



# 윤성우의 열혈 TCP/IP 소켓 프로그래밍

윤성우 저 열혈강의 TCP/IP 소켓 프로그래밍 개정판

## Chapter 07. 소켓의 우아한 종료

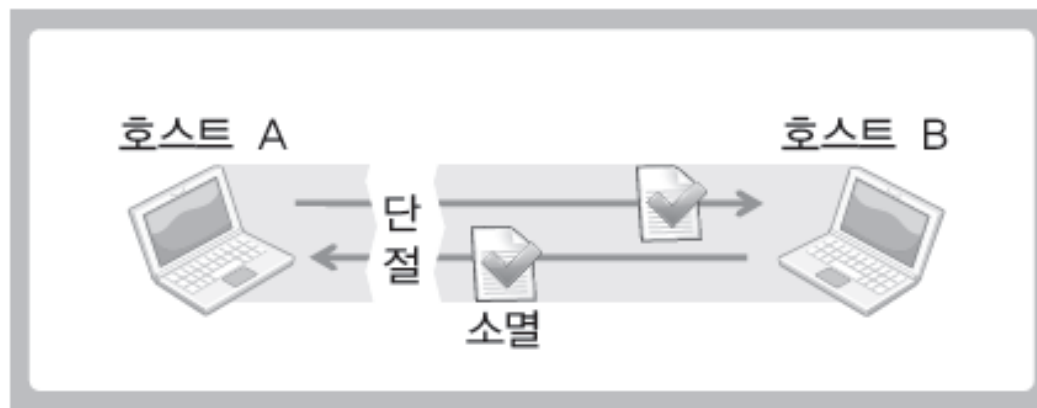


## Chapter 07-1. TCP 기반의 Half-close

윤성우 저 열혈강의 TCP/IP 소켓 프로그래밍 개정판

# 일방적인 연결 종료의 문제점

- ▶ close 및 closesocket 함수의 기능
  - ▶ 소켓의 완전 소멸을 의미한다.
  - ▶ 소켓이 소멸되므로 더 이상의 입출력은 불가능하다.
  - ▶ 상대방의 상태에 상관 없이 일방적인 종료의 형태를 띤다.
  - ▶ 때문에 상대 호스트의 데이터 송수신이 아직 완료되지 않은 상황이라면, 문제가 발생할 수 있다.
  - ▶ 이러한 문제의 대안으로 Half-close 기법이 존재한다.



# 소켓의 Half-close

## ▶ Half-close

- ▶ 종료를 원한다는 것은, 더 이상 전송할 데이터가 존재하지 않는 상황이다.
- ▶ 따라서 출력 스트림은 종료를 시켜도 된다.
- ▶ 다만 상대방도 종료를 원하는지 확인되지 않은 상황이므로, 출력 스트림은 종료시키지 않을 필요가 있다.
- ▶ 때문에 일반적으로 Half-close라 하면, 입력 스트림만 종료하는 것을 의미한다.
- ▶ Half-close를 가리켜 ‘우아한 종료’라고도 한다.



입력 또는 출력 스트림 중  
하나만 종료하는 것을 가리  
켜 Half-close라 한다.

# 우아한 종료를 위한 Shut-down 함수와 그 필요성

```
#include <sys/socket.h>
```

```
int shutdown(int sock, int howto);
```

➔ 성공 시 0, 실패 시 -1 반환

- sock      종료할 소켓의 파일 디스크립터 전달.
- howto    종료방법에 대한 정보 전달.

- SHUT\_RD      입력 스트림 종료
- SHUT\_WR      출력 스트림 종료
- SHUT\_RDWR    입출력 스트림 종료

close 함수가 호출되면 **상대 호스트(소켓)으로 EOF가 전달된다**. 그리고 이는 **모든 데이터의 전송이 끝났다는 신호의 의미**를 갖는다. 이것이 종료 이외의 close 함수를 호출하는 목적이다. 그런데 **출력 스트림만 종료를 해도 EOF가 전달이 되니**, close 함수의 호출을 대체하고도, 상대 호스트의 종료를 기다릴 수 있다.

