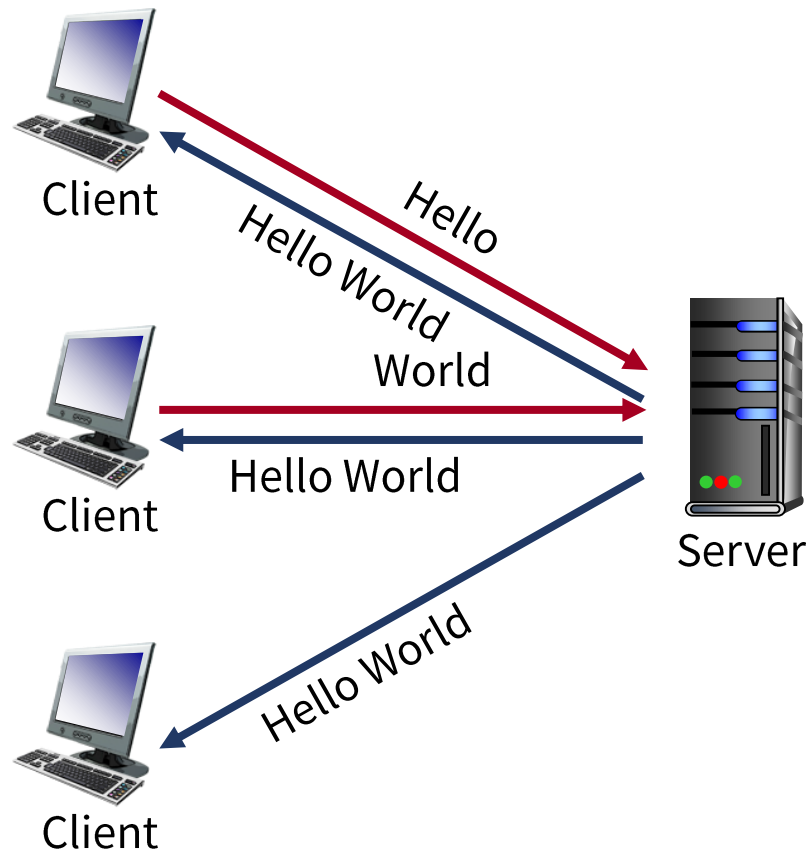


# 과제 3: 채팅 서버 / 클라이언트 제작

## ■ 채팅 서버 / 클라이언트

- 서버에 접속한 모든 클라이언트가 함께 대화
  - 클라이언트가 전송한 대화를 서버에 접속한 **모든 클라이언트에게 전달**



# 과제 3: 채팅 서버 / 클라이언트 제작

## ■ 구현해야 하는 기능

### ● 서버

- Thread를 이용하여 각 클라이언트와의 통신 처리
- 접속된 클라이언트 목록을 유지 (클라이언트 이름과 IP 주소)
- 클라이언트로부터 메시지를 받으면 접속된 모든 클라이언트에게 전달

### ● 클라이언트

- 1:1채팅 클라이언트와 비슷한 기능 수행 (Thread 사용)
- 서버에 접속하여 타이핑한 내용 서버로 전송
- 서버로부터 받은 메시지는 화면에 출력

# 과제 3: 채팅 서버 / 클라이언트 제작

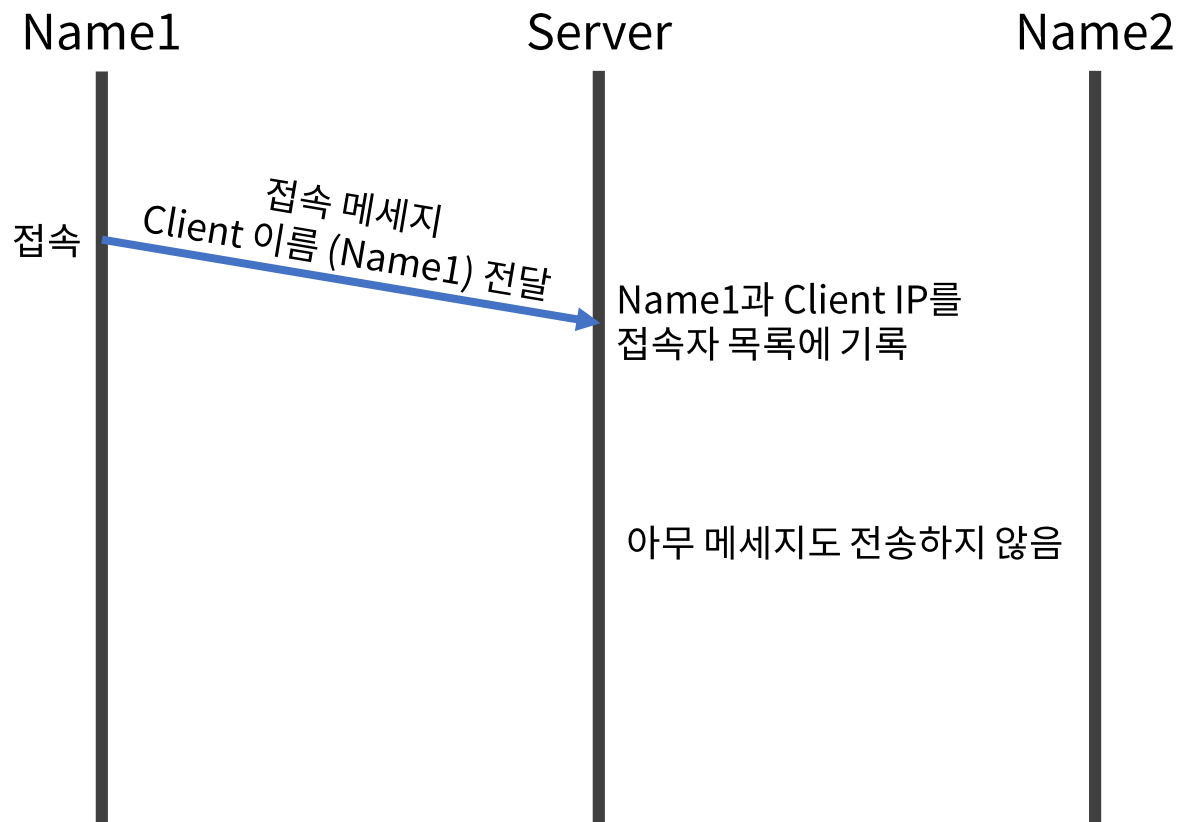
## ■ 클라이언트 명령어

- 서버 접속 후 특수 동작을 위한 명령어
  - 일반 채팅 프로그램과 같이 슬래시(/)로 시작하는 명령어 사용
  - /q
    - 클라이언트 종료
    - 서버에 종료 메시지 전송
    - 서버는 종료 메시지를 받으면 접속자 목록에서 해당 클라이언트를 삭제

# 과제 3: 채팅 서버 / 클라이언트 제작

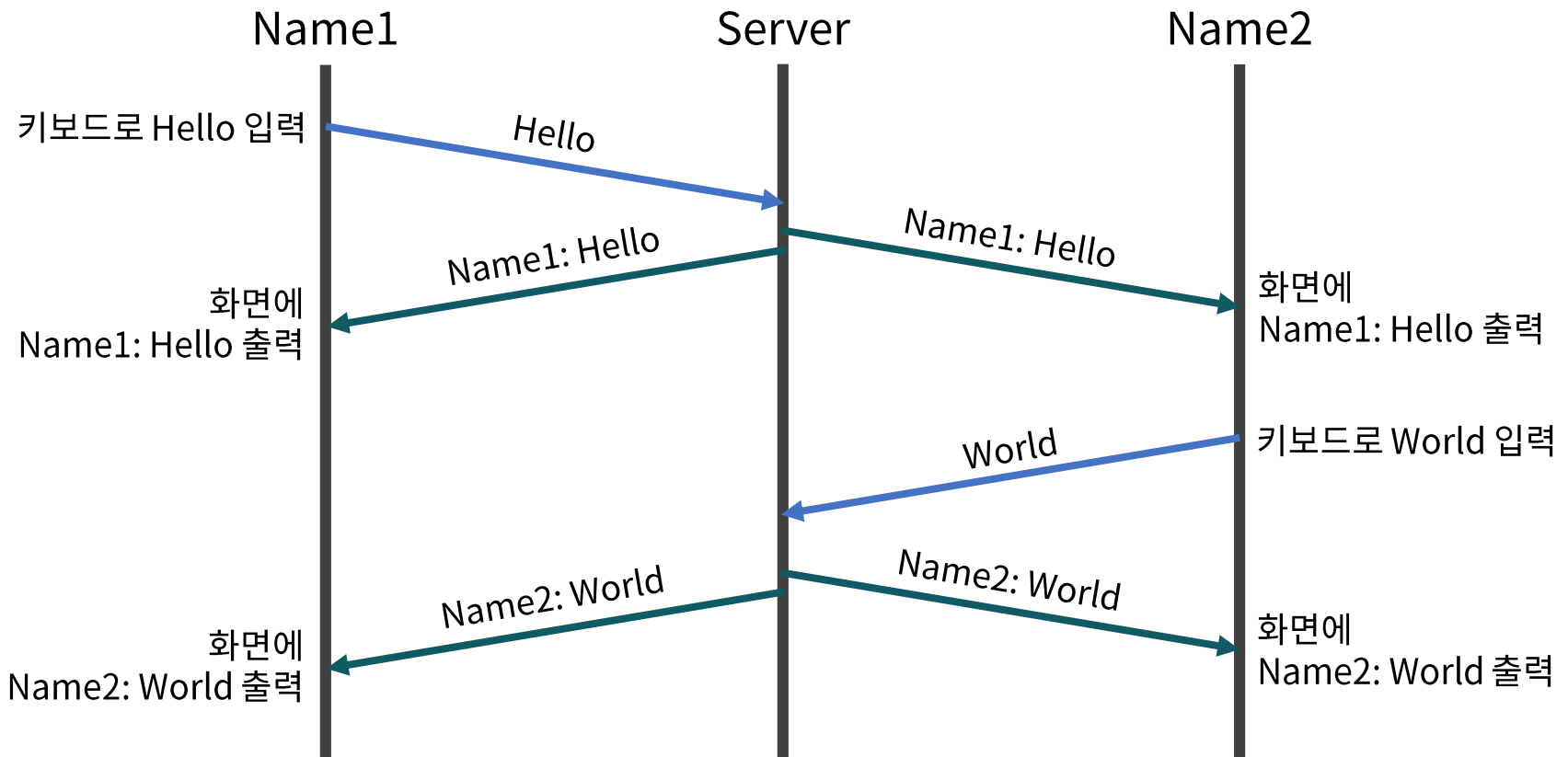
## ■ 전송 프로토콜

- Client 접속 (이름 : Name1)



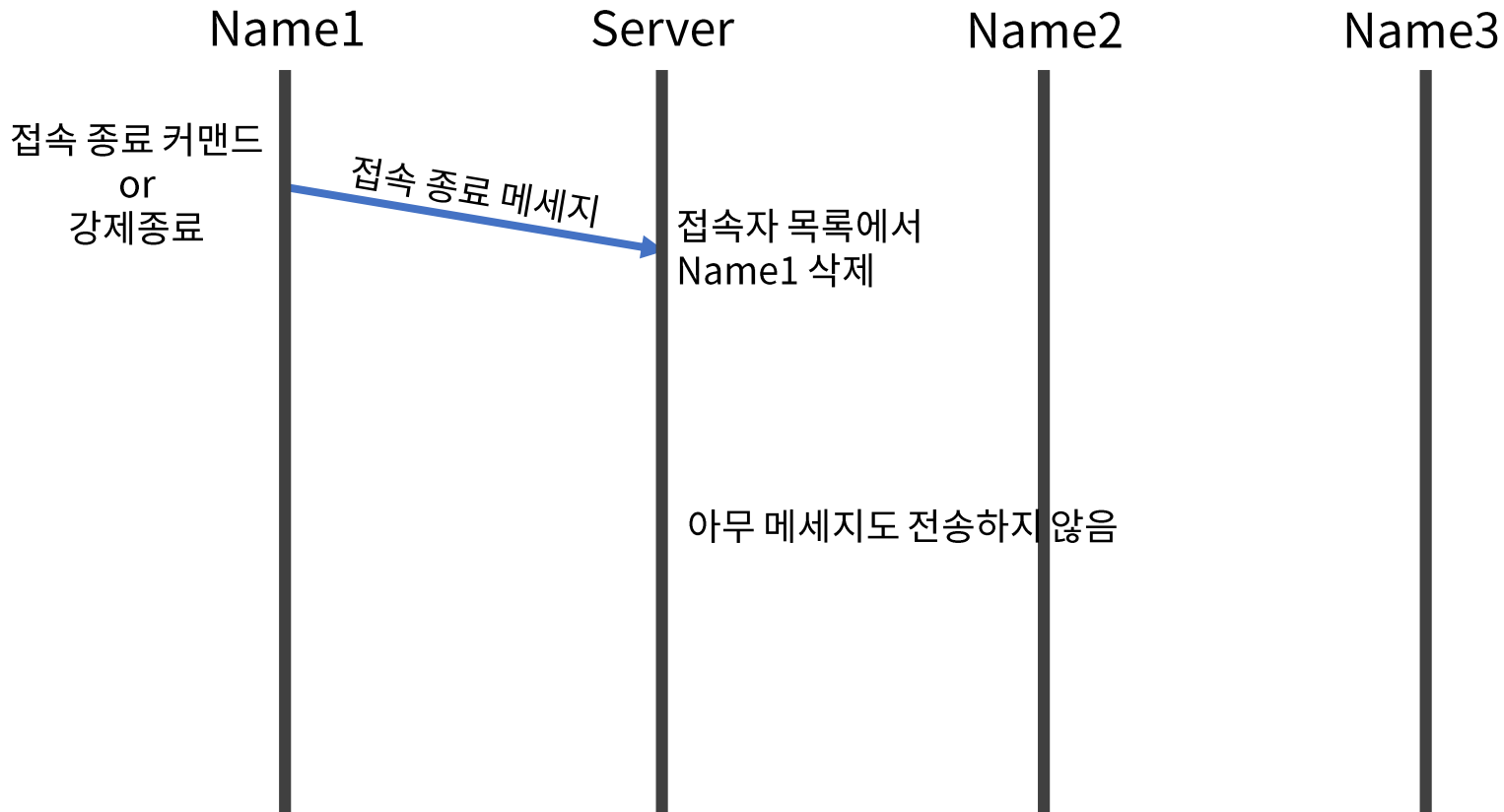
# 과제 3: 채팅 서버 / 클라이언트 제작

- 채팅 메시지 전송
  - Name1 과 Name2가 각각 "Hello", "World" 전송



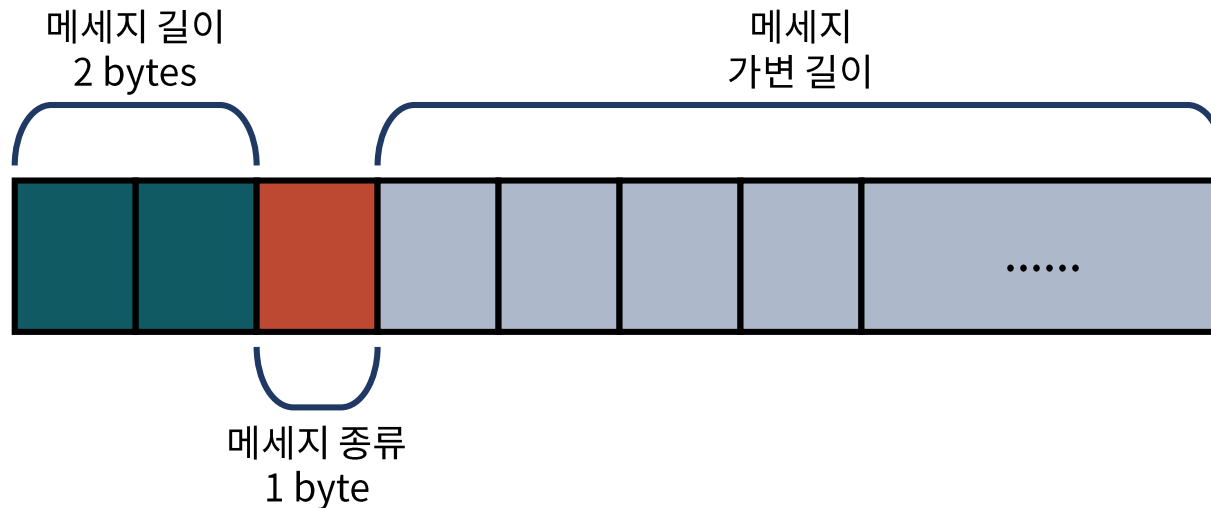
## 과제 3: 채팅 서버 / 클라이언트 제작

- Client 접속 종료
  - Name1이 접속 종료



# 과제 3: 채팅 서버 / 클라이언트 제작

## ■ 메시지 형식



- 메시지 길이 : 3번째 항목인 메시지의 길이 (몇 byte인지, **Big endian**)
- 메시지 종류 : 이 메시지의 종류가 무엇인지 표시

'm'	일반 채팅 메시지
'q'	접속 종료 메시지
'c'	접속 메시지

