

Mreže računala

Ispit, 18. veljače 2026. 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

20

Zadana je sljedeća struktura podataka koja predstavlja ocjene nekog studenta:

```
struct Ocjene {  
    int n;  
    char** kolegiji;  
    int* ocjene;  
};
```

Pritom je n broj ocjena, a $ocjene[i]$ predstavlja ocjenu iz kolegija čiji je naziv string $kolegiji[i]$. Server (s IP-adresom "123.4.23.11" i portom 65500) za svakog studenta (koji je jedinstveno određen stringom koji predstavlja njegov JMBAG) pamti po jednu strukturu tipa `struct Ocjene` koja sadrži podatke o ocjenama tog studenta.

Implementirajte klijentsku funkciju

```
int dohvatiOcjene(struct Ocjene* ocj)
```

koja se spaja na navedeni server s ocjenama, učitava od korisnika string koji predstavlja JMBAG (smijete pretpostaviti da je učitani JMBAG ispravnog formata, tj. učitano je točno 10 znamenaka) te ga šalje serveru. Zatim od servera prima podatke (prvo n , zatim nazive kolegija te naposljetku sve ocjene) i sprema ih na adresu ocj . Funkcija na kraju treba prekinuti komunikaciju sa serverom. Ukoliko student s navedenim JMBAG-om ne postoji ili nema ocjena, server kao vrijednost od n vraća broj 0 (te to u ovom zadatku ne tretiramo kao grešku).

Pazite na greške pri komunikaciji koje se mogu dogoditi. Funkcija mora vratiti 0 ako je sve dobro prošlo, inače 1.

Napomena: Za jedan cijeli broj tipa `int` možete pretpostaviti da može biti primljen/poslan jednom `recv/send` funkcijom, no to ne možete pretpostaviti za stringove. Pri slanju/primanju stringova, prvo je potrebno poslati/primiti njegovu duljinu, a zatim je potrebno poslati/primiti pojedini znak tog stringa.

DRUGI ZADATAK

OSTVARENI BODOVI

20

Mrežna aplikacija *KrKr* nudi registriranim korisnicima međusobno igranje igre *Križić Kružić*. Svaki korisnik ima svoje ime (dinamički alociran string) i lozinku (točno 8 znakova). Pojedinu igru igraju istovremeno dva korisnika, a svaki korisnik može istovremeno sudjelovati u najviše jednoj igri. Svakom pobjedom, korisnik dobiva jedan bod (izjednačenja ne računamo - samo pobjede!). Bodovi se pamte kako bi znali koji igrač je najuspješniji. Igra se odvija po standardnim pravilima igre *Križić Kružić* na 3×3 ploči (3×3 matrici znakova gdje znak '-' predstavlja prazno polje).

- (a) (8 bodova) Potrebno je dizajnirati protokol za mrežnu aplikaciju koja služi za odvijanje jedne igre. Istaknimo da se u ovome podzadatku **ne očekuje** pisanje programskog koda! Klijent mora moći:

- (i) Ulogirati se svojim korisničkim imenom i lozinkom.
- (ii) Dobiti top listu 10 najboljih igrača po broju bodova.
- (iii) Zatražiti da mu se dodijeli neki igrač za igru (nakon čega se igra pokreće).
- (iv) Dobiti trenutni prikaz ploče u igri te povući potez (primjerice, ako mu je dodijeljen u igri znak 'X' treba ga moći postaviti na polje s koordinatama (1,1), naravno ako je to polje slobodno - koordinate ovdje ne kreću od 0 nego od 1).
- (v) Saznati koliko trenutno ima bodova i odlogirati se (kad nije u nekoj igri).

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) - (v) (u skladu s vašim definicijama mogućih vrsta i formata poruka) te pripadne odgovore servera. Prikažite primjerom i neku grešku.

- (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).

