

Gyakorló ZH

Készíts egy lottóalkalmazást, ahol a szerver generál 5 számot 1 és 20 között (hogyan legyen esély nyerni). Ezek lesznek a lottószámok. A kliensek beküldik a tippjeiket, illetve, hogy mennyi pénzt raknak fel a szelvényre. A szerver megmondja, hogy mik voltak a nyerőszámok, és, hogy mennyit nyert az adott játékos: 1 találat 1x, 2 találat 2x, 3 találat 3x, stb.

Feladat I.

- a) Oldd meg a feladatot TCP felett!
- b) Oldd meg a feladatot UDP felett!

Feladat II.

- a) Az üzenet formátuma bytes legyen ':' szeparátorral!
- b) Az üzenet formátuma struct legyen, és legalább 2 mező legyen benne!
- c) Az üzenet mellé rakjunk egy ellenőrzőkódot is, amit MD5-tel vagy CRC-vel generálunk.

Feladat III.

A szerverre egyszerre több kliens is tudjon csatlakozni! (TCP-nél selecttel)

Feladat IV.

- a) A kliens a tippet random adja meg!
- b) A kliens a tippet standard inputról olvassa be!
- c) A kliens a tippet json file-ból olvassa be!
- d) A kliens a tippet parancssori argumentumként kapja!

Feladat V.

Készíts egy history szerveret, amely eltárolja az eddig kihúzott számokat, a játékosok tippjeit és a nyereményeit. A kapott adatot írja egy json file-ba (*append*). A protokoll ne ugyanaz legyen, mint eddig. Ha a kliens TCP-s, akkor a history szerver UDP-s, és fordítva.

- a) A history szerver egy proxyként működjön. A kliens neki küldi a tippjét, és továbbítja a lottó szervernek, aki a választ visszaküldi a history szervernek, aki ekkor letárolja, majd elküldi a kliensnek az eredményt.
- b) A history szerver külön szerverként fusson. A lottó szerver hívja meg a kliens kérése után. Elküld minden adatot, amit le kell tárolni a file-ba.

- c) A kliens kommunikáljon a history szerverrel, miután megkapta a választ a lottó szervertől.