

MREŽE RAČUNALA – 1. KOLOKVIJ

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 $\min(40, \# \text{bodova})$.

ZADATAK 1

1

(8 bodova)

Napišite dio serverskog koda od trenutka prihvaćanja konekcije do trenutka prekidanja konekcije (uključivo) koji šalje klijentu broj njegovih imena (*hostname*), te sva imena. Nakon toga server prekida konekciju. Možete pretpostaviti da se svaki broj i svaki string mogu poslati jednom *send* naredbom.

MREŽE RAČUNALA – 1. KOLOKVIJ – ZADATAK 2

(13 bodova)

Potrebno je dizajnirati protokol za rezervaciju mjesta za predstave u Hrvatskom narodnom kazalištu u Zagrebu. Možete pretpostaviti da se svaki dan izvodi neka predstava, ali se ista predstava ne izvodi više puta dnevno. Klijent treba moći sljedeće:

1. Dobiti aktualni raspored predstava koji se nalazi na serveru, tj. popis svih trojki: naziv predstave, datum, vrijeme početka predstave.
2. Dobiti popis svih naziva predstava zajedno s vremenima početaka za dani datum.
3. Dobiti popis slobodnih mjesta za dani naziv predstave na dani datum.
4. Rezervirati slobodno mjesto (jedno!) za izvedbu predstave na dani datum. Ako je rezervacija uspješna, dobiva se kod (niz od 10 slova i brojki) koji je jedinstven za pojedinu rezervaciju.
5. Otkazati postojeću rezervaciju (danu kodom).
6. Završiti komunikaciju.

U cijeloj komunikaciji korisnik se početno mora ulogirati svojim korisničkim imenom i lozinkom. Na predstave referiramo jednostavno njihovim nazivom. Datumi su oblika *xx.yy.zzzz*, a vrijeme početka predstave je oblika *hh:mm*. Na mjesta u kazalištu referiramo oznakom od tri znaka oblika *Xyy*, gdje je *X* oznaka kategorije B (balkon), L (loža), P (parket), a *yy* je neki dvoznamenkasti broj mjesta u kategoriji *X* (od 00 do 99).

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 od prvih pet funkcionalnosti (primjeri u skladu s vašim definicijama mogućih vrsta i formata poruka), te pripadne odgovore od strane servera.

MREŽE RAČUNALA – 1. KOLOKVIJ – ZADATAK 3

(17 bodova)

Tvrtka posjeduje pedeset službenih automobila koji su identificirani registracijskom oznakom (npr. ZG5534XD). Serverska aplikacija simulira rad tehničke službe te tvrtke u kojoj korisnici (identificirani svojim korisničkim imenom) mogu zauzeti neko od tih vozila preko Interneta (pri čemu ključ podižu osobno kod domara, nakon uspješnog zauzimanja vozila putem Interneta). Protokol za komunikaciju serverskog i klijentskog programa je definiran na sljedeći način:

- Zaglavlje poruke se sastoji od koda zahtjeva, korisničkog imena i datuma zauzimanja automobila.
- Mogući kodovi zahtjeva u poruci su 1 - zauzimanje automobila, 2 - vraćanje automobila, 3 - zauzimanje automobila uspješno provedeno, 4 - zauzimanje automobila neuspješno provedeno, 5 - povrat automobila uspio.
- U donjim funkcijama poruka predstavlja cijelu poruku (zaglavlje + tijelo) gdje su zaglavlje i tijelo, kao i njihove komponente, odijeljeni znakom razmaka. Pretpostavite da korisnik u jednom trenutku može zauzeti maksimalno jedan automobil.

