1. Client – Server paradigm
   1. 클라이언트
      1. 사용자가 직접 호출하는 프로그램(web browser)
      2. 서버에 네트워크로 작업을 요청
   2. 서버
      1. 클라이언트의 요청에 따른 작업 수행
      2. 자료저장, 자료공유, 대용량 데이터 처리 (하둡)
2. Application Layer Protocol
   1. 어플리케이션 간 통신
   2. SOCKET (application 과 transport layer를 연결하는 인터페이스) 에 대한 이해
      1. 하부 구조 : TCP, UDP ( transport layer )
      2. 성능 이슈 : 지연시간, 대역폭
      3. 성능을 극대화 하기 위해서는 하부구조를 잘 알아야 한다.
   3. HTTP reply
      1. Status line
         1. HTTP version
         2. 3 digit response code (200 정상종료, 4xx 접속 오류, 5xx 서버 오류)
         3. Response phrase 아무 말이나 지껄임
      2. 응답 방식
         1. char output[] = “쏼라 쏼라”; send(connfd, output, strlen(output), 0);
         2. 결국엔 byte Stream교환
            1. Browser

HTTP request byte stream 으로 만들어서 보냄.

들어온 HTTP response를 해석해서 출력

* + - * 1. Server

블라블라

* + - 1. TCP stream을 통해서 전달
         1. 어떤 것이 먼저 올지 알 수 없다.
         2. segment단위로 buffer를 한 개씩 전달
         3. segment들이 IP Packet으로 변해서 전달됨
      2. TCP
         1. 정보를 stream방식으로 안전하게 전달함
         2. First in fitst out 을 보장한다.
         3. 번호 붙여서 보내고 빠진 번호에 대해서 다시 보내달라고 요청함
         4. Sequencing, timeout, retransmission
         5. Segmentation Encapsulation

header : 인덱스, 보내는 곳, 보낼 곳, 통신방식 등등..

TCP는 Ip Layer를 사용해서 패킷을 잘라 줌.

* + - 1. IP packet
         1. Version, Header Length, Type of Service(latency, bandwidth), Length
         2. TTL(지나온 라우터 정보), Protocol, SrcAddr, DstAddr