

Barkóba

Készítsünk egy barkóba alkalmazást. A szerver legyen képes kiszolgálni több klienst. A szerver válasszon egy egész számot 1..100 között véletlenszerűen. A kliensek próbálják kitalálni a számot.

- A kliens logaritmikus keresés segítségével találja ki a gondolt számot. AZAZ a kliens NE a standard inputról dolgozzon.
- Ha egy kliens kitalálta a számot, akkor a szerver minden újabb kliens üzenetre a "Vége" (V) üzenetet küldi, amire a kliensek kilépnek.
- Nyertél (Y), Kiestél (K) és Vége (V) üzenet fogadása esetén a kliens bontja a kapcsolatot és terminál. Igen (I) / Nem (N) esetén folytatja a kérdezgetést.
- A kommunikációhoz TCP-t használjunk!
- A server a SELECT fv-t használja több kliens kiszolgálásához!!!!
- Ha a játék végetért, akkor minden kérésre a szerver válaszoljon, hogy "Vége" (V).

Üzenet formátum:

- Klientől: **bináris formában** egy db karakter, 32 bites egész szám. (struct) Ne használjuk a byte sorrend módosító operátort a struct-ban! ('!')
 - A karakter lehet: <: kisebb-e, >: nagyobb-e, =: egyenlő-e
 - pl: ('>',10) //a szám nagyobb mint 10
- Szervertől: ugyanaz a **bináris formátum** , de a számnak nincs szerepe, bármi lehet (struct)
 - A karakter lehet: **I**: Igen, **N**: Nem, **K**: Kiestél, **Y**: Nyertél, **V**: Vége
 - pl: ('V',0)

Script paraméterezése:

- python3 client.py < hostname > < port szám >
 - pl: python3 client.py localhost 10000
- python3 server.py < hostname > < port szám >
 - pl: python3 server.py localhost 10000

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

#####

Question / Answer

Let's make a question / answer application. The server should be able to serve several clients. The server should choose an integer between 1..100 at random. Clients try to guess the number.

- The client uses a logarithmic search to find the intended number. The client should NOT work from the standard input.
- If a client has guessed the number, the server sends the message "End" (V) to every new client message, after which the clients exit.
- Upon receiving Win (Y), Lose (K) and End (V) messages, the client disconnects and terminates the connection. In case of Yes (I) / No (N), it continue the questioning.
- Use TCP for communication!
- The server uses SELECT funtion to serve multiple clients!!!!
- If the game is over, the server send End (V) for every question!

Message format:

- Form Client: **bytes format** one char, one integer. (struct) Do not use the byte order manipulation operator! ('!')
 - The char is: <: lower then, >: bigger then, =: equals
 - ex: ('>',10) //the number is bigger then 10
- From Server: same **bytes format** , but the number is irrelevant (struct)
 - The char is: **I**: Yes, **N**: No, **K**: Quit, **Y**: Win, **V**: End
 - ex: ('V',0)

Parameters of the script:

- python3 client.py < hostname > < port number >
 - ex: python3 client.py localhost 10000
- python3 server.py < hostname > < port number >
 - ex: python3 server.py localhost 10000

Submission: Send it to the TMS in **.zip** file which should contain a **client.py** and a **server.py** only!