Pod pretpostavkom da su implementirane funkcije: `int posaljiPoruku(int sock, const char *poruka)` i `int primiPoruku(int sock, char **poruka)`, implementirajte:

1. strukture `automobil` i `zauzimanje` koje pamte potrebne podatke o automobilu te trenutno zauzetim automobilima. Definirajte i pripadajuća polja koja pamte postojeće automobile i zauzeća. (4 boda)
2. serversku funkciju `void obradiZauzimanje(int sock, char *poruka)` koja ažurira odgovarajuća polja i strukture na serveru te vraća poruku o (ne)uspješno provedenom zauzimanju klijentu. Zauzimanje nije uspjelo ukoliko zatraženo vozilo ne postoji unutar tvrtke (ili je trenutno zauzeto). (7 bodova)
3. klijentsku funkciju `void provjeriPovrat(int sock)` koja pita server je li ključ automobila koji klijent vraća predan domaru. Funkcija iz tijela pristigle poruke koje je u formatu: `"OK:#dani"` ili `"NIJEOK:-1"` - ili javlja da povrat još nije obavljen ili izračunava i ispisuje iznos potencijalne kazne. Ukoliko je automobil bio kod korisnika više od 2 dana, korisnik plaća kaznu od 100 kn/dan. (6 bodova)

MREŽE RAČUNALA – 1. KOLOKVIJ – ZADATAK 4

(12 bodova)

Zadana je sljedeća struktura podataka koja reprezentira kvadratnu matricu. `dimenzija` nam kaže koliki je broj redaka i stupaca matrice. Dvodimenzionalno polje `elementi` sadrži elemente matrice:

```
struct Matrix {
    int dimenzija;
    int** elementi;
};
```

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

- `int sendProduct(int sock, struct Matrix A, struct Matrix B)`
- `int receiveProduct(int sock, struct Matrix* C)`

Funkcija `sendProduct` uzima dva elementa tipa `Matrix` i mora poslati njihov umnožak (AB). Funkcija `receiveProduct` mora taj produkt primiti i pobrinuti se da on završi u elementu `C`. Pazite na greške koje se mogu dogoditi. Svaka funkcija mora vratiti 0 ako je sve dobro prošlo, inače 1.

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

MREŽE RAČUNALA – 1. KOLOKVIJ

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 $\min(40, \# \text{bodova})$.

ZADATAK 1

1

(8 bodova)

Napišite dio serverskog koda od trenutka prihvaćanja konekcije do trenutka prekidanja konekcije (uključivo) koji šalje klijentu broj njegovih IP adresa, te sve te adrese (u dekadskom obliku). Nakon toga server prekida konekciju. Možete pretpostaviti da se svaki broj i svaki string mogu poslati jednom *send* naredbom.

MREŽE RAČUNALA – 1. KOLOKVIJ – ZADATAK 2

(13 bodova)

Potrebno je dizajnirati protokol za rezervaciju mjesta za projekcije filmova u Cinestar kinu u Branimirovoj. Možete pretpostaviti da se svaki dan prikazuje neki film, ali se isti film ne prikazuje više puta dnevno (u stvarnosti je malo drugačije). Klijent treba moći sljedeće:

1. Dobiti aktualni raspored projekcija koji se nalazi na serveru, tj. popis svih četvorki: naziv filma, datum, vrijeme početka prikazivanja i broj dvorane u kojoj se film prikazuje.
2. Dobiti popis svih naziva filmova koji se prikazuju zajedno s vremenima početaka i brojevima dvorana za dani datum.
3. Dobiti popis slobodnih mjesta u dvorani za dani naziv filma na dani datum (pod dvoranom se misli na onu u kojoj je na serveru zabilježeno da se u njoj tog dana prikazuje film).
4. Rezervirati slobodna mjesta (ali najviše 3) za projekciju filma na dani datum. Ako je rezervacija uspješna, dobiva se kod (niz od 20 slova i brojki) koji je jedinstven za pojedinu rezervaciju.
5. Otkazati postojeću rezervaciju (danu kodom).
6. Završiti komunikaciju.

U cijeloj komunikaciji korisnik se početno mora ulogirati svojim korisničkim imenom i lozinkom. Na filmove referiramo jednostavno njihovim nazivom. Datumi su oblika *xx.yy.zzzz*, a vrijeme početka prikazivanja filma je oblika *hh:mm*. Na dvorane referiramo brojevima od 1 do 13, a na mjesta u dvorani jednostavno brojevima od 1 do ukupnog kapaciteta dvorane koji je spremljen na serveru.

