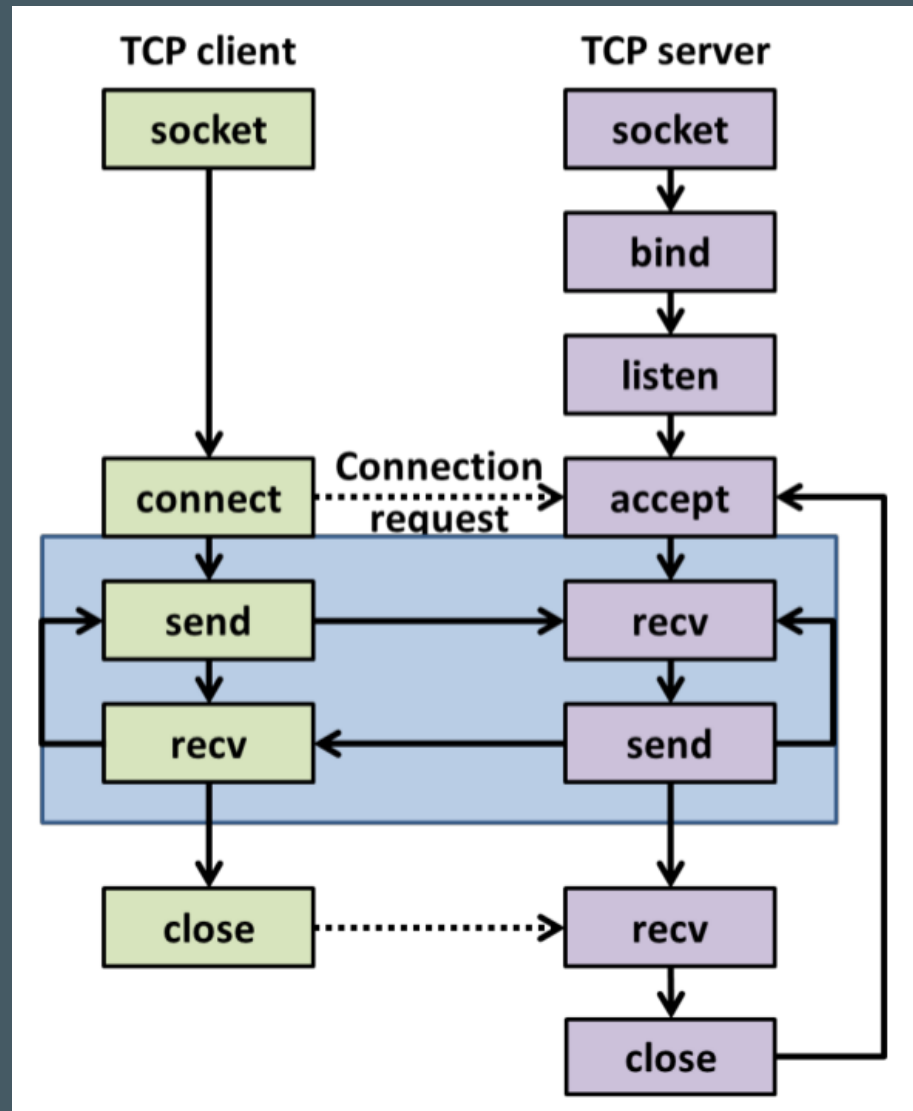


# Telekommunikációs Hálózatok

3. gyakorlat

# TCP



# TCP

- `socket()`

```
import socket  
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

- `bind()`

```
server_address = ('localhost', 10000)  
server.bind(server_address)
```

- `listen()`

```
server.listen(1)
```

- `accept()`

```
connection, client_address = sock.accept()
```

# TCP

- `send()`, `sendall()`

```
connection.sendall(data.encode())
```

- `recv()`

```
data = connection.recv(16).decode()
```

- `close()`

```
connection.close()
```

- `connect()`

```
server_address = ('localhost', 10000)  
client.connect(server_address)
```

# Gyakorlás I.

- Készítsünk egy egyszerű kliens-server alkalmazást, ahol a kliens elküld egy 'Hello server' üzenetet, és a szerver pedig válaszol neki egy 'Hello client' üzenettel!
- Változtassuk meg, hogy ne beégetett portot használjunk, hanem adjunk át parancssori argumentumként

# Gyakorlás II.

- Készítsünk egy szerver-kliens alkalmazást, ahol a kliens elküld 2 számot és egy operátort a szervernek, amely kiszámolja és visszaküldi az eredményt. A kliens üzenete legyen struktúra.

# Listen - sok kliens

Készítsünk egy olyan alkalmazást, ahol a szerver várakozási sora 1 hosszú:

```
server.listen(0)
```

A klienshez tartozó script-ben 3 db. kliens socket-et hozunk létre és mindegyikkel a szerverhez próbáljunk csatlakozni egymás után. Figyeljük meg mi történik!

Windows-nál a végtelen ciklusban futó szervert sima **Ctrl+C**-vel nem tudjuk kilőni parancssorban, hanem **Ctrl+Break** billentyűkombinációval lehet. A **Break** billentyű helye laptoponként eltérhet: pl. **Ctrl+Fn+Pause**, **Ctrl+Fn+B** stb

# Gyakorlás III.

Készítsünk egy szerver-kliens alkalmazást, amely a tic-tac-toe játékot valósítja meg

Szerveren 3x3 mátrix van:

- Ellenőrzi hogy vége-e a játéknak
- Üzenetek: Nyertel, Vesztettel, Lepes, Varj,

Kliens:

- Kapcsolódik, majd fogadja a szerver üzenetét, ha Ő a kezdő játékos, akkor:
  - Bekéri a stdinutról x,y koordinátát és elküldi a szervernek
  - Várakozik a szerver válaszára, amiben majd a másik játékos koordinátája van
- Ha nem kezdő:
  - Akkor várakozik a másik játékos koordinátájára



# II. Beadandó

## *Protokoll*

- Leírás

## II. Beadandó

- A paraméterben kapott bináris fájlokat kell beolvasni és kiíratni az első soruk tartalmát a standard outputra! (A különböző fájlok sorainak formátuma hallgatónként változó)
- Ki kell írni a standard outputra különböző értékeket bináris formátumban (azaz a `pack` eredményét)! (Az értékek hallgatónként változók.) A string hosszát a szöveg mögött lévő szám jelzi!  
Használandó `struct` paraméterek: `f`, `i`, `c`, `?`, `Xs` (ahol a X a string hossza, pl: `3s`)

**Leadás:** A program leadása a TMS rendszeren .zip formátumban, amiben egy `client.py` szerepeljen!

**Határidő:** TMS rendszerben