# 17. Tétel

Hogyan valósítunk meg Linux alkalmazásunkban összeköttetés nélküli kommunikációt

## Linux hálózatkezelés

* socketek
  + mint a Linux többi erőforrásai 🡪 egy-egy hálózati kapcsolat is állományként van reprezentálva
  + kapcsolat létrehozása 🡪 létrejön egy socket állomány 🡪read() write() 🡪 nem ajánlott
  + ajtó az alkalmazás és a hálózat között
    - IP cím
    - portszám
* hálózati kapcsolat felépítése
  + aszimmetrikus művelet

|  |  |
| --- | --- |
| Kliens | Szerver |
| Socket létrehozása | Socket létrehozása |
|  | Kapcsolódás portjának megadása (bind(9) |
|  | Hallgatózás (listen()) |
| csatlakozási kérelem küldése: (connect()) |  |
|  | csatlakozás elfogadása (accept()) |
|  | csatlakozás létrejött, létrehoz egy kliens-socket-et 🡪 a kommunikáció ezek után ezen keresztül történik |

## Összeköttetés alapú kommunikáció (connection-oriented)

* folyamatos kapcsolatban van a két oldal között
* send, receive

## Összeköttetés nélküli kommunikáció (connectionless)

* a kommunikációra „megbízhatóként tekint” 🡪 nem foglalkozik azzal, hogy valóban célba ért-e a csomag (ha hiba van, vagy megszakadt a csomag, akkor úgyis dob hibát a kommunikációs interface)
* olyan alkalmazásoknál, ahol nem probléma, ha egy-egy csomag elveszik
* UDP csomagok TCP helyett
  + UDP
    - User Datagram Protocol
    - korlátozott méretű adatcsomagok átvitelére képes, nem megbízható kommunikációs útvonalat biztosító protokoll
    - nem ganadtál célba jutást, csak hibakezelst nyújt
    - gyors
    - Hálózat menedzsment, Voip, média streaming, névfeloldás
  + TCP
    - Transmission Control Protocol
    - megbízható átvitel
    - sorrendhelyes, hibamentes szállítást nyújt
    - ára: késleltetés (kapcsolat felépítése, bontása)
    - e-mail (smtp), ssh, web (http), ftp
* küldő oldalon: socket létrehozása 🡪 adat küldése
* vevő oldalon: listening 🡪 csomagfogadás 🡪 socket porthoz rendelése
* blokkolás elkerülése: többszállúság, multiplexing