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 od prvih pet funkcionalnosti (primjeri u skladu s vašim definicijama mogućih vrsta i formata poruka), te pripadne odgovore od strane servera.

MREŽE RAČUNALA – 1. KOLOKVIJ – ZADATAK 3

(17 bodova)

Tvrtka posjeduje dvadeset konferencijskih dvorana koje su identificirane svojim imenom (jedna riječ od 8 znakova, npr. Velebajt). Serverska aplikacija simulira rad tehničke službe te tvrtke u kojoj korisnici (identificirani svojim korisničkim imenom) mogu zauzeti jednu od tih dvorana preko Interneta (pri čemu ključ podižu osobno kod domara, nakon uspješnog zauzimanja dvorane putem Interneta). Protokol za komunikaciju serverskog i klijentskog programa je definiran na sljedeći način:

- Zaglavlje poruke se sastoji od koda zahtjeva, korisničkog imena i datuma zauzimanja dvorane.
- Mogući kodovi zahtjeva u poruci su 1 - zauzimanje dvorane, 2 - oslobađanje dvorane, 3 - zauzimanje dvorane uspješno provedeno, 4 - zauzimanje dvorane neuspješno provedeno, 5 - oslobađanje dvorane uspješno.
- U donjim funkcijama poruka predstavlja cijelu poruku (zaglavlje + tijelo) gdje su zaglavlje i tijelo, kao i njihove komponente, odijeljeni znakom razmaka. Pretpostavite da korisnik u jednom trenutku može zauzeti maksimalno jednu dvoranu.

Pod pretpostavkom da su implementirane funkcije: `int posaljiPoruku(int sock, const char *poruka)` i `int primiPoruku(int sock, char **poruka)`, implementirajte:

1. strukture `dvorana` i `zauzimanje` koje pamte potrebne podatke o dvorani te trenutno zauzetim dvoranama. Definirajte i pripadajuća polja koja pamte postojeće dvorane i zauzeća. (4 boda)
2. serversku funkciju `void obradiZauzimanje(int sock, char *poruka)` koja ažurira odgovarajuća polja i strukture na serveru te vraća poruku o (ne)uspješno provedenom zauzimanju klijentu. Zauzimanje nije uspješno ukoliko zatražena dvorana ne postoji unutar tvrtke (ili je trenutno zauzeta). (7 bodova)
3. klijentsku funkciju `void provjeriOslobadjanje(int sock)` koja pita server je li ključ dvorane koju klijent oslobađa predan domaru. Funkcija iz tijela pristigle poruke koje je u formatu: "OK:#dani" ili "NIJEOK:-1" - ili javlja da ključ još nije vraćen ili izračunava i ispisuje iznos potencijalne kazne. Ukoliko je ključ dvorane bio kod korisnika više od 3 dana, korisnik plaća kaznu od 499 kn/dan. (6 bodova)

MREŽE RAČUNALA – 1. KOLOKVIJ – ZADATAK 4

(12 bodova)

Zadana je sljedeća struktura podataka koja reprezentira matricu. `retci` nam kaže koliki je broj redaka, a `stupci` broj stupaca matrice. Dvodimenzionalno polje `elementi` sadrži elemente matrice:

```
struct Matrix {
    int retci;
    int stupci;
    int** elementi;
};
```

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

- `int sendSum(int sock, struct Matrix A, struct Matrix B)`
- `int receiveSum(int sock, struct Matrix* C)`

Funkcija `sendSum` uzima dva elementa tipa `Matrix` i mora poslati njihovu sumu (zbrajaju se elementi na podudarnim pozicijama). Funkcija `receiveSum` mora tu sumu primiti i pobrinuti se da ona završi u elementu `C`. Pazite na greške koje se mogu dogoditi. Svaka funkcija mora vratiti 0 ako je sve dobro prošlo, inače 1.

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