

# Mreže računala – prvi kolokvij, 3. studenog 2010.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

1. Dvije aplikacije (Alfa i Beta) komuniciraju putem Interneta. Dolje možete vidjeti fragmente koda aplikacija Alfa i Beta koji odgovaraju dijelu komunikacije. Pretpostavljamo da se `int`-ovi uvijek mogu poslati, onosno primiti u jednom pokušaju.

```
Alfa

/* a je statički alocirano
   polje int-ova */
int sz = sizeof(a);
send(sck, &sz, sizeof(sz), 0);
for(i = 0; i < sz/sizeof(int); ++i){
    int x = htonl(a[i]);
    send(sck, &x, sizeof(x), 0);
}
```

```
Beta

int sz;
recv(sck, &sz, sizeof(sz), 0);
int* a = malloc(sz);
int recvd = 0;
while(recvd < sz){
    recvd += recv(sck, (char*)a + recvd,
                 sz - recvd, 0);
}
for(i = 0, i < sz/sizeof(int), ++i)
    a[i] = ntohl(a[i]);
```

- [1] (a) U kojem smjeru gornji fragmenti koda šalju podatke?
- A. Aplikacija Alfa podatke šalje aplikaciji Beta.
  - B. Aplikacija Beta podatke šalje aplikaciji Alfa.
  - C. Podaci idu u oba smjera.
  - D. Iz priloženih fragmenata koda nije moguće iščitati odgovor na ovo pitanje.
- [1] (b) Koja aplikacija ima ulogu servera, odnosno klijenta?
- A. Alfa je klijent, Beta je server.
  - B. Beta je klijent, Alfa je server.
  - C. Obje aplikacije su istovremeno klijent i server.
  - D. Iz priloženih fragmenata nije moguće odrediti tko ima ulogu klijenta, odnosno servera.
- [1] (c) Uz pretpostavku da neće biti grešaka u komunikaciji, hoće li svi podaci poslani u priloženim fragmentima koda biti uredno primljeni?
- A. Da.
  - B. Ne, jer svakom pozivu funkcije `send` mora odgovarati točno jedan poziv funkcije `recv`.
  - C. Ne, jer server nije obavijestio klijenta o količini podataka koje mu šalje.
  - D. Ne, jer klijent nije obavijestio servera o prekidu konekcije po završetku primanja podataka.
- [2] (d) Kako bi izgledao gornji fragment koda aplikacije Alfa u kojem bi se provjeravalo je li došlo do neočekivanog prekida konekcije i ostalih grešaka prilikom mrežne komunikacije?

## Mreže računala – prvi kolokvij, 3. studenog 2010.

Ime i prezime: \_\_\_\_\_ JMBAG: \_\_\_\_\_

- [5] 2. Dizajnirajte protokol (vrste porukâ, njihovo značenje i izgled) za prijenos datoteka između dva računala. Protokol mora podržavati sljedeće: nakon što se spoji na server, klijent može dobiti listu datoteka koje postoje na serveru, preuzeti ili postaviti jednu više datoteka sa servera, odnosno na server. Konekcija se prekida isključivo na zahtjev klijenta.

## Mreže računala – prvi kolokvij, 3. studenog 2010.

Ime i prezime: \_\_\_\_\_ JMBAG: \_\_\_\_\_

[5] 3. Zadana je struktura

```
typedef struct _node{
    char* str;
    struct _node* next;
} node;
```

pomoću koje je implementirana vezana lista stringova. Napišite funkciju

```
void send_list(int sock, node* start);
```

koja preko socketa `sock` šalje vezanu listu na čiji prvi element pokazuje pointer `start`. (Smijete pisati pomoćne funkcije.)

## Mreže računala – prvi kolokvij, 3. studenog 2010.

Ime i prezime: \_\_\_\_\_ JMBAG: \_\_\_\_\_

- [5] 4. Što je multipleksiranje? Objasnite razliku između multipleksiranja dijeljenjem frekvencije i multipleksiranja dijeljenjem vremena. Što je multipleksor? Koja vrsta multipleksiranja se pojavljuje kod tehnologije ADSL? Kakvo multipleksiranje imamo kod spajanja domova na Internet preko infrastrukture za kabelsku televiziju?

## Mreže računala – prvi kolokvij, 3. studenog 2010.

Ime i prezime: \_\_\_\_\_ JMBAG: \_\_\_\_\_

- [5] 5. Što je paketna sklopka? U kojoj vrsti mreža se ona pojavljuje? Opišite barem dvije strukture podataka koje se pohranjuju u paketnoj sklopki. Je li paketna sklopka sinonim za koncentrador (hub)? – Obrazložite.

# Mreže računala – prvi kolokvij, 3. studenog 2010.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

1. Dvije aplikacije (Az i Buky) komuniciraju putem Interneta. Dolje možete vidjeti fragmente koda aplikacija Az i Buky koji odgovaraju dijelu komunikacije. Pretpostavljamo da se `int`-ovi uvijek mogu poslati, onosno primiti u jednom pokušaju.

