

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

Uvid i rezultati: Rezultati će biti objavljeni na web stranicama kolegija do 07.12.2018 u ponoć. Uvidi u kolokvije će se održati ?? .12.2018. Točan termin i predavaonica će biti objavljeni na web stranici kolegija.

ZADATAK 1

1

(8 bodova)

Napišite funkciju `void ispis(char *hostname)` koja prima ime nekog računala. Funkcija treba ispisati prvu IP adresu zadanog računala u dekadskom obliku, te sva njegova imena.

MREŽE RAČUNALA – 1. KOLOKVIJ – ZADATAK 2

(13 bodova)

Potrebno je dizajnirati protokol za rad s informacijama o studentskim statusima osoba u Hrvatskoj. Klijent treba moći sljedeće.

1. Dobiti popis svih fakulteta koji su pohranjeni na serveru, uz primjenu nula ili više sljedećih mogućih filtara:
 - samo fakulteti koji imaju barem jednog studenta,
 - samo fakulteti koji pripadaju nekom akademskom području. Pretpostavite da su moguća područja: prirodne znanosti, humanističke znanosti, i društvene znanosti, i da svaki fakultet pripada točno jednom području.
2. Dobiti popis svih fakulteta na koje je dana osoba upisana.
3. Dodati osobu u evidenciju pojedinog fakulteta.
4. Izbrisati osobu iz evidencije pojedinog fakulteta.
5. Završiti komunikaciju.

U cijeloj komunikaciji na osobe možete referirati pomoću npr. OIB-a, a na fakultete pomoću MB-a. Svaki fakultet ima svoju lozinku (pohranjenu na serveru), i poruke za rad s evidencijom moraju sadržavati tu lozinku.

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

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

Napišite primjer po jedne klijentske poruke za svaku od prvih četiri 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)

Serverska aplikacija simulira rad knjižnice iz koje korisnici (identificirani svojim korisničkim imenom) mogu posuditi knjigu preko Interneta (fizička kopija knjige se dostavlja poštom na kućnu adresu). Protokol za komunikaciju serverskog i klijentskog programa je definiran na sljedeći način:

- Zaglavlje poruke se sastoji od koda zahtjeva, korisničkog imena i vremena početka slanja zahtjeva.
- Mogući kodovi zahtjeva u poruci su 1 - posudba knjige, 2 - vraćanje knjige, 3 - posudba uspješno provedena, 4 - posudba neuspješno provedena, 5 - povrat knjige uspio.
- U donjim funkcijama poruka predstavlja cijelu poruku (zaglavlje + tijelo) gdje su zaglavlje i tijelo, kao i njihove komponente, odijeljeni znakom razmaka. Naslov knjige je jedna riječ bez razmaka. Pretpostavite da korisnik u jednom trenutku može posuditi maksimalno jednu knjigu.

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

1. strukture `knjiga` i `posudba` koje pamte potrebne informacije o broju knjiga koje se mogu posuditi te trenutno posuđenim knjigama. Definirajte i pripadajuća polja koja pamte postojeće knjige i posudbe. (4 boda)
2. serversku funkciju `void obradiPosudbu(int sock, char *poruka)` koja ažurira odgovarajuća polja strukture `knjiga` i `posudbi` na serveru te vraća poruku o (ne)uspješnoj provedenoj posudbi klijentu. Posudba je neuspjela ukoliko zatražena knjiga ne postoji u knjižnici. Polje struktura `knjiga` je sortirano silazno po broju kopija određenog naslova dostupnog u knjižnici. (7 bodova)
3. klijentsku funkciju `void provjeriPovrat(int sock)` koja pita server je li knjiga koju klijent vraća (poštom) stigla u knjižnicu. Funkcija iz tijela pristigle poruke koje je u formatu: "OK:#dana" ili "NIJEOK:-1" ili javlja da povrat još nije obavljen ili izračunava i ispisuje iznos potencijalne zakasnine. Ukoliko je knjiga bila u rukama korisnika više od 90 dana, korisnik plaća zakasninu od 20 lipa/dan. (6 bodova)

MREŽE RAČUNALA – 1. KOLOKVIJ – ZADATAK 4

(12 bodova)

1. (4 bodova) napišite funkcije

- `void posalji(int sock, char *buffer, int duljinaPoruke)`
- `void primi(int sock, char *buffer, int duljinaPoruke)`

Uputa: ovdje slobodno pretpostavite da je `buffer` već alociran i dovoljno velik da spremi cijelu primljenu poruku.

2. (8 bodova) Zadana je sljedeća struktura podataka koja reprezentira čvor u nekom grafu:

```
struct Cvor {
    char* ime;
    int br_povezanih;
    char **imena_povezanih;
};
```

Koristeći funkcije iz prve točke, implementirajte:

- `void posaljiCvor(int sock, struct Cvor cvor)`
- `void primiCvor(int sock, struct Cvor *cvor)`

Upute: pobrinite se da polja strukture `Cvor` koja koristite budu adekvatne veličine (alocirajte potrebne podatke). Ovdje ne morate paziti na greške u slanju i primanju. Smijete slati i primiti podatke koji ne pišu u strukturi. Sva slanja i primanja realizirajte koristeći funkcije `posalji` i `primi` (čak i ako ih niste implementirali). 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 $\min(40, \# \text{bodova})$.

