
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Operativni sistemi 2
Nastavnik: prof. dr Dragan Milićev
Odsek: Računarska tehnika i informatika
Kolokvijum: Prvi, decembar 2019.
Datum: 1. 12. 2019.

Prvi kolokvijum iz Operativnih sistema 2

Kandidat: _____

Broj indeksa: _____ *E-mail:* _____

Kolokvijum traje 1,5 sat. 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

U nekom sistemu za rad u realnom vremenu (*real-time system*) koristi se raspoređivanje „najskoriji rok - prvi“ (*Earliest Deadline First, EDF*) na jednom procesoru. U sistemu postoji četiri periodična procesa čije su karakteristike date u tabeli. U svakoj aktivaciji nekog periodičnog procesa, taj proces zahteva izvršavanje na procesoru dužine koja je vrlo malo manja od vrednosti date u tabeli. Vremenski rok (*deadline*) svakog procesa jednak je njegovoj periodi ($D = T$), što znači da se u svakoj periodičnoj aktivaciji svakog procesa izvršavanje procesa u toj aktivaciji mora završiti najkasnije do trenutka sledeće periodične aktivacije tog istog procesa. U trenutku $t = 0$ poklopili su se trenuci periodične aktivacije sva četiri procesa. Sistem je sa preotimanjem (*preemptive*), a trajanje reakcije na periodičnu aktivaciju periodičnog procesa na prekid od tajmera, kao i trajanje promene konteksta je zanemarljivo (mnogo kraće od 1 ms).

Proces	Perioda [ms]	Dužina izvršavanja [ms]
A	40	10
B	80	10
C	60	10
D	160	50

Navesti koji se od ovih procesa izvršava u sledećim trenucima (ako u navedenom trenutku nijedan proces nije spreman i ne izvršava se, napisati „nijedan“):

Trenutak [ms]	Izvršava se proces
15	
25	
35	
55	
65	
75	
95	
110	
125	

2. (10 poena) Međuprocesna komunikacija pomoću deljene promenljive

Za problem filozofa koji večeraju (*dining philosophers*) napraviti monitor sa klasičnim uslovnim promenljivim koji ima dve procedure, *startEating* i *stopEating*, a koje poziva svaki proces koji modeluje ponašanje jednog filozofa pre i nakon jela, respektivno. Svaki filozof treba da uzme obe svoje viljuške zajedno (tj. da zauzme ili obe viljuške, ili nijednu), što znači da filozof treba da čeka (tj. da se proces koji modeluje ponašanje filozofa suspenduje) ako i samo ako je neka od njegovih viljušaka zauzeta i ako se suspenduje, ne treba da drži zauzetu bilo koju viljušku, tako da rešenje nema problem potencijalne mrtve blokade (*deadlock*), ali može ima problem potencijalnog izgladnjivanja (*starvation*).

Rešenje:

3. (10 poena) Međuprocesna komunikacija razmenom poruka

Implementirati veb servis na programskom jeziku Java, koji će krajnjem korisniku omogućiti da kupuje sadržaj. Svaki korisnik se indentifikuje korisničkim imenom i može u jednoj sesiji tražiti više usluga uzastopno. Prilikom prvog prijavljivanja na sistem korisnik dobija na računu N jedinica virtuelnog novca, gde je N globalna konstanta. Kada korisnik iskoristi neku uslugu sa računa mu se oduzima onoliko jedinica virtuelnog novca kolika je cena usluge. Na raspolaganju je sledeća klasa:

```
public class ContentPrice {
    public int getPrice(String contentName) {...}
}
```

Klasa `ContentPrice` omogućava dohvatanje cene usluge. Server treba da obrađuje zahteve korisnika konkurentno. Samo pružanje usluge nije potrebno realizovati, samo navesti komentar gde bi bilo. Korisnik može da kupuje usluge sve dok ima virtuelnog novca na računu.

Međuprocesnu komunikaciju realizovati preko priključnica (*socket*) i razmenom poruka (*message passing*). Dozvoljeno je korišćenje koda prikazanog na vežbama (kod sa vežbi ne treba prepisivati, nego npr. reći koja klasa ili koji metod se koriste i/ili menjaju, nasleđuju, ...). Napisati samo kod servera kao rešenje zadatka. Server prihvata zahteve klijenata na portu 5555.