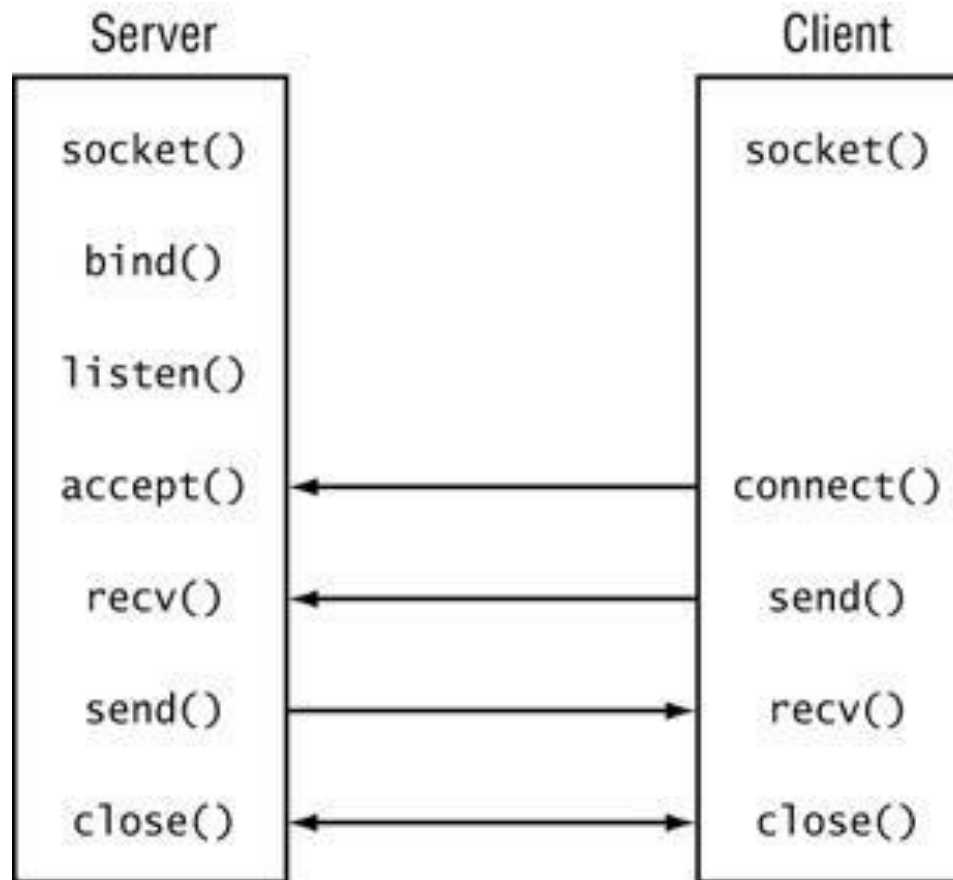


Telekommunikációs Hálózatok

3. gyakorlat

TCP



TCP

- `socket()`

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

- `bind()`

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

- `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 oprendszertől! (sys.argv[1])

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.

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, amely a tic-tac-toe játékot valósítja meg.
 - Szerveren 3x3 matrix van
 - Ellenorzi hogy vege-e a jateknak
 - Üzenetek: Nyertel, Vesztettel, Lapes, Varj,
 - Kliens:
 - Kapcsolódik, majd fogadja a szerver üzenetet, ha Ő a kezdő játékos, akkor:
 - Bekéri a stdinputról x,y koordinatát és elküldi a szervernek
 - Várakozik a szerver válaszára, amiben a majd a másik játékos koordinataja van
 - Ha nem kezdő:
 - Akkor várakozik a másik játékos koordinátájára

Házi Feladat

Két részfeladatból fog állni:

1. A paraméterben kapott bináris fájlokat kell beolvasni és kiíratni az első soruk tartalmát a standard outputra! (A különböző fájlok sorainak formátuma hallgatónként változó lesz.)
2. Ki kell írni a standard outputra különböző értékeket bináris formátumban (azaz a pack eredményét)! (Az értékek hallgatónként változók lesznek.) 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

VÉGE
KÖSZÖNÖM A FIGYELMET!