

1. echo_client.c:

- 프로토콜: TCP(SOCK_STREAM)
- `socket()`, `connect()`, `write()`, `read()`를 사용하여 통신합니다.
- `connect()` 함수가 호출되어 서버와 연결을 설정합니다.
- 통신은 스트림 기반이므로 지속적인 스트림으로 메시지를 보내고 받습니다.

2. uecho_client.c:

- 프로토콜: UDP(SOCK_DGRAM)
- `socket()`, `sendto()`, `recvfrom()`을 사용하여 통신합니다.
- UDP는 연결이 없으므로 서버와 연결을 설정하지 않습니다.
- 메시지는 개별 패킷으로 전송 및 수신됩니다.

3. uecho_con_client.c:

- 프로토콜: UDP(SOCK_DGRAM)
- TCP 클라이언트와 유사하게 `socket()`, `connect()`, `write()`, `read()`를 사용하여 통신합니다.
- 일반적인 UDP 클라이언트와 달리 이 클라이언트는 `connect()`를 호출합니다. 이는 TCP와 같은 실제 연결을 설정하지는 않지만 `send()` 및 `recv()`에 대한 기본 서버 주소를 설정하고 서버는 이 클라이언트로부터만 메시지를 수신합니다.
- UDP를 사용하지만 `connect()` 함수는 `sendto()` 및 `recvfrom()` 대신 `write()` 및 `read()`를 허용하여 코드를 단순화합니다.
- 기본적으로 이는 연결 지향 동작을 시뮬레이션하는 UDP 클라이언트입니다.

주요 차이점:

- `echo_client.c`와 다른 두 가지(`uecho_client.c` 및 `uecho_con_client.c`)의 주요 차이점은 사용되는 전송 프로토콜입니다. `echo_client.c`는 지속적인 연결이 필요한 TCP 클라이언트인 반면, 나머지 두 개는 지속적인 연결이 필요하지 않은 UDP 클라이언트입니다.