Az	Buky
<pre>/* a je statički alocirano    polje int-ova */ int sz = sizeof(a); int sent = 0; send(sck, &amp;sz, sizeof(sz), 0); for(i = 0, i &lt; sz/sizeof(int), ++i)     a[i] = htonl(a[i]); while(sent &lt; sz){     sent += send(sck, (char*)a+sent,                 sz - sent, 0); }</pre>	<pre>int sz; recv(sck, &amp;sz, sizeof(sz), 0); for(sz /= sizeof(int); sz &gt; 0; --sz){     int x;     recv(sck, &amp;x, sizeof(x), 0);     printf("%d ", ntohl(x)); }</pre>

- [1] (a) U kojem smjeru gornji fragmenti koda šalju podatke?
- A. Aplikacija Az podatke šalje aplikaciji Buky.
  - B. Aplikacija Buky podatke šalje aplikaciji Az.
  - C. Podaci idu u oba smjera.
  - D. Iz priloženih fragmenata koda nije moguće iščitati odgovor na ovo pitanje.
- [1] (b) Koja aplikacija ima ulogu servera, odnosno klijenta?
- A. Az je klijent, Buky je server.
  - B. Buky je klijent, Az je server.
  - C. Obje aplikacije su istovremeno klijent i server.
  - D. Iz priloženih fragmenata nije moguće odrediti tko ima ulogu klijenta, odnosno servera.
- [1] (c) Uz pretpostavku da neće biti grešaka u komunikaciji, hoće li svi podaci poslani u priloženim fragmentima koda biti uredno primljeni?
- A. Da.
  - B. Ne, jer svakom pozivu funkcije `send` mora odgovarati točno jedan poziv funkcije `recv`.
  - C. Ne, jer server nije obavijestio klijenta o količini podataka koje mu šalje.
  - D. Ne, jer klijent nije obavijestio servera o prekidu konekcije po završetku primanja podataka.
- [2] (d) Kako bi izgledao gornji fragment koda aplikacije Buky u kojem bi se provjeravalo je li došlo do neočekivanog prekida konekcije i ostalih grešaka prilikom mrežne komunikacije?

## Mreže računala – prvi kolokvij, 3. studenog 2010.

Ime i prezime: \_\_\_\_\_ JMBAG: \_\_\_\_\_

- [5] 2. Dizajnirajte protokol (vrste porukâ, njihovo značenje i izgled) za sljedeću situaciju: klijent se spaja na server koji ima mogućnost računanja raznih hash funkcija na podacima koji mu se pošalju. Klijent može serveru poslati proizvoljne podatke uz zahtjev da server izračuna odgovarajući hash primjenom jedne ili više hash funkcija. Server klijentu mora poslati odgovor iz kojeg je jasno koji hash je dobiven primjenom koje funkcije. Osim toga, klijent može od servera zatražiti i popis svih hash funkcija dostupnih serveru. Konekcija se prekida isključivo na zahtjev klijenta.

## Mreže računala – prvi kolokvij, 3. studenog 2010.

Ime i prezime: \_\_\_\_\_ JMBAG: \_\_\_\_\_

[5] 3. Zadana je struktura

```
typedef struct _node{
    int* niz;
    struct _node* next;
} node;
```

pomoću koje je implementirana vezana lista nizova cijelih brojeva. Napišite funkciju

```
void send_list(int sock, node* start);
```

koja preko socket-a `sock` šalje vezanu listu na čiji prvi element pokazuje pointer `start`. (Smijete pisati pomoćne funkcije. Smijete pretpostaviti da će `int`-ovi biti poslani u jednom pokušaju.)



## Mreže računala – prvi kolokvij, 3. studenog 2010.

Ime i prezime: \_\_\_\_\_ JMBAG: \_\_\_\_\_

- [5] 4. Objasnite razliku između mreže s prospajanjem paketa (packet switching) i mreže s prospajanjem linija (circuit switching). U koju od ovih dviju vrsta spada Ethernet? Navedite prednosti i mane upotrebe paketa. Koja je razlika između paketa i okvira?

## Mreže računala – prvi kolokvij, 3. studenog 2010.

Ime i prezime: \_\_\_\_\_ JMBAG: \_\_\_\_\_

- [5] 5. Objasnite kako se rasprostranjena mreža (WAN) modelira pomoću usmjerenog grafa. Kako se uz pomoć tog grafa rješava problem usmjerenja u WAN-u? Zašto je pritom važan Bellmanov princip optimalnosti? Što treba napraviti da se paketi u WAN-u usmjeravaju tako da na putu od polazišta do odredišta naprave što manje skokova sa sklopke na sklopku? Kako postići da se paketi usmjeravaju tako da put od polazišta do odredišta prelaze u što kraćem vremenu?