

---

Elektrotehnički fakultet u Beogradu  
Katedra za računarsku tehniku i informatiku

*Predmet:* Operativni sistemi 2  
*Nastavnik:* prof. dr Dragan Milićev  
*Odsek:* Računarska tehnika i informatika, Softversko inženjerstvo  
*Kolokvijum:* Prvi, januar 2023.  
*Datum:* 20. 1. 2023.

*Prvi kolokvijum iz Operativnih sistema 2*

*Kandidat:* \_\_\_\_\_

*Broj indeksa:* \_\_\_\_\_ *E-mail:* \_\_\_\_\_

*Kolokvijum traje 90 minuta. Dozvoljeno je korišćenje literature.*

*Zadatak 1* \_\_\_\_\_/10                      *Zadatak 3* \_\_\_\_\_/10  
*Zadatak 2* \_\_\_\_\_/10

**Ukupno:** \_\_\_\_\_/30 = \_\_\_\_\_%

**Napomena:** Ukoliko u zadatku nešto nije dovoljno precizno definisano, student treba da uvede razumnu pretpostavku, da je uokviri (da bi se lakše prepoznala prilikom ocenjivanja) i da nastavi da izgrađuje preostali deo svog odgovora na temeljima uvedene pretpostavke. Ocenjivanje unutar potpitanja je po sistemu "sve ili ništa", odnosno nema parcijalnih poena. Kod pitanja koja imaju ponuđene odgovore treba **samo zaokružiti** jedan odgovor. Na ostala pitanja odgovarati **čitko, kratko i precizno**.

---

## 1. (10 poena) Raspoređivanje procesa

Neki sistem za rad u realnom vremenu (*real-time*) primenjuje algoritam „prvo onaj sa najskorijim vremenskim rokom“ (*earliest-deadline-first, EDF*) raspoređivanja procesa. Red spremnih procesa i ovaj algoritam implementira klasa `EDFScheduler` čija je definicija data dole. Za ovaj sistem važi sledeće:

- Svi procesi su ili periodični (aktiviraju se kad dođe vreme periodične aktivacije, obave jednu aktivaciju i suspenduju se do sledeće periode) ili sporadični (aktiviraju se na spoljašnji sporadičan događaj, odrade aktivaciju i suspenduju se do sledećeg događaja).
- U svakoj aktivaciji svakog procesa, polje `deadline` u strukturi `PCB` postavljeno je na apsolutno vreme isteka vremenskog roka u toj aktivaciji. Ovo vreme je izraženo kao celobrojna vrednost u broju nanosekundi od početka izvršavanja sistema.
- Uvek se izvršava onaj proces koji ima najskoriji vremenski rok. Taj proces dobija se pozivom operacije `getRunning` klase `EDFScheduler`.
- Ovaj tekući proces se ne izbacuje iz reda spremnih sve dok se sam ne suspenduje (uspava) do svoje sledeće aktivacije. Tada sistem poziva operaciju `removeRunning` klase `EDFScheduler`.
- Proces se u red spremnih dodaje kada dođe trenutak njegove aktivacije (perioda ili događaj). Tada ga sistem stavlja u red spremnih pozivom operacije `add` ove klase i ponovo poziva operaciju `getRunning` da bi odredio tekući proces i uradio eventualno preotimanje.

Procesi su u redu spremnih ulančani u jednostruko ulančanu listu čija je glava u `EDFScheduler::head`, a preko polja `PCB::next`. Implementirati u potpunosti klasu `EDFScheduler`.

```
typedef unsigned long long Time;
struct PCB { PCB *next; Time deadline; ... };

class EDFScheduler {
public:
    EDFScheduler ();

    PCB* getRunning () const;
    void removeRunning ();
    void add (PCB*);

private:
    PCB* head;
};
```

Rešenje:

**2. (10 poena) Međuprocesna komunikacija pomoću deljene promenljive**

Na programskom jeziku Java napisati kod za monitor koji ima dve operacije, *flip* i *flop*, uz sledeću sinhronizaciju: između dva susedna izvršavanja operacije *flip* mora se najmanje  $N$  puta izvršiti operacija *flop*. Drugim rečima, nakon jednog izvršavanja *flip*, mora doći najmanje  $N$  (može i više) izvršavanja *flop*, pa onda opet može *flip* itd.

Rešenje:

### 3. (10 poena) Međuprocesna komunikacija razmenom poruka

Napisati kod klijenta koji komunicira sa više servera. Klijent treba da ima metodu:

```
void processDataFromServers();
```

koja dohvata stringove sa servera. String ne sadrži znak za prelazak u novi red. Komunikacija se sastoji od traženja stringa i primanja stringa. Svi stringovi su imenovani celim brojevima od 0 do `MAX_INDEX`, gde je `MAX_INDEX` celobrojna konstanta. Jedan server ne mora sadržati sve stringove, ali je sigurno da svaki string postoji bar na jednom serveru. Stringovi sa istim imenom su identični. Sve stringove sa imenima od 0 do `MAX_INDEX` treba dohvatiti sa servera, spojiti u jedan string (sortirane po imenu), i proslediti metodi:

```
void processData(String data);
```

koja je `data` i ne treba je realizovati. String sa jednim imenom treba dohvatiti tačno jednom.

U klasi klijenta postoji već inicijalizovan niz:

```
static String[] SERVERS = ...;
```

koji sadrži adrese dostupnih servera. Svi serveri pružaju identičnu uslugu. Svi serveri osluškiju na portu 5555.

Rešenje: