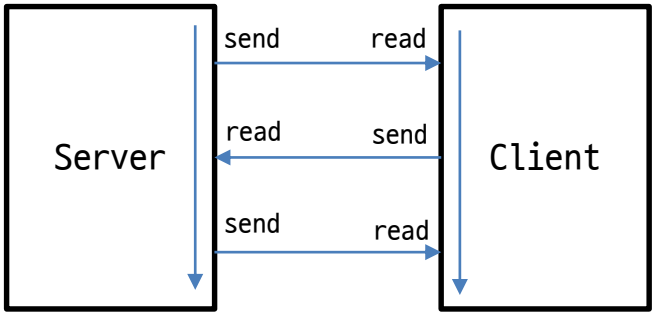
Server Programming Practice 3

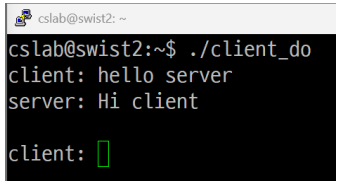
CSLAB
Jo, Minwoo

Messenger

문자열 송-수신

- 일반적으로 서버-클라이언트 소켓 연결은 “양방향” 연결
  - 즉, 서버는 클라이언트에게 메시지를 보내고, 받을 수 있으며 마찬가지로 클라이언트는 서버에게 메시지를 보내고, 받을 수 있음
  - send(sock, buff, strlen(buff), 0)
    - 사용할 소켓, 보낼 메시지(버퍼), 메시지 크기, 플래그(0은 생략)
  - read(sock, buff, BUFF_SIZE)
    - 사용할 소켓, 받을 메시지, 받을 버퍼의 최대크기
      - » strlen으로 받지 않는 이유 : 안정성
- 그러나 read와 동시에 send할 수 없음
  - 단일 쓰레드의 한계

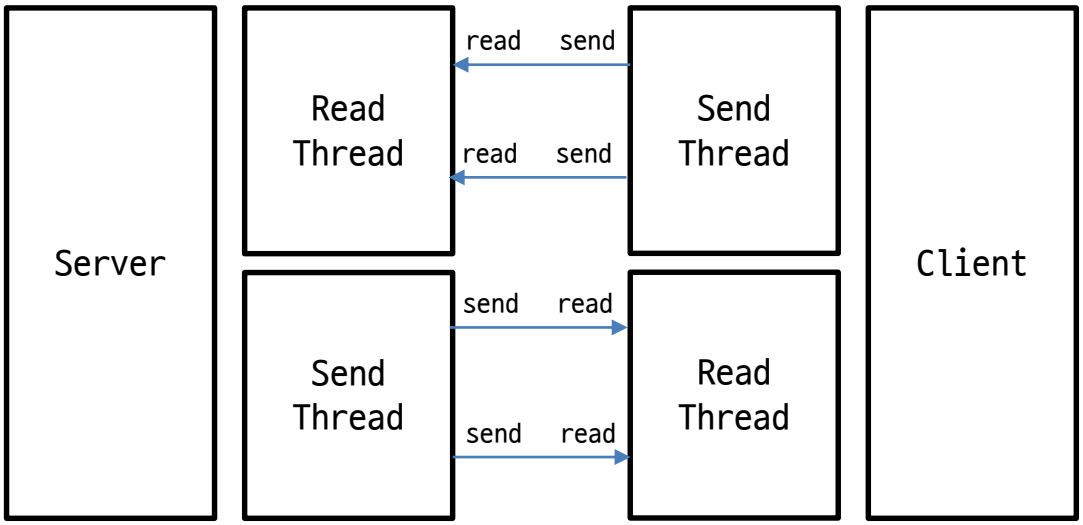




Messenger

멀티 쓰레드 활용

- 프로그램 구조 변경
  - 소켓 연결 까지는 동일, 그러나 read와 send를 다른 쓰레드로
  - 해당 방식 사용시 하나의 쓰레드는 read, 하나의 쓰레드는 send하므로 대기의 필요 없이 메시지 송수신 가능




```
cslab@swist1:~/ta3$ ./server_2
server: test
server: message
server: client : yes
client : I
client : can
client : do
client : too
```

```
cslab@swist2:~$ ./client_2
client: server : test
server : message
yes
client: I
client: can
client: do
client: too
client:
```

Today's Goal

Part 1 : 메신저 만들기

- 단일 쓰레드 활용

Part 2 : 메신저 만들기 v2

- 멀티 쓰레드 활용

Additional : 파일 전송

- 텍스트 파일 전송

Part 1. 메신저 만들기

1. 다음 요구사항에 맞춰 메신저를 제작하시오

요구사항

- 소켓 프로그래밍(TCP)를 활용해야함
  - 전송을 받을 시 받은 메시지는 recv_log.txt, 보낸 메시지는 send_log.txt에 기록
  - 반드시 서버가 먼저 켜져야 동작
    (즉, 서버가 클라이언트에 연결하려고 시도해선 안됨)
  - 최대 1KiB까지의 메시지 전송이 가능해야함
  - 클라이언트부터 메시지를 보낼 수 있어야함
- Makefile을 이용한 컴파일을 해야함

Part 2. 메신저 만들기 v2

1. 다음 요구사항에 맞춰 메신저를 제작하시오

요구사항

- 소켓 프로그래밍(TCP)를 활용해야함
  - 전송을 받을 시 받은 메시지는 recv_log.txt, 보낸 메시지는 send_log.txt에 기록
  - 반드시 서버가 먼저 켜져야 동작
    (즉, 서버가 클라이언트에 연결하려고 시도해선 안됨)
  - 최대 1KiB까지의 메시지 전송이 가능해야함
  - 클라이언트부터 메시지를 보낼 수 있어야함
  - read와 send가 동시적으로 작동해야함
  - read만 쓰레드로 띄울 것
- Makefile을 이용한 컴파일을 해야함

Additional. 파일 전송

1. 다음 요구사항에 맞춰 파일 전송기를 제작하시오

요구사항

- 소켓 프로그래밍(TCP)를 활용해야함
  - 서버는 클라이언트의 파일 전송을 대기 해야함
  - 클라이언트는 argument로 보낼 텍스트 파일을 받음
  - 서버는 클라이언트가 보낸 파일을 “클라이언트가 전송하고자 하는 파일명”으로 저장
- Makefile을 이용한 컴파일을 해야함

Thank you