

JMBAG

IME & PREZIME

Mreže računala

Ispit, 7. veljače 2024. godine

Na ispitu je dozvoljeno korištenje samo pribora za pisanje i službenog šalabahtera. Predajete samo papire koje ste dobili. Ispit se sastoji od šest zadataka na kojima je moguće ostvariti najviše 80 bodova. Ispit se piše 2 sata.

PRVI ZADATAK

OSTVARENI BODOVI

Zadana je sljedeća struktura podataka koja sadrži cijeli broj n te niz cijelih brojeva `numbers`. Broj n je duljina niza `numbers`.

```
struct Integers{
    int n;
    char* numbers;
};
```

Koristeći pretpostavku da se svaki broj i i svaki string može poslati jednom naredbom `send` i primiti jednom naredbom `recv`, riješite sljedeće zadatke.

a) Implementirajte funkcije:

- `int sendEven(int sock, struct Integers I)`
- `int receiveEven(int sock, struct Integers* J)`

Funkcija `sendEven` uzima jedan element `I` tipa `Integers`. Funkcija šalje broj parnih brojeva u nizu `numbers` te sve parne brojeve iz niza `numbers`.

Funkcija `receiveEven` mora te podatke primiti i pobrinuti se da oni završe na adresi `J`.

Pazite na greške koje se mogu dogoditi. Svaka funkcija mora vratiti 0 ako je sve dobro prošlo, inače 1.

b) Napišite dio serverskog koda od trenutka prihvaćanja konekcije do trenutka prekidanja konekcije (uključivo) koji od klijenta prima prirodan broj $n > 0$ te string `s` duljine n . Ako je `s` validan `hostname` nekog računala, koristeći funkcije iz a) dijela klijentu šalje sve parne znamenke IP adrese tog računala u dekadskom obliku. U suprotnom šalje -1. Nakon toga server prekida konekciju.

Upute:

- Pobrinite se da polja strukture `Integers` koje koristite budu adekvatne veličine (alocirajte potrebne podatke). Strukture možete slati element po element, ali ne cijele odjednom.
- Ako je `c` tipa `char` te predstavlja dekadsku znamenku, tada se konverzija u integer vrši na sljedeći način: `int n = c - '0'`.

JMBAG

IME & PREZIME

DRUGI ZADATAK

OSTVARENI BODOVI

 20

Platforma *Ful Kul Koncerti* omogućuje korisnicima kupovinu ulaznica za koncerte. Za svaki se koncert pamti naziv izvođača, datum, mjesto, vrijeme početka, cijena ulaznica te dostupan broj ulaznica. Primjerice: *Baby Lasagna*, 10.5.2025., *Boćarski dom Zagreb*, 20:30, 28€, 1500. Dan, mjesec, godina, sat, minute, cijena i dostupan broj ulaznica su tipa `int`. Ime izvođača i mjesto koncerta su stringovi (dinamički alocirani). Također, ni jedan od navedenih dijelova ne sadrži znak zarez (',').

- (a) (8 bodova) Potrebno je dizajnirati protokol za mrežnu aplikaciju koja služi za kupovinu ulaznica za koncert. Istaknimo da se u ovome podzadatku **ne očekuje** pisanje programskog koda! Klijent mora moći:
- Za izvođača i vrijeme (primjerice, za *Baby Lasagna* i 20:30) dobiti informaciju o svim koncertima tog izvođača koji počinju najranije u zadano vrijeme. Ako korisnik navede samo izvođača, potrebno mu je poslati informacije o svim koncertima tog izvođača.
 - Kupiti ulaznice za koncert. Prilikom kupovine ulaznica potrebno je navesti izvođača, datum, mjesto, vrijeme početka te željeni broj ulaznica. U slučaju uspješne kupovine, potrebno je vratiti kodove kupljenih ulaznica (svaka ulaznica ima svoj kod). Kod ulaznice je niz od 5 znamenki.

Osmislite vrste poruka i njihov format koje razmjenjuju klijent i server. Navedite kakvi se sve tipovi grešaka mogu dogoditi u komunikaciji. Odaberite primjer po jedne klijentske poruke za svaku od točaka (i) i (ii) (u skladu s vašim definicijama mogućih vrsta i formata poruka) te pripadne odgovore servera.

- (b) (5 + 7 = 12 bodova) Pretpostavimo da su implementirane funkcije za slanje i primanje poruke (s utičnice `sock`):

- `int posalji(int sock, const char *poruka)`
- `int primi(int sock, char **poruka)`

pri čemu u tim funkcijama `poruka` predstavlja cijelu poruku (zaglavlje + tijelo). Sami odredite kako su odijeljeni zaglavlje i tijelo, kao i njihove komponente. Funkcije vraćaju 1 ako je došlo do greške (inače 0).

