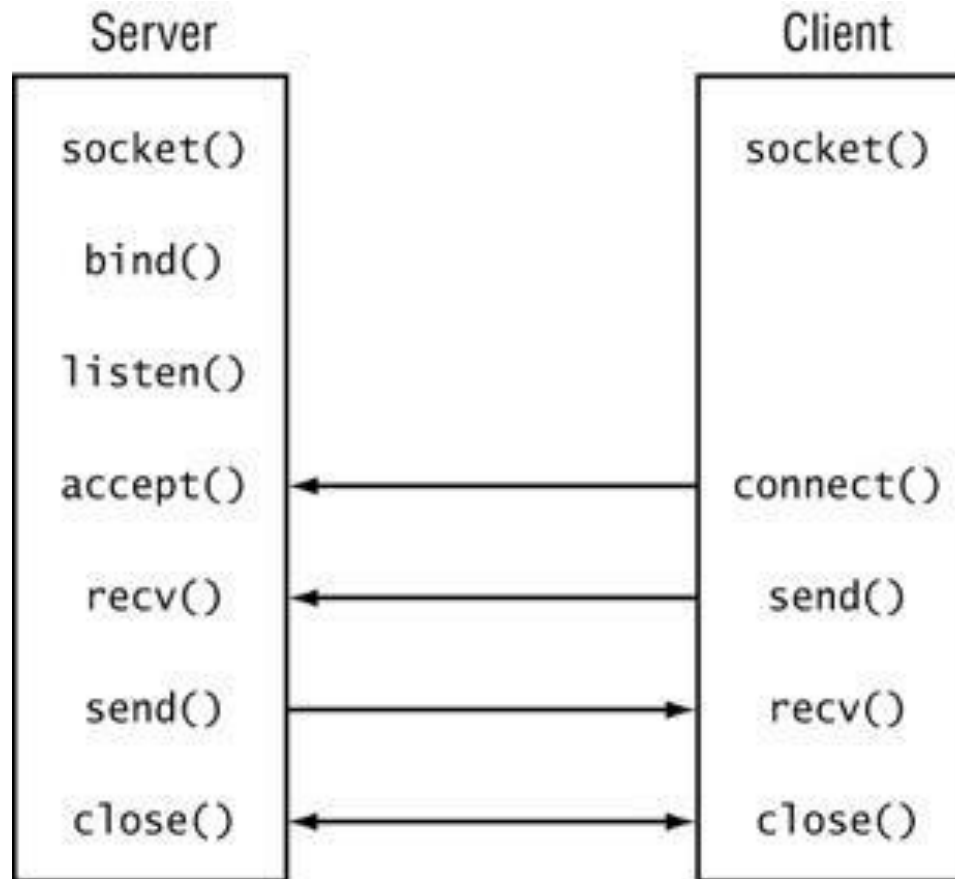


# Telekommunikációs Hálózatok

## 3. gyakorlat

# TCP



# TCP

- `socket()`

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

- `bind()`

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

```
server_address = ("", 10000)  
sock.bind(server_address)
```

- `listen()`

```
sock.listen(1)
```

- `accept()`

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

# TCP

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

```
connection.sendall(data) #python 2.x
```

```
connection.sendall(data.encode()) #python 3.x
```

- `recv()`

```
data = connection.recv(16) #python 2.x
```

```
data = connection.recv(16).decode() #python 3.x
```

- `close()`

```
connection.close()
```

- `connect()`

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

# Feladat

- Készítsünk egy egyszerű kliens-server alkalmazást, ahol a kliens elküld egy ,Hello server' üzenetet, és a szerver pedig válaszol neki egy ,Hello kliens' üzenettel!
- Változtassuk meg hogy ne az előre megadott portot adjuk, hanem egy tetszőlegesen kapjunk az oprendszer-től! (sys.argv[1])

# Listen – sok kliens

- Készítsünk egy olyan alkalmazást, ahol a szerver oldalon:

```
sock.listen(1)
```

- A klienshez tartozó szkriptben 3 db. kliens socket-et hozunk létre és mindegyikkel a szerverhez próbálunk csatlakozni egymásután
- **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.)

# Feladat

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.

# Hibakezelés

```
C:\Users\Gogo\Documents\git\comnet_course\202223-1\gyak3\codes>python socket.py
Traceback (most recent call last):
  File "C:\Users\Gogo\Documents\git\comnet_course\202223-1\gyak3\codes\socket.py", line 1, in <module>
    from socket import socket, AF_INET, SOCK_STREAM, SOCK_DGRAM, timeout, error
  File "C:\Users\Gogo\Documents\git\comnet_course\202223-1\gyak3\codes\socket.py", line 1, in <module>
    from socket import socket, AF_INET, SOCK_STREAM, SOCK_DGRAM, timeout, error
ImportError: cannot import name 'socket' from partially initialized module 'socket' (most likely due to a circular import) (C:\Users\Gogo\Documents\git\comnet_course\202223-1\gyak3\codes\socket.py)
```

- A file neve ne socket legyen, mert a python előbb a saját mappájából importál!

```
Traceback (most recent call last):
  File "C:\Users\Gogo\Documents\git\comnet_course\202223-1\gyak3\codes\server.py", line 8, in <module>
    server.bind(server_addr)
OSError: [WinError 10048] Az összes szoftvercsatorna-cím használatának általában csak egy módja (protokoll/hálózat cím/port) engedélyezett
```

- A port már használatban van
  - másik port
  - `server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)`



# Hibakezelés

```
C:\Users\Gogo\Documents\git\comnet_course\202223-1\gyak3\codes>python client.py
Traceback (most recent call last):
  File "C:\Users\Gogo\Documents\git\comnet_course\202223-1\gyak3\codes\client.py", line 7, in <module>
    server.connect(server_addr)
ConnectionRefusedError: [WinError 10061] Nem hozható létre kapcsolat, mert a célszámítógép már visszautasította
a kapcsolatot
```

- Nem fut a szerver!
- Hiba üzenet van a szerveren!

```
Traceback (most recent call last):
  File "C:\Users\Gogo\Documents\git\comnet_course\202223-1\gyak3\codes\server.py", line 15, in <module>
    packer.unpack(data)
struct.error: unpack requires a buffer of 8 bytes
```

- Nem megfelelő üzenet érkezett
- Hiba van a szerveren, emiatt 0 byte-os csomag érkezett!

# Feladat

## 1. Csináljunk egy date servert!

- A kliens elküldi hogy 'getDate' és a szerver visszaküldi az aktuális dátumot.
  - pl: `datetime.now().isoformat()`

## 2. Csináljunk egy Nagybetűző servert!

- A kliens elküld egy szöveget a szerver visszaküldi nagybetűzve
  - pl: `"as".upper() → "AS"`

## 3. Csináljunk egy dobóocka servert!

- A kliens csatlakozik és elküldi hogy hány oldalú dobókockára kér egy dobást.
  - pl: `"20" → random.randint(1,20)`

# Házi feladat

<https://ggombos.web.elte.hu/halobeadando/2-protokoll/>

**KÖSZÖNÖM A FIGYELMET!**