

Mreže računala

Ispit, 18. veljače 2026. godine

RJEŠENJE PRVOG ZADATKA

UKUPNO BODOVA

20

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

```
    int mojSocket = socket(PF_INET, SOCK_STREAM, 0);
    if(mojSocket == -1)
        return 1;
```

Stvaranje utičnice za komunikaciju.
1 bod

```
    struct sockaddr_in adresaServera;
    adresaServera.sin_family = AF_INET;
    adresaServera.sin_port = htons(65500);
    if(inet_aton("123.4.23.11", &adresaServera.sin_addr) == 0) {
        close(mojSocket); return 1;
    }
    memset(adresaServera.sin_zero, '\\0', 8);
    if(connect(mojSocket, (struct sockaddr *) &adresaServera,
        sizeof(adresaServera)) == -1) {
        close(mojSocket); return 1;
    }
```

Spajanje na server.

4 boda

```
    char jmbag[11];
    scanf("%s", jmbag);
```

Učitavanje JMBAG-a od korisnika
1 bod

```
    int duljJmbaga = 10;
    if(send(mojSocket, &duljJmbaga, sizeof(int), 0) <= sizeof(int)) {
        close(mojSocket); return 1;
    }
```

Slanje duljine JMBAG-a serveru*
1 bod

(*Prema tekstu zadatka, pri slanju/primanju stringa se prvo šalje/prima njegova duljina, a zatim se šalje/prima znak po znak tog stringa.)

```
    int i;
    for(i = 0; i < 10; ++i) {
        if(send(mojSocket, &jmbag[i], sizeof(char), 0) != sizeof(char)) {
            close(mojSocket); return 1;
        }
    }
```

Slanje JMBAG-a serveru

2 boda

```
    int nn;
    if(recv(mojSocket, &nn, sizeof(int), 0) != sizeof(int)) {
        close(mojSocket); return 1;
    }
    int n = htonl(nn);
```

Primanje broja n od servera

2 boda

```

int j, duljina, duljina_n;
ocj->kolegiji = (char**)malloc(n * sizeof(char*));
for(i = 0; i < n; ++i) {
    if(recv(mojSocket, &duljina_n, sizeof(int), 0)
        != sizeof(int)) {
        close(mojSocket); return 1;
    }
    duljina = htonl(duljina_n);
    ocj->kolegiji[i] = (char*)malloc((duljina+1) * sizeof(char));
    for(j = 0; j < duljina; ++j) {
        if(recv(mojSocket, &ocj->kolegiji[i][j], sizeof(char), 0)
            != sizeof(int)) {
            close(mojSocket); return 1;
        }
    }
    ocj->kolegiji[i][duljina] = '\\0';
}

```

Alociramo
mjesto za n na-
ziva kolegija. Za
svaki naziv, prvo
se prima njegova
duljina, a zatim
znak po znak na-
ziva.

5 bodova

```

ocj->ocjene = (int*)malloc(n * sizeof(int));
for(i = 0; i < n; ++i) {
    if(recv(mojSocket, &nn, sizeof(int), 0)
        != sizeof(int)) {
        close(mojSocket); return 1;
    }
    ocj->ocjene[i] = ntohl(nn);
}

```

Alociramo mjesto za n
ocjena. Zatim se primi
svaka od njih.

3 boda

```

if(close(mojSocket) == -1)
    return 1;
return 0;

```

Zatvaranje utičnice.
1 bod

}



RJEŠENJE DRUGOG ZADATKA

UKUPNO BODOVA

20

(a) podzadatak

Svaka poruka sastoji se od 2 dijela: zaglavlja i tijela poruke. Zaglavlje se sastoji od dva integera: prvi predstavlja duljinu poruke (ne uključujući zaglavlje), a drugi predstavlja kod kojim se jednoznačno određuje vrsta poruke.

1 bod

Vrste poruka između servera i klijenta su:

3 boda

- (1) *LOGIN ime lozinka* - klijent šalje serveru svoje ime i lozinku kako bi se ulogirao
- (2) *TOP_LISTA* - klijent šalje serveru poruku da želi dobiti top listu 10 najboljih igrača po broju bodova
- (3) *TOP_LISTA_R ime₁ ime₂ ... ime₁₀* - server šalje klijentu 10 najboljih igrača poredanih po broju bodova
- (4) *DODIJELI* - klijent šalje serveru poruku da želi da mu se dodijeli neki igrač za igru
- (5) *PRIKAŽI* - klijent šalje serveru poruku da želi dobiti trenutni prikaz ploče
- (6) *PLOČA matrica_znakova* - server šalje klijentu trenutni prikaz ploče (string koji program može samo ispisati na ekran u obliku matrice)
- (7) *POTEZ x_koordinata y_koordinata* - klijent šalje serveru zahtjev za postavljanje svog znaka na koordinate (x, y) na ploči
- (8) *BODOVI* - klijent želi od servera saznati koliko ima bodova
- (9) *BODOVI_R bodovi* - server šalje klijentu koliko ima bodova
- (10) *ODLOGIRAJ* - klijent šalje serveru poruku da se želi odlogirati
- (11) server na **svaki** klijentov zahtjev odgovara je li zahtjev uspio ili nije - odgovor je oblika: *ODGOVOR poruka* gdje je poruka string koji je jednak "OK" ako je zahtjev uspješno obrađen, a inače sadrži opis greške.