# Half-close 기반 파일 전송 프로그램

## file\_server\_win.c 의 일부

```
while(1)
{
    read_cnt=fread((void*)buf, 1, BUF_SIZE, fp);
    if(read_cnt<BUF_SIZE)
    {
        write(clnt_sd, buf, read_cnt);
        break;
    }
    write(clnt_sd, buf, BUF_SIZE);
}

shutdown(clnt_sd, SHUT_WR);
read(clnt_sd, buf, BUF_SIZE);
printf("Message from client: %s \n", buf);

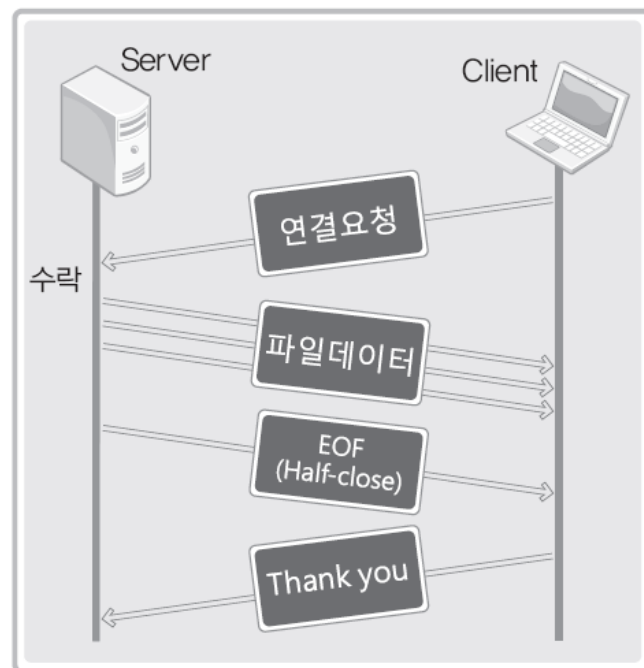
fclose(fp);
close(clnt_sd); close(serv_sd);
```

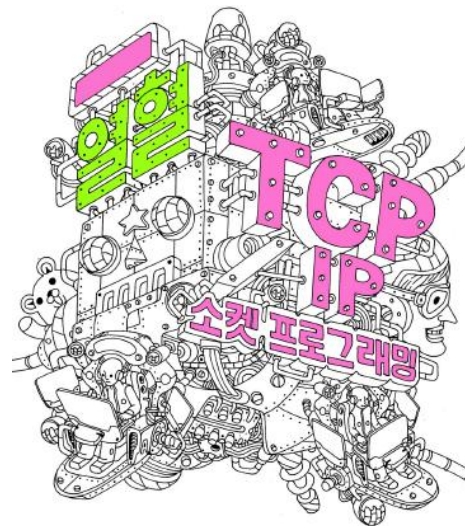
```
while((read_cnt=read(sd, buf, BUF_SIZE ))!=0)
    fwrite((void*)buf, 1, read_cnt, fp);

puts("Received file data");
write(sd, "Thank you", 10);
fclose(fp);
close(sd);
```

## file\_client\_win.c 의 일부

Half-close가 필요한 상황의 연출





## Chapter 07-2. 윈도우 기반으로 구현하기

윤성우 저 열혈강의 TCP/IP 소켓 프로그래밍 개정판

# 윈도우 기반 shutdown 함수

```
#include <winsock2.h>
```

```
int shutdown(SOCKET sock, int howto);
```

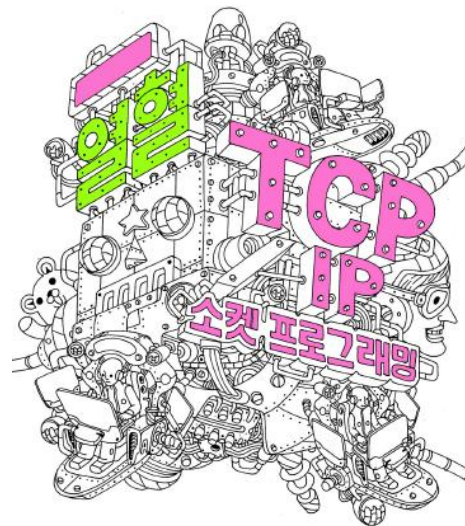
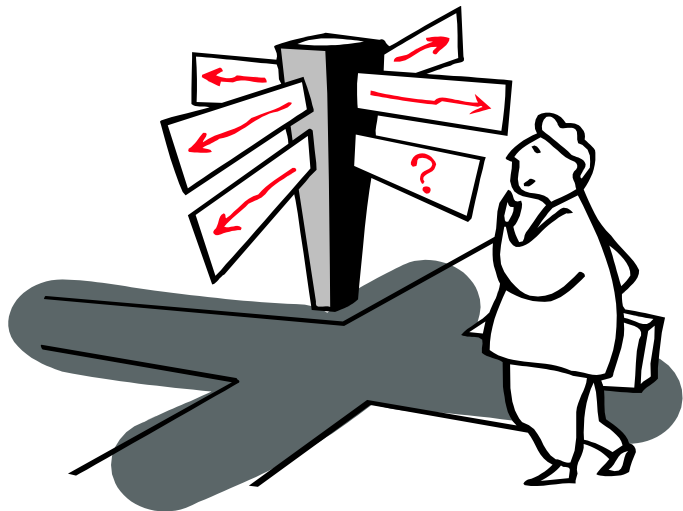
➔ 성공 시 0, 실패 시 SOCKET\_ERROR 반환

- sock      종료할 소켓의 핸들.
- howto    종료방법에 대한 정보 전달.

- SD\_RECEIVE    입력 스트림 종료
- SD\_SEND      출력 스트림 종료
- SD\_BOTH      입출력 스트림 종료

전달인자의 이름에서만 차이가 있을 뿐 그 이외의 모든 것은 Linux의 shutdown 함수와 동일하다.





Chapter 07이 끝났습니다. 질문 있으신지요?