Uvid i rezultati: Rezultati će biti objavljeni na web stranicama kolegija do 07.12.2018 u ponoć. Uvidi u kolokvije će se održati ?? .12.2018. Točan termin i predavaonica će biti objavljeni na web stranici kolegija.

ZADATAK 1

1

(8 bodova)

Napišite funkciju `void ispis(char *dekadskiIP)` koja prima IP adresu računala zapisanu u dekadskom obliku. Funkcija treba ispisati sva imena tog računala.

Uputa: pogledajte dostupne funkcije na službenom podsjetniku.

MREŽE RAČUNALA – 1. KOLOKVIJ – ZADATAK 2

(13 bodova)

Potrebno je dizajnirati protokol za rad s informacijama o azilima (skloništima) za životinje u Hrvatskoj. Klijent treba moći sljedeće.

1. Dobiti popis svih azila koji su pohranjeni na serveru, uz primjenu nula ili više sljedećih mogućih filtara:
 - samo azili na području Zagreba,
 - samo azili koji nisu u Zagrebu,
2. Dobiti popis svih životinja u danom azilu.
3. Dodati životinju u evidenciju pojedinog azila.
4. Izbrisati životinju iz evidencije pojedinog azila.
5. Završiti komunikaciju.

U cijeloj komunikaciji na azile možete referirati pomoću npr. OIB-a ustanove, na životinje pomoću MB-a, a na gradove pomoću dvoslovne kratice grada. Svaki azil ima svoj četveroznamenkasti PIN (pohranjen na serveru), i poruke za rad s evidencijom moraju sadržavati taj PIN.

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 četiri 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)

Serverska aplikacija simulira rad videoteke iz koje korisnici (identificirani svojim korisničkim imenom) mogu posuditi DVD filma preko Interneta (DVD se fizički dostavlja poštom na kućnu adresu). Protokol za komunikaciju serverskog i klijentskog programa je definiran na sljedeći način:

- Zaglavlje poruke se sastoji od koda zahtjeva, korisničkog imena korisnika i vremena početka slanja zahtjeva.
- Mogući kodovi zahtjeva u poruci su 1 - posudba DVD-a, 2 - vraćanje DVD-a, 3 - posudba uspješno provedena, 4 - posudba neuspješno provedena, 5 - povrat DVD-a uspio.
- U donjim funkcijama **poruka** predstavlja cijelu poruku (header + tijelo) gdje su zaglavlje i tijelo, kao i njihove komponente, odijeljeni znakom razmaka. Naslov DVD-a je jedna riječ bez razmaka. Pretpostavite da korisnik u jednom trenutku može posuditi samo jedan DVD.

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

1. strukture `DVD` i `posudba` koje pamte potrebne informacije o broju DVD-ova koji se mogu posuditi te trenutno posuđenim DVD-ovima. Definirajte i pripadajuća polja koja pamte postojeće filmove i posudbe. (4 boda)
2. funkciju `void obradiPosudbu(int sock, char *poruka)` koja ažurira odgovarajuća polja strukture `DVD` i `posudba` na serveru te vraća poruku o (ne)uspješnoj provedenoj posudbi klijentu. Posudba je neuspjela ukoliko zatraženi DVD ne postoji u videoteci. Polje struktura `DVD`-ova je sortirano uzlazno po broju kopija određenog naslova dostupnog u videoteci. (7 bodova)
3. funkciju `void provjeriPovrat(int sock)` koja pita server je li DVD poslan od strane klijenta (poštom) stigao u videoteku. Funkcija iz tijela pristigle poruke koje je u formatu: "`OK:#dana`" ili "`NIJEOK:-1`" ili javlja da povrat još nije obavljen ili izračunava broj dana koji je DVD bio kod korisnika te iznos potencijalne zakasnine. Ukoliko je DVD bio u rukama korisnika više od 14 dana, korisnik plaća zakasninu od 15 lipa/dan. (6 bodova)

MREŽE RAČUNALA – 1. KOLOKVIJ – ZADATAK 4

(12 bodova)

1. (4 bodova) napišite funkcije

- `void posalji(int sock, char *buffer, int duljinaPoruke)`
- `void primi(int sock, char *buffer, int duljinaPoruke)`

Uputa: ovdje slobodno pretpostavite da je `buffer` već alociran i dovoljno velik da spremljuje cijelu primljenu poruku.

2. (8 bodova) Zadana je sljedeća struktura podataka koja reprezentira računalo u nekoj mreži računala:

```
struct Racunalo {
    char* naziv;
    int broj_spojenih;
    char **nazivi_spojenih;
};
```

Koristeći funkcije iz prve točke, implementirajte:

- `void posaljiRacunalo(int sock, struct Racunalo rac)`
- `void primiRacunalo(int sock, struct Racunalo *rac)`

Upute: pobrinite se da polja strukture `Racunalo` koja koristite budu adekvatne veličine (alocirajte potrebne podatke). Ovdje ne morate paziti na greške u slanju i primanju. Smijete slati i primiti podatke koji ne pišu u strukturi. Sva slanja i primanja realizirajte koristeći funkcije `posalji` i `primi` (čak i ako ih niste implementirali). Strukture šaljite element po element, a ne cijele odjednom.