

JMBAG

IME & PREZIME

Mreže računala

Prvi kolokvij, 26. studenoga 2025. godine

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

Svaki zadatak rješavati na zasebnom papiru/papirima i predati odvojeno od ostalih!

PRVI ZADATAK

OSTVARENI BODOVI

	2+4+2
--	-------

Objasnite pojmove *mreža računala* i *protokol*.

Navedite prednosti i mane žičanih odnosno bežičnih medija te ih usporedite.

Što je nosač (eng. carrier)?

JMBAG

IME & PREZIME

DRUGI ZADATAK

OSTVARENI BODOVI

 13

Dovršite ispod prikazani kod servera. Server komunicira s klijentima na IP-adresi 161.53.8.14 i portu 65100. Za svakog klijenta za koji server prihvati konekciju, server određuje broj klijentovih alternativnih *host-nameova* te šalje klijentu taj broj. Nakon toga, server klijentu šalje koji je on ukupno po redu klijent kojem je prihvaćena konekcija. Primjerice, ako je neki klijent 2025. po redu klijent s kojim server komunicira (brojeći od trenutka pokretanja servera), tada tom klijentu treba poslati broj 2025. Zatim server prekida konekciju s klijentom.

Server ne treba moći istovremeno komunicirati s više klijenata, a najveći broj klijenata koji odjednom može čekati na uslugu servera je 5. Pripazite na moguće greške (u slučaju greške ispišite prikladnu poruku o grešci). Možete pretpostaviti da se svaki broj može poslati jednom *send* naredbom.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <malloc.h>

#include <sys/socket.h>
#include <sys/types.h>
#include <netdb.h>
#include <netinet/in.h>
#include <unistd.h>
#include <arpa/inet.h>

int main(void) {
    int listenerSocket = socket(PF_INET, SOCK_STREAM, 0);
    if(listenerSocket == -1)
        perror("socket");
```

TREĆI ZADATAK

OSTVARENI BODOVI

12

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

Potrebno je dizajnirati mrežnu aplikaciju koja omogućava igranje igre *Kviz*. Svaki igrač prije početka mora odabrati svoje ime od točno 8 znakova. Ime svakog igrača mora biti jedinstveno. Igrači koji su u igri podijeljeni su u 5 soba od kojih svaka ima svoju temu: *matematika*, *geografija*, *fizika*, *kemija*, *biologija*. Svaki igrač mora biti u nekoj sobi koju odabere, ali ju u bilo kojem trenutku može promijeniti. Svaki igrač koji se nalazi u pojedinoj sobi istovremeno dobiva isto pitanje vezano uz temu te sobe (primjerice, za sobu s temom *biologija* pitanje *Koliko pauk ima nogu?*). Samo igrač koji prvi da točan odgovor na pitanje ostvaruje jedan bod. Nakon toga svi igrači u toj sobi moraju vidjeti koji je bio točan odgovor i tko je dao taj odgovor. Igrač u bilo kojem trenutku može napustiti igru ili vidjeti koliko ima bodova i na kojem je mjestu u rang ljestvici svih trenutno aktivnih igrača po broju bodova.

- (a) Osmislite vrste poruka i njihov format (zaglavlje/*header* i tijelo/*payload*) koje razmjenjuju klijent i server. **6 bodova**
- (b) Navedite kakvi se sve tipovi grešaka mogu dogoditi u komunikaciji. **2 boda**
- (c) Odaberite primjer po jedne klijentske poruke za svaku vrstu poruke klijenta (primjeri u skladu s Vašim definicijama mogućih vrsta i formata poruka) te pripadne odgovore od strane servera. **4 boda**

ČETVRTI ZADATAK

OSTVARENI BODOVI

17

Tvrtka *Instrukcije Things* na kraju svakog mjeseca isplaćuje svojim zaposlenicima plaću prema broju sati instrukcija koje su taj mjesec odradili. U ovome zadatku promatramo mrežnu aplikaciju kojom zaposlenici nakon svake odradene instrukcije prijavljuju koliko sati su te instrukcije trajale (kako bi se na kraju mjeseca odredio njihov ukupan broj). Svakom zaposleniku prilikom zaposlenja dodijeljen je njegov jedinstven prirodan broj (strogo) manji od 1000 koji ga identificira. Osim tog broja, pamti se njegovo ime, prezime i područja za koje daje instrukcije, a svaki od tih podataka je string duljine najviše 20. Područja su: matematika, fizika, kemija, informatika i geografija. Svaki zaposlenik može davati instrukcije iz više područja. U komunikaciji, zaposlenik osim svog identifikatora koristi i svoju šifru kako bi potvrdio svoj identitet. Šifra se sastoji od točno 8 znakova i ne sadrži znak točku (' . '). Broj sati u ovome zadatku je tipa `int`.

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

- Zaglavlje poruke sastoji se od koda (koji određuje vrstu poruke) te jedinstvenog broja koji određuje zaposlenika.
- Vrste poruka uključuju sljedeće (u drugom stupcu je naveden kod; uočite da klijent u svojim porukama potvrđuje svoj identitet svojom lozinkom *lozinka*):

<i>DODAJ lozinka br_sati</i>	1	Korisnik želi evidentirati broj sati instrukcija koje je odradio.
<i>INFO lozinka</i>	2	Korisnik želi saznati koliko je do sad ukupno odradio sati.
<i>INFO_R br_sati</i>	3	Server šalje klijentu broj sati koje je on do sad ukupno odradio.
<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 komponente zaglavlja, odijeljeni znakom zareza (' , '). Sami odredite kako su odijeljene komponente tijela poruke. Funkcije vraćaju 0 ako je došlo do greške (inače 1).

1. Implementirajte **sve** strukture/polja/varijable koje trebaju serveru za pamćenje svih navedenih podataka o zaposlenicima. 4 boda
2. Implementirajte serversku funkciju `void evidentiraj(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 uspješnom ili neuspješnom (u kom slučaju *poruka* sadrži opis greške) evidentiranju broja sati održane instrukcije. Evidentiranje nije uspješno ako vrijedi barem jedno od sljedećeg: ne postoji korisnik s jedinstvenim brojem iz zaglavlja, lozinka tog korisnika nije ispravna, broj sati je negativan ili veći od 8 (tvrtka ne dopušta da jedne instrukcije traju više od 8 sati). 7 bodova
3. Implementirajte klijentsku funkciju `void informacije(int sock)` koja od korisnika učitava njegov jedinstveni broj i lozinku. Zatim funkcija pita server koliko taj korisnik ima ukupan broj sati odradenih instrukcija. Potrebno je korisniku ispisati podatke dobivene od servera (ili poruku o grešci ako se ona dogodila). 6 bodova