을
알린다.
(0 ~ 49151)

서버측 포트번호는
고객번호가 정한다

보내는이	IP/port
받는이	IP/port
Data	

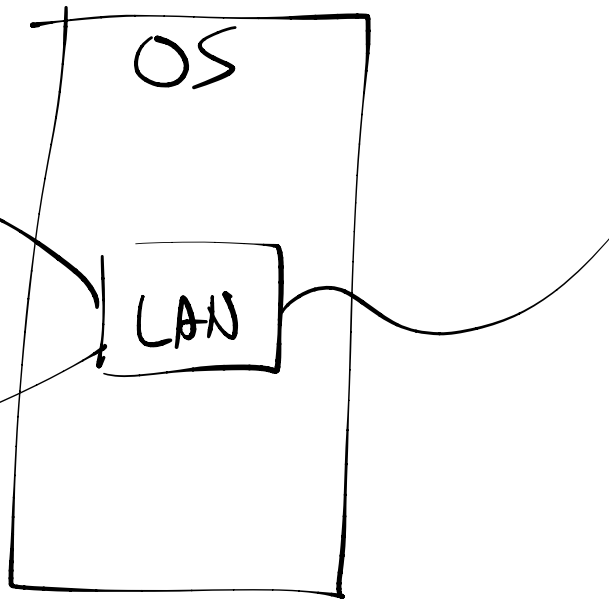
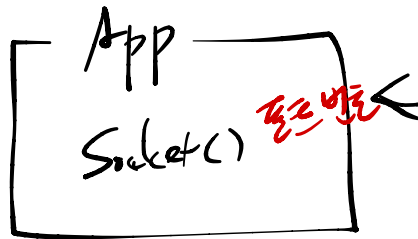
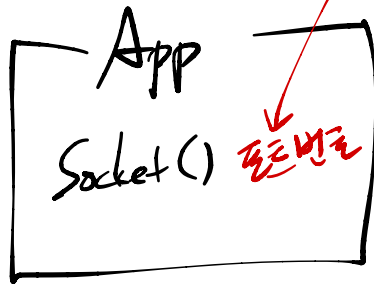


port 번호로 Data를 받는 App은 정해!

HI HI

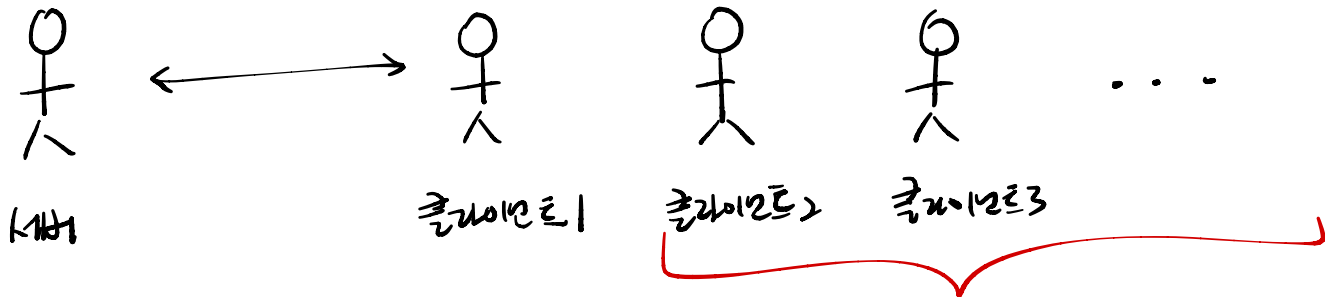
* port

클라이언트 포트번호로
OS가 자동으로 번호를 = 기계적으로 정해 주기 때문
(49152 ~ 65535)



클라이언트

* 순차 처리 방식의 문제점



Client가 순차적으로 연결됨



먼저 연결된 클라이언트가 연결/종료될 때까지 기다려야 한다.



동시 작업 불가!