- Implementirajte **sve** strukture/polja/varijable koje trebaju serveru za pamćenje svih navedenih podataka o koncertima.
- Implementirajte serversku funkciju `void kupi(int sock, char *poruka)` (pri čemu `poruka` kao i `gore` ima zaglavlje + tijelo) koja ažurira odgovarajuće strukture, polja i varijable na serveru te vraća poruku klijentu o uspješnoj kupovini ili neuspješnoj kupovini (u kom slučaju poruka sadrži opis greške).

JMBAG

IME & PREZIME

TREĆI ZADATAK

OSTVARENI BODOVI

 8

Pretpostavite da je zadan sljedeći HTML kod i CSS pravila. Na donjoj slici skicirajte izgled odgovarajuće stranice. Nekako, primjerice križanjem, naznačite obojana područja i napišite o kojoj je boji riječ.

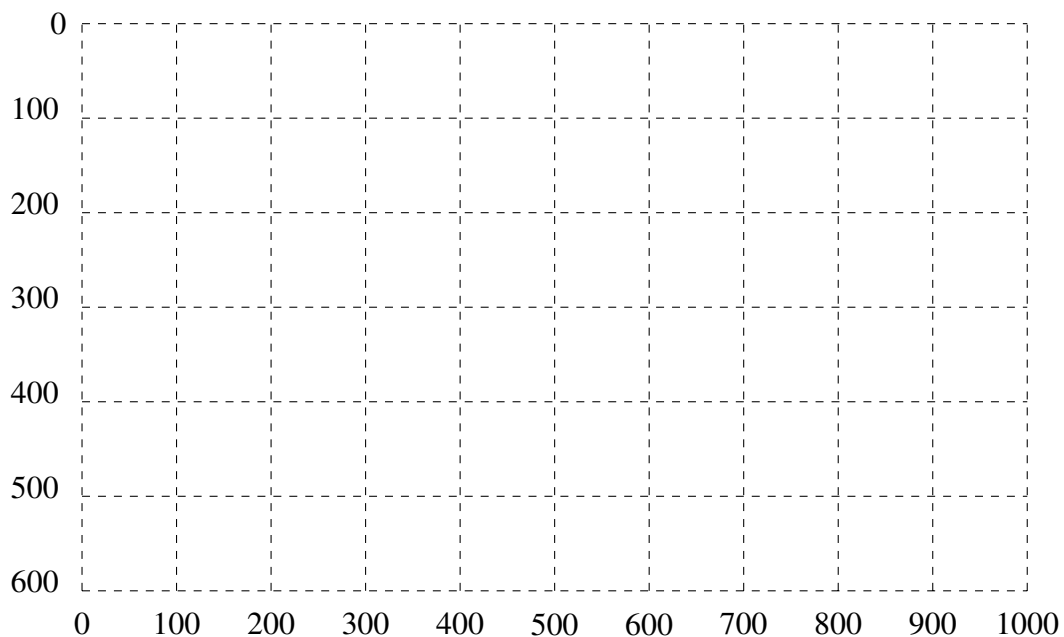
CSS

```
* {
  margin: 0;
  width: 1000px;
  height: 600px;
}
.Z {
  background-color: purple;
  height: 50%;
  width: 50%;
}
.Y {
  background-color: orange;
  height: 25%;
}
.X {
  float: right;
  background-color: green;
  height: 25%;
}
#gen {
  height: 50%;
}
.Z p {
  color: yellow;
}
```

HTML

```
...
<div class="Z">
  <div class="X" id="gen">
    <p>Lorem</p>
  </div>
</div>
<div class="Y">
  <p>Ipsum</p>
</div>
<div class="X">
  Dolor
</div>
```

Stranica dimenzija 1000 × 600 px:



JMBAG

IME & PREZIME

ČETVRTI ZADATAK

OSTVARENI BODOVI

	10
--	-----------

Navedite slojeve TCP/IP modela poredane tako da se na zadnjoj poziciji nalazi sloj najbliži krajnjem korisniku. Za svaki sloj koji sadrži protokole, navedite po dva primjera i objasnite što rade.

JMBAG

IME & PREZIME

PETI ZADATAK

OSTVARENI BODOVI

	10
--	----

Objasnite dinamičko usmjeravanje pomoću vektora udaljenosti, navedite prednosti i mane. Napišite pseudokod odgovarajućeg algoritma.

JMBAG

IME & PREZIME

ŠESTI ZADATAK

OSTVARENI BODOVI 12

Navedite i opišite sva značajna softverska rješenja (od hardverskog do aplikacijskog sloja) potrebna za implementaciju igre tipa Call of Duty (pucačina u prvom licu). Pretpostavka je da sjedite za računalom spojenim Ethernet kabelom na usmjernik.