

MREŽE RAČUNALA – 1. KOLOKVIJ

Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i službeni šalabahter. Predajete samo papire koje ste dobili.

Uvid i rezultati:

ZADATAK 1

1

(7 bodova)

Dizajnirajte protokol koji će omogućiti naručivanje hrane. Klijent treba moći:

1. dobiti popis svih trenutno otvorenih restorana
2. filtrirati popis restorana po:
 - minimalnoj cijeni za naručivanje
 - vrsti hrane (postoji nekoliko predefiniраниh opcija za izabrati npr. meksička, vegetarijanska...)
 - prosječnom vremenu dostave
3. dobiti meni jednog od restorana
4. naručiti više jela s menija jednog od restorana (čime se komunikacija prekida)
5. odustati od naručivanja (čime se komunikacija prekida)

Osmislite format i vrste poruka koje razmjenjuju server i klijent. Obratite pažnju na moguće pogrešne unose od strane klijenta. Napišite primjer komunikacije između klijenta i servera.

Kratko opišite organizaciju podataka (implementaciju popisa restorana, menija) na serveru.

MREŽE RAČUNALA – 1. KOLOKVIJ – ZADATAK 2

(9 bodova)

Napišite dio serverske aplikacije koja može posluživati više klijenata istovremeno. Relevantni dio sastoji se od dvije funkcije sa sljedećim prototipovima:

```
void slusaj(int listenerSocket, int brojPrimanja)
```

```
void *obradi( void *p )
```

Sama obrada vrši se u funkciji `obradi` na sljedeći način:

1. server od klijenta prima `brojPrimanja` bajtova (tipa `char`)
2. server mora sve primljene bajtove zbrojiti i sumu dodati u globalnu varijablu `globalna_suma`

Funkcija `slusaj` prihvata veze i pokreće/zaustavlja dretve. Možete pretpostaviti da će biti najviše 50 dretvi.

Napomena: Obavezno minimizirajte vrijeme čekanja dretvi (neka čekaju samo kad moraju). Možete pretpostaviti da će se svi podaci primiti u jednom pozivu `recv` i da će uvijek uspjeti (nije potrebno obrađivati greške).

MREŽE RAČUNALA – 1. KOLOKVIJ – ZADATAK 3

(9 bodova)

Neka je zadana struktura:

```
struct student{
    char* ime;
    int broj_kolegija;
    char** nazivi_kolegija;
    int* ocjene;
};
```

Napišite funkciju `int posalji (int sock, struct student S)` koja šalje strukturu `S` na socket `sock`. U slučaju greške funkcija vraća `0`, a inače `1`. Podatak `broj_kolegija` predstavlja duljinu nizova `nazivi_kolegija` i `ocjene`, a svi podaci tipa `char*` su uobičajeni C-ovski stringovi.

Napišite i analognu funkciju `primi` koja prima tako poslanu strukturu (i alokira memoriju za nju).