Tipovi grešaka u komunikaciji:

1 bod

- klijent šalje nepostojeće ime ili pogrešnu šifru,
- klijent tijekom igre želi do-
djelu igrača za drugu igru,
- klijent šalje koordinate izvan
raspona ili koje nisu slobodne,
- nema barem 10 igrača za top
listu,
- klijent želi prikaz ploče ili
povući potez kad nije u igri,
- klijent se želi odlogirati kad je
u igri.

Navodimo po jedan primjer klijentske poruke za točke (i)-(v) te pripadne odgovore od strane servera:

2 boda

- | | |
|---|---|
| (i) klijent: [11,LOGIN] igrač1 12ab
server: [2,ODGOVOR] OK | (iv) klijent: [3,POTEZ] 1 2
server: [2,ODGOVOR] OK |
| (ii) klijent: [0,TOP_LISTA]
server: [21,ODGOVOR] Nema dovoljno igrača. | (v) klijent: [0,ODLOGIRAJ]
server: [2,ODGOVOR] OK |
| (iii) klijent: [0,DODIJELI]
server: [2,ODGOVOR] OK | |

Jedan primjer greške prikazan je u prethodnim primjerima pod stavkom (ii).

1 bod

(b) podzadatak

- (i) Podatke o korisnicima i igrama možemo pamtit na sljedeći način:

```
typedef struct {
    char* ime;
    char lozinka[8]; // ili [9] ako je string
    int bodovi;
} korisnik;
```

Za podatke o jednom korisniku.

2 boda

```
typedef struct {
    korisnik *prvi, *drugi;
    char ploca[3][3];
} igra;
```

Za podatke o jednoj igri.

2 boda

```
int ukupno_korisnika, ukupno_igara;
korisnik* korisnici;
igra* igre;
```

Svi korisnici i sve igre koje se trenutno odvijaju.

1 bod

- (ii) Zaglavlje i tijelo poruke, kao i njihove komponente, će u sljedećem kodu biti odvojeni znakom zareza (',').

```
void povuciPotez(int sock, char *poruka) {
```

```
    sprintf(poruka, "0,%d", PRIKAŽI);
    posalji(sock, poruka);
    primi(sock, &poruka);
```

Slanje i primanje poruke.

1 bod

```
    int duljina, tipPoruke;
    char* tijeloPoruke = (char*)malloc(strlen(poruka)*sizeof(char));
    sscanf(poruka,"%d,%d,%[^,]", &duljina, &tipPoruke, tijeloPoruke);
```

Čitanje poruke.

2 boda

```
    printf("%s", tijeloPoruke); // to je stanje ploče ili opis greške
    if(tipPoruke != PLOČA)
        return; // ako je bila greška, gotovi smo
```

Ispis poruke.

1 bod

```
    int x, y;
    scanf("%d %d", &x, &y);
```

Unos koordinata.

1 bod

```
    sprintf(poruka, "%d,%d", x, y);
    int duljinaTijela = strlen(poruka);
    sprintf(poruka, "%d,%d,%d,%d", duljinaTijela, PRIKAŽI, x, y);
    posalji(sock, poruka);
```

Sastavljanje i slanje poruke.

1 bod

```
    primi(sock, &poruka);
    sscanf(poruka,"%d,%d,%[^,]",&duljina,&tipPoruke,tijeloPoruke);
    printf("%s", tijeloPoruke); // stanje ploče ili opis greške
```

Primanje i čitanje poruke.

1 bod

```
}
```

(a) Kako bi se stranica mogla bez grešaka validirati u validatoru za HTML5 nedostaje:

rel atribut za <link> element

1 bod

(preciznije, treba pisati: <link rel="stylesheet" href="stil.css">).

(b) Jedan mogući sadržaj datoteke "stil.css":

```
div {  
  border: 2px dotted red;  
  width: 500px;  
  height: 200px;  
}
```

1 bod

```
p {  
  border: 3px dashed blue;  
  width: 25%;  
  height: 25%;  
}
```

1 bod

```
#A {  
  float: right;  
}  
#B {  
  float: right;  
}  
.C {  
  clear: right;  
}  
#C {  
  float: right;  
}
```

2 boda

```
* {  
  margin: 0;  
}
```

1 bod

```
#A, #C {  
  background-color: orange;  
}
```

1 bod

(c) Traženi CSS selektor je sljedeći:

#B::first-line

1 bod