- 메시지 : 실제 전달될 내용

# 과제 3: 채팅 서버 / 클라이언트 제작

## ■ 메시지 사용 예

- 일반 채팅 메시지
  - Client → Server (Hello World라고 전송)

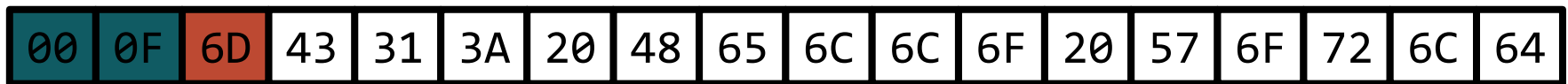


- Server는 받은 'Hello World' 메시지를 다른 client들에게 전달



## 과제 3: 채팅 서버 / 클라이언트 제작

- Server → Client ('Hello World'를 이름이 'C1'인 유저에게 받았을 경우)



메세지 길이  
15 bytes = 0x000F

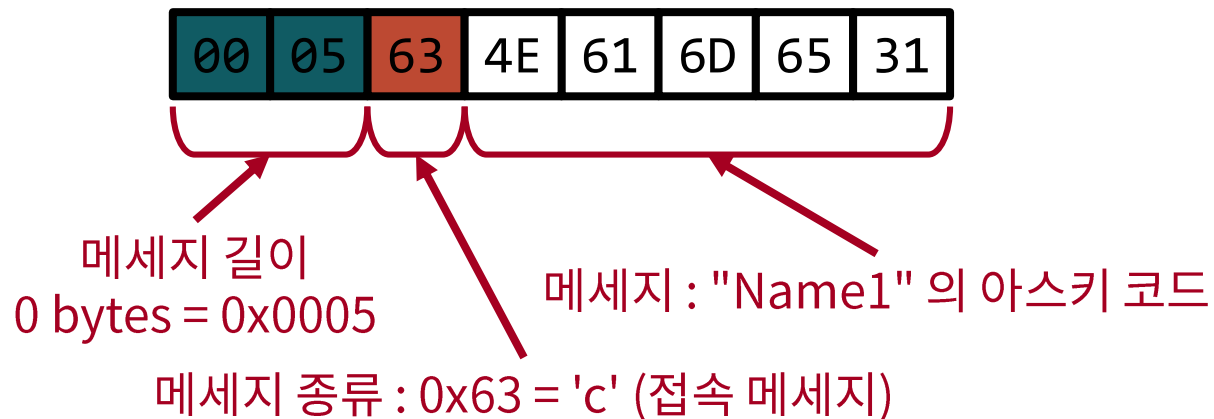
메세지 종류 : 6D = 'm' (일반 채팅 메세지)

