

Zadanie 1.

TCP The Transmission Control Protocol

Diagram jest na stronie 138. Protokół używa **gniazd strumieniowych** (wymagają połączenia dane są otrzymywane w kolejności wysłania).

Serwer po wywołaniu `socket()`, wywołuje `bind()` (aby powiązać z socketem adres lokalny (dla TCP IP i nr portu), następnie czeka na połączenie (`listen()`) oraz je akceptuje.

Po wywołaniu `socket()` klient woła `connect()` (tam się podaje IP i nr portu serwera), nie musi wołać `bind()` – jądro przydzieli mu **port ulotny** (port, który został przydzielony tylko do danego połączenia, jest zwalniany, gdy ono wygaśnie).

Potem komunikacja następuje przez pisanie do `fd` socketa, jest zakończone po otrzymaniu EOF od klienta.

UDP User Datagram Protocol

Diagram jest na stronie 299. Protokół używa **gniazd datagramowych** (nie wymagają połączenia, dane są dzielone na pakiety, które mogą dotrzeć w dowolnej kolejności lub nie dotrzeć).

Serwer woła `bind()` na swoim sockecie, aby słuchać na odpowiednim adresie (IP oraz port).

Klient nie musi inicjować połączenia z serwerem. Po prostu wysyła pakiety na wskazany adres. Nie woła on `bind()`, więc to on używa **portu ulotnego**

Strony komunikują się za pomocą `recvfrom()` i `sendto()`, które odpowiednio zwracają i przyjmują adres drugiej strony komunikacji.

`Read` i `write` nie miałyby sensu, bo nie ma ustalonego odbiorcy ani nadawcy pod danym socketem. Serwer może odbierać i wysyłać pakiety od wielu klientów przez ten sam socket, podobnie klient.

Nie ma `close()`, bo to jest protokół bezpołączeniowy.