
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Programiranje u realnom vremenu

Nastavnik: Prof. dr Dragan Milićev

Asistent: Dipl. ing. Bojan Furlan

Ispitni rok: Jun 2009.

Datum: 15.6.2009.

Kandidat: _____

Broj Indeksa: _____ *E-mail:* _____

Ispit traje 3 sata. Dozvoljeno je korišćenje literature.

<i>Zadatak 1</i>	_____ /10	<i>Zadatak 4</i>	_____ /20
<i>Zadatak 2</i>	_____ /10	<i>Zadatak 5</i>	_____ /10
<i>Zadatak 3</i>	_____ /20	<i>Projekat</i>	_____ /40

Ukupno na ispitu: _____ /70 *Ukupno na projektu:* _____ /40

Ukupno: _____ /110

Ocena: _____ (_____)

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) Tolerancija otkaza

Posmatra se sistem koji pouzdano vrši prenos matrice određenih dimenzija. Potrebno je implementirati funkciju `void correctMatrix(int** matrix, int n)` koja vrši proveru i korekciju zadate matrice veličine $n \times n$. Detekcija greške pri prenosu i korekcija se vrše tehnikom oporavka od greške unapred (*Forward Error Recovery*) pomoću redundantnih vektora i dvodimenzionalne kontrolne sume (*checksum*). Vektor kontrolne sume za redove nalazi se u n -toj koloni, pri čemu njegov svaki element predstavlja sumu elemenata iz istog reda, a za kolone u n -tom redu gde je svaki element suma elemenata iz date kolone. 2D kontrolna suma može ispraviti jednu i detektovati više grešaka. Ispravka pogrešno primljenog podatka se vrši na osnovu redundantnih vektora, tako što se detektuje pozicija na kojoj je došlo do greške pri prenosu. Na primer, za datu matricu dimenzija 2×2 :

1	2
3	4

matrica koja se prenosi će biti veličine 3×3 , gde poslednji red i poslednja kolona predstavljaju vektor kontrolne sume:

1	2	3
3	4	7
4	6	10

U slučaju pogrešno primljenog elementa na poziciji (1,1), npr. vrednosti 9:

9	2	3
3	4	7
4	6	10

na osnovu nekonzistentnih vrednosti sume prvog reda i prve kolone detektovaće se pozicija na kojoj je došlo do greške pri prenosu, a ispravljena vrednost će biti razlika kontrolne sume i sume ostalih uspešno primljenih elemenata (u ovom slučaju $3-2=1$ i $4-3=1$). U slučaju nemogućnosti oporavka od greške potrebno je da implementirana metoda baci izuzetak tipa `ErrorCorrectionException`.

Na raspolaganju su sledeće bibliotečne funkcije:

`int sumRow(int** matrix, int startRow, int startColumn, int lenght)`-Sumira red matrice određene dužine od zadate pozicije.

`int sumColumn(int** matrix, int startRow, int startColumn, int lenght)`-Sumira kolonu matrice određene dužine od zadate pozicije.

Rešenje:

2. (10 poena) Konkurentno programiranje

Na jeziku Ada dati rešenje problema filozofa koji večeraju (*dining philosophers*) koje nema problem izgladnjivanja (*starvation*), živog (*lovelock*), ni mrtvog blokiranja (*deadlock*). Filozofe predstaviti procesima, a sinhronizaciju obezbediti pomoću jednog serverskog procesa koji komunicira sa filozofima mehanizmom randevua.

Rešenje:

3. (20 poena) Realno vreme

U nekom sistemu za nadzor nekog osetljivog hemijskog procesa jedna kritična fizička veličina prati se senzorom koji je na računar povezan preko A/D konvertora čije je maksimalno vreme konverzije $t_{ad} = 40$ ms. Konvertor može da izmeri ovu veličinu samo ako je ona u određenom opsegu. Zbog toga senzor postavlja vrednost jednog svog registra kada ova veličina uđe u dati merni opseg, što predstavlja diskretan događaj u sistemu. Od tog trenutka, potrebno je u roku od $t_c = 200$ ms očitati praćenu veličinu i njenu vrednost ispisati na izlaz. Program za nadzor i upravljanje