메세지 : "C1: Hello World"의 아스키 코드  
※ 메세지를 보낸 사용자 이름을 앞에 추가해야 함

- 위의 메세지를 받은 Client는 화면에 메세지 출력

## 과제 3: 채팅 서버 / 클라이언트 제작

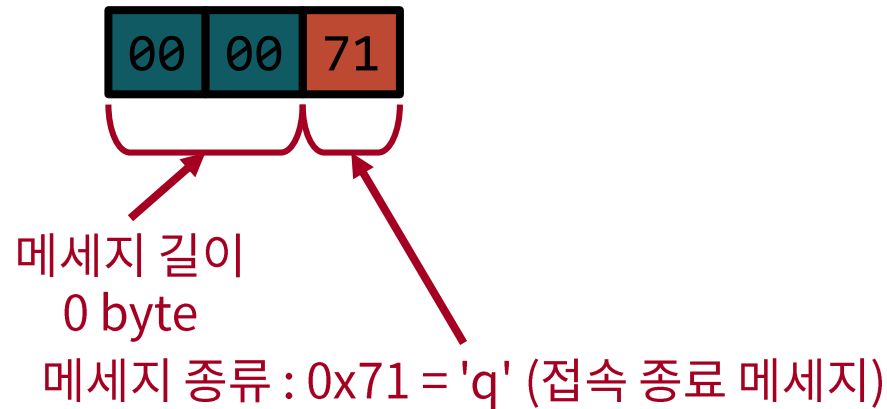
- 접속 메시지
  - Client → Server (Client의 이름 전송)
    - 이름이 "Name1"인 Client가 처음 서버에 접속했을 때 전송



- Server는 메세지를 전송한 client의 이름과 IP를 접속자 목록에 기록

## 과제 3: 채팅 서버 / 클라이언트 제작

- 접속 종료 메시지
  - Client → Server
    - 특별히 따로 메시지를 전송하지 않음 (TCP의 종료 프로세스 사용)
    - 서버는 메시지를 받으면 클라이언트 목록에서 해당 클라이언트 삭제



# 과제 3: 채팅 서버 / 클라이언트 제작

## ■ 실행

- Server

- 포트 번호가 12345일 경우

```
# ./chat_server 12345
```

- Client

- Server의 IP가 192.168.0.10, 포트 번호가 12345, Client의 이름이 Name1일 경우

```
# ./chat_client 192.168.0.10 12345 Name1
```

# 과제 3: 채팅 서버 / 클라이언트 제작

## ■ 제출 방법

- 소스 코드 (.h, .c 파일)
  - 파일을 여러 개로 나눠 제출해도 됨
  - 여러 개로 나눌 경우 Makefile 함께 제출
    - make 입력 시 server와 client 둘 다 컴파일 되도록 작성
- 구현 내용 작성 파일 (.pdf)
  - 실행 방법
  - 접속이나 데이터 전송 등을 어떻게 구현했는지 작성
  - 특히 Thread 관련 부분과 접속자 관리 부분을 상세히 작성
- 위의 파일들을 스마트 캠퍼스의 과제 항목에 업로드

## ■ 제출 기한

- 12월 7일 23:59 까지