

JMBAG

IME & PREZIME

Mreže računala

Prvi kolokvij, 19. studenoga 2020. godine

Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i službeni šalabahter. Predajete samo papire koje ste dobili. Kolokvij ima ukupno **50 bodova**, međutim konačni broj bodova se računa kao $\min(40, \#bodova)$.

PRVI ZADATAK

OSTVARENI BODOVI

| | |
|--|---|
| | 8 |
|--|---|

Napišite dio serverskog koda od trenutka prihvaćanja konekcije do trenutka prekidanja konekcije (uključivo) koji

- od klijenta prima broj n i string duljine n
- ako je taj string validan host-name nekog računala šalje klijentu dekadsku IP-adresu tog računala (kao string), a u suprotnom, obavijest da nije primljen validan host-name.

Nakon toga server prekida konekciju. Možete pretpostaviti da se svaki broj i svaki string mogu poslati/primiti jednom *send/recv* naredbom.

JMBAG

IME & PREZIME

DRUGI ZADATAK

OSTVARENI BODOVI

 12

U ovom zadatku se **ne očekuje** pisanje programskog koda.

Potrebno je dizajnirati protokol za mrežnu aplikaciju koja služi za rezervaciju turističkog smještaja i popratnih aktivnosti. Klijent mora moći:

1. Dobiti popis:

- hotela, hostela ili stanova u kojima ima slobodnih mjesta za prihvat turista. Za svaki takav smještaj, klijent treba primiti: adresu (ime ulice, broj, grad, država), tip smještaja (hotel, hostel, stan) i broj slobodnih mjesta.
 - smještaja koji imaju bazen. Za svaki takav smještaj, klijent treba primiti: adresu, tip smještaja, broj slobodnih mjesta i veličinu bazena (dimenzije u obliku visina, širina, dubina).
2. Rezervirati boravak u zadanom smještaju (pretpostavimo da je identificiran jedinstvenom adresom). Prilikom rezervacije obavezno treba specificirati OIB i godinu rođenja osobe koja vrši rezervaciju.
 3. Dobiti načine prijevoza od zadane adrese do zadanog smještaja. Server šalje adresu i GPS koordinate u obliku: (xy.abcde, kl.fghij) najbliže taksi službe, željezničke, autobusne ili tramvajske stanice. Ukoliko se radi o željezničkoj, autobusnoj ili tramvajskoj stanici, šalje i broj linije koju klijent treba koristiti. Za sve tipove prijevoza šalje i očekivano trajanje vožnje (u minutama) do izabranog smještaja.
 4. Zatražiti preporuku za jednodnevni izlet. Server šalje niz trojki oblika (ime odredišta, udaljenost, kratki opis). Kratki opis se sastoji od liste svih gradova koji će se posjetiti na izletu.
 5. Zatražiti izvještaj o troškovima boravka. Server šalje duljinu boravka (dani, sati i minute), ukupnu cijenu te specifikaciju troškova. Specifikacija troškova je lista parova oblika (usluga, cijena), gdje usluga \in {najam, obroci, bazen, bar, izleti}.

U cijeloj komunikaciji korisnik se početno mora ulogirati svojim korisničkim imenom i lozinkom.

Osmislite vrste poruka i njihov format (zaglavlje/*header* i tijelo/*payload*) koje razmjenjuju klijent i server.

Kratko navedite kakvi se sve tipovi grešaka mogu dogoditi u komunikaciji.

Odaberite primjer po jedne klijentske poruke za svaku točku pod 1. i svaku od preostale 4 funkcionalnosti (primjeri u skladu s vašim definicijama mogućih vrsta i formata poruka), te pripadne odgovore od strane servera.

JMBAG

IME & PREZIME

TREĆI ZADATAK

OSTVARENI BODOVI 17

Knjižnica *Tolkien* posjeduje mrežnu aplikaciju pomoću koje članovi te knjižnice mogu rezervirati neku knjigu iz te knjižnice. Svaki član prilikom učlanjenja u tu knjižnicu dobiva svoje jedinstveno korisničko ime (od točno 8 znakova) i svoju lozinku (od točno 8 znakova) za korištenje te aplikacije. Korisnik prvo rezervira knjigu u aplikaciji, a zatim ju u roku od 3 dana mora doći podići u knjižnicu (inače se rezervacija briše). U svakom trenutku jedan korisnik može imati najviše 3 rezervirane knjige.