- (i) Implementirajte **sve** strukture/polja/varijable koje trebaju serveru za pamćenje svih navedenih podataka o korisnicima i igrama.
- (ii) Implementirajte klijentsku funkciju `void povuciPotez(int sock, char *poruka)` (pri čemu **poruka** kao i gore ima zaglavlje + tijelo) koja:
 - 1. korisniku prikaže trenutno stanje ploče u igri (to treba dobiti od servera!),
 - 2. traži od korisnika da unese koordinate na koje želi postaviti svoj znak,
 - 3. ispisuje dobivenu poruku od servera - ako je riječ o grešci, onda samo ispiše opis greške koji je dobiven od servera, a inače ispisuje novo stanje na ploči. Primjerice:

```
X - 0
- X 0
X - -
```

TREĆI ZADATAK

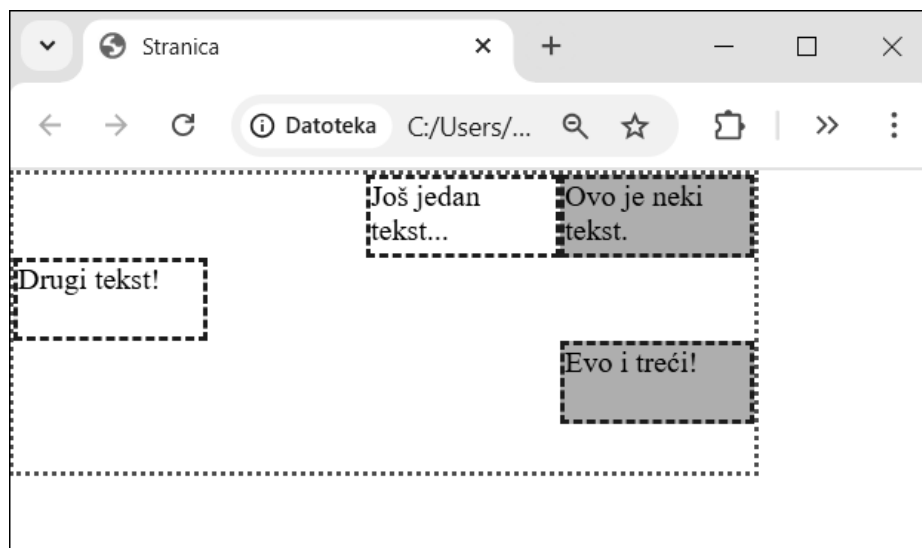
OSTVARENI BODOVI

8

Prikazan je sadržaj datoteke "stranica.html":

```
<!DOCTYPE html>
<html lang="hr">
  <head>
    <meta charset="UTF-8">
    <link href="stil.css">
    <title>Stranica</title>
  </head>
  <body>
    <div>
      <p id="A"> Ovo je neki tekst. </p>
      <p id="B"> Još jedan tekst... </p>
      <p class="C"> Drugi tekst! </p>
      <p class="C" id="C"> Evo i treći! </p>
    </div>
  </body>
</html>
```

- (a) (1 bod) Navedite koji HTML element ili atribut nedostaje u toj datoteci kako bi se dobivena stranica mogla bez grešaka validirati u validatoru za HTML5.
- (b) (6 bodova) Napišite sadržaj datoteke "stil.css" pomoću kojeg se dobiva sljedeći prikaz te stranice u web-pregledniku:



Pritom su sve linije debljine 3 piksela, isprekidane linije su plave boje, a ostale (točkaste) linije su crvene i debljine 2 piksela. Točkasta linija čini pravokutnik dimenzija 500 × 200 piksela. Sami približno odredite veličine pravokutnika unutar njega, pri čemu obavezno koristite relativne jedinice. Pozadine pravokutnika koji su na gornjoj slici osjenčani su narančaste boje. Uočite da niti jedan element na gore prikazanoj stranici nema margine!

- (c) (1 bod) Navedite CSS selektor kojim biste boju prve linije teksta "Još jedan tekst..." na gornjoj slici postavili na žutu boju.

JMBAG

IME & PREZIME

ČETVRTI ZADATAK

OSTVARENI BODOVI 10

Objasnite pojmove:

- a) LAN
- b) WAN
- c) Internet
- d) Mrežna aplikacija
- e) Peer-to-peer

JMBAG

IME & PREZIME

PETI ZADATAK

OSTVARENI BODOVI

	10
--	----

Objasnite bežične načine prijenosa podataka. Navedite prednosti i mane svakog od pristupa.

Navedite i objasnite najvažnije mjere za mjerenje performansi mreže.

JMBAG

IME & PREZIME

ŠESTI ZADATAK

OSTVARENI BODOVI

 12

Navedite i konceptualno objasnite glavna hardverska i tehnička dostignuća koja su omogućila slanje digitalnih poruka (ovdje se ne treba osvrnati na softverska rješenja). Opišite načine prijenosa digitalnih informacija i odgovarajuću hardversku infrastrukturu potrebnu za njihovo slanje. Treba spomenuti i opisati potreban hardver od najnižeg fizičkog sloja do (uključivo) Internet sloja.