

Számítógépes hálózatok - Géptermi ZH - NAPPALI

May 20, 2011

1 Készlet lekérdezés (15 pont)

Készítse el egy bolt "készlet lekérdező rendszerét" a következő módon! Adott egy szerver, amihez egyszerre több kliens is csatlakozhat. A kliensek lekérdezhetik a szervertől, hogy adott cikkszámú terméknek mi a megnevezése és mekkora a raktárkészlet.

1.1 A szerver

A szerver felelős a cikkszámhoz tartozó kérdések megválaszolásáért. Működése a következő:

- (*) A kliensek csatlakozására várakozik valamely előre rögzített TCP porton. Selecttel valósítsd meg, hogy egyszerre legalább két kliens tudjon csatlakozni!
- Ha már csatlakozott legalább 1 kliens, akkor kezeli a tőle jövő kérdéseket.
- (**) Minden kliens egy egész számot, egy cikkszámot küld a szervernek. Ha megkapta a szerver, akkor visszaküld egy 0 és 10 közötti véletlen számot (ez a raktárkészlet) és a termék megnevezését (max. 1024 hosszú sztring). A visszaküldött üzenet formátuma tetszőleges lehet, de többlet pont jár, ha a válasz üzenethez egy struktúrát használ, és azt bináris formátumban küldöd vissza. (A terméknév bármi lehet. Értelmetlen karakterek is, de legyen benne sztringet lezáró karakter. Legegyszerűbb, ha az `strcpy(buffer,"valami")`; függvényt használod. Minden cikkszámra válaszolhat ugyanazzal az üzenettel is!)
- Eztán a kliens újabb kérdéseket küldhet a fenti formában. A program kezelje, ha a kliens bontotta a kapcsolatot!
- A szerver mindenképp írja ki a képernyőre a felhasználóktól kapott és a nekik küldött üzeneteket!!!

1.2 A kliens

A kliens működése legyen a következő:

- Csatlakozik a szerver előre rögzített TCP portjához.
- Majd elküld egy cikkszámot (pl. 12).
- Erre válaszol a szerver: egy egész számmal (raktárkészlet) és egy 1024 byte hosszú üzenettel (terméknév). Írjad ki a kapott üzenetet!
- Eztán a kliens vár 2 mp-ig, majd feltesz még egy kérdést és várja a választ a szervertől.
- Ezt követően bontja a kapcsolatot a szerverrel.

2 Biztonságos küldés, fogadás (5 pont)

Készítse el a következő két függvényt:

- **int sendSafe(int sock, void *buffer, int bufferSize)** : a függvény a bufferből bufferSize-nyi bájtot garantáltan elküld a megadott socketen keresztül. A visszatérési érték, az elküldött bájtok száma (ez mindig bufferSize), vagy hibakód (kisebb mint 0). (A függvény okosan addig hívja a send metódust, amíg az összes bájt el nem lett küldve...)
- **int recvSafe(int sock, void *buffer, int bufferSize)** : a függvény a buffer-be garantáltan bufferSize-nyi bájtot olvas a megadott socketről. A visszatérő érték hasonló a fentihez.

3 Segítség

- Véletlen szám generálása: Szükséges fejláományok: `time.h`, `stdlib.h`. A program elején inicializáljuk a random generátort az `srand(time(0))`; utasítással. Majd `rand()` paranccsal kapunk egy 0 és `RAND_MAX` közötti egész értékű számot.
- Sztring összehasonlítása: fejláomány: `string.h`; Példa: `if (strcmp(buffer, "apple") == 0) //akkor alma van a bufferban`
- Sztring "értékadás" C-ben: `strcpy(a, "valami");` függvény segítségével!
- C-s sztringek esetén, a karakter tömbben szereplő `\0` karakter jelzi a sztring végét! Erre érdemes odafigyelni!
- Ha adott egy `void * buffer`, ami egy 1000 elemű bájtömb elejére mutat, akkor a 100. elemre (100. elemnél kezdődő résztömbre) így hivatkozhatunk: `buffer+100`;