

1. 데이터 송수신 동작의 개요

OS 내부의 프로토콜 스택에 메시지 송신 동작을 의뢰할 때는 Socket라이브러리 프로그램 부품을 결정된 순번대로 호출

송수신 과정에 파이프와 같은 것을 통해 데이터가 흐르며 송수신을 위해서는 파이프를 연결하는 동작이 필요

소켓 : 파이프 양끝에 있는 데이터의 출입구

데이터 송수신 동작 과정

- ➔ 소켓 만들기 (소켓 작성 단계)
- ➔ 서버 측의 소켓에 파이프 연결 (접속 단계)
- ➔ 데이터를 송수신 (송 수신 단계)
- ➔ 파이프를 분리하고 소켓을 말소 (연결 끊기 단계)

위의 4가지 동작을 실행하는 것이 OS 내부의 프로토콜 스택

2. 소켓의 작성 단계

클라이언트 측의 소켓 작성 방법

- ➔ 소켓 라이브러리의 socket프로그램 부품만 호출
- ➔ 호출한 후 socket에 제어가 넘어가 소켓을 만들
- ➔ 이후 애플리케이션에 제어가 돌아옴

참고 : Socket : 라이브러리

socket : 부품

소켓 : 파이프의 양끝에 있는 출입구

- ➔ 소켓이 생기면 '디스크립터'가 반환 (디스크립터는 소켓이 여러 개일 경우 식별하기 위해 사용)
- ➔ 애플리케이션은 '디스크립터'라는 번호표로 소켓 식별

3. 파이프를 연결하는 접속 단계

- ➔ 애플리케이션은 Socket라이브러리의 connect라는 프로그램 부품을 호출하여 서버 측의 소켓에 접속하도록 프로토콜 스택에 의뢰
- ➔ 이때 디스크립터, 서버의 IP주소, 포트번호 세가지 값이 필요
 - 포트 번호는 접속 상대 측에서 소켓을 지정하기 위해 사용(웹 서버 80, 메일 25 등과 같이 서버에 접속할 때 포트번호는 고정되어 있음)
 - 서버에서도 클라이언트의 소켓 번호를 알 수 있도록 프로토콜 스택이 접속동작을 실행할 때 서버 측에 자신의 소켓 번호를 알려줌

4. 메시지를 주고받는 송수신 단계

- ➔ 소켓이 상대 측과 연결 되면 소켓으로 데이터를 쏘아 부으면 상대 측의 소켓에 데이터가 도착
 - 송신 데이터 : HTTP 리퀘스트 메시지
- ➔ 이때 write라는 프로그램 부품을 사용하여 프로토콜 스택에 송신 동작 의뢰
- ➔ 수신할 때에는 read라는 프로그램 부품을 통해 프로토콜 스택에 수신 동작 의뢰
 - 수신한 응답 메시지를 저장하기 위해 이것을 "수신 버퍼"라는 메모리 영역을 지정
- ➔ "수신 버퍼"는 애플리케이션 프로그램의 내부에 마련된 메모리 영역이므로 메시지를 저장한 시점에서 애플리케이션에 건네 줌

5. 연결 끊기 단계에서 송수신 종료

데이터 송수신이 완료되면 Socket라이브러리의 close라는 프로그램 부품을 호출하여 연결 끊기 단계로 들어 감

- ➔ 서버에서 응답 메시지의 송신을 완료하면 서버 측에서 먼저 close를 호출하여 연결 끊음
- ➔ 이것이 클라이언트 측에 전달되면 클라이언트 측에서도 연결 끊기 단계로 들어 감