Knjižnica dopušta da upisano bude **najviše** 200 članova, a sadrži **najviše** 1000 različitih naslova, od kojih je svaki u više primjeraka. Za svaki naslov, osim jedinstvenog naslova (npr. *Dvije kule*) i broja primjeraka tog naslova koji se trenutno nalaze u knjižnici (tj. nisu posuđeni), aplikacija pamti i autora, godinu izdanja te broj stranica. Pretpostavite da naslov i autor nemaju više od 30 znakova svaki.

Protokol za komunikaciju serverskog i klijentskog programa definiran je na sljedeći način:

- Zaglavlje poruke sastoji se od koda (koji određuje vrstu poruke), korisničkog imena i lozinke.
- Vrste poruka uključuju sljedeće (u drugom stupcu je naveden kod):

| | | |
|---------------------------|---|---|
| <i>UPIT naslov</i> | 1 | korisnik želi sve informacije o danom naslovu |
| <i>UPIT_R informacije</i> | 2 | server šalje sve informacije o traženom naslovu |
| <i>REZERVIRAJ naslov</i> | 3 | korisnik želi rezervirati jedan primjerak danog naslova |
| <i>ODGOVOR poruka</i> | 4 | server šalje string koji je jednak "OK" za uspješno obrađen zahtjev ili opis greške |

- 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) gdje su zaglavlje i tijelo, kao i njihove komponente, odijeljeni znakom razmaka. Funkcije vraćaju 1 ako je došlo do greške (inače 0).

1. Implementirajte **sve** strukture/polja/varijable koje trebaju serveru za pamćenje svih navedenih podataka o članovima, naslovima, te postojećim rezervacijama. 4 boda
2. Implementirajte serversku funkciju `void rezerviraj(int sock, char *poruka)` (oprez: `poruka` kao i `gore` ima zaglavlje + tijelo) koja ažurira odgovarajuća polja i strukture na serveru te vraća poruku klijentu o uspješnoj rezervaciji ili neuspješnoj rezervaciji (u kom slučaju poruka sadrži opis greške). Rezervacija nije uspjela ako: ne postoji korisničko ime iz zaglavlja ili lozinka ne odgovara tom korisničkom imenu, ne postoji taj naslov u knjižnici ili su svi primjerci tog naslova već rezervirani, ili korisnik već ima rezervirane 3 knjige. 7 bodova
3. Implementirajte klijentsku funkciju `void informacije(int sock)` koja od korisnika učitava njegovo korisničko ime, lozinku i neki naslov. Zatim funkcija pita server za sve informacije koje ima o tome naslovu. Iz pristigle poruke treba izvaditi sve informacije o tom naslovu i ispisati ih klijentu (ili u slučaju greške ispisati poruku o grešci). 6 bodova

JMBAG

IME & PREZIME

ČETVRTI ZADATAK

OSTVARENI BODOVI

Zadana je sljedeća struktura podataka koja reprezentira funkciju kojoj su domena i kodomena skup $\{0, 1, \dots, n\}$. Garantirano je da je ta funkcija bijekcija, dakle, ta funkcija je permutacija.

```
struct Permutation {
    int n;
    int* values;
};
```

n je broj spomenut u gornjem tekstu. Ako funkciju f prezentiramo elementom f tipa `Permutation`, tada je $f.values[i] = j$ ako i samo ako je $f(i) = j$.

Koristeći pretpostavku da se jedan `int` može poslati jednom naredbom `send` i primiti jednom naredbom `recv`, implementirajte funkcije:

- `int sendComposition(int sock, struct Permutation f, struct Permutation g)`
- `int receiveComposition(int sock, struct Permutation* h)`

Funkcija `sendComposition` uzima dva elementa tipa `Permutation` i mora poslati njihovu kompoziciju $f \circ g$.

Funkcija `receiveComposition` mora tu kompoziciju primiti i pobrinuti se da ona završi u elementu `h`.

Pazite na greške koje se mogu dogoditi. Ne treba provjeravati jesu li f i g zaista bijekcije sa zadanom domenom i kodomenom. Svaka funkcija mora vratiti 0 ako je sve dobro prošlo, inače 1.

Upute: pobrinite se da polja strukture `Permutation` koja koristite budu adekvatne veličine (alocirajte potrebne podatke). Strukture možete slati element po element, ali ne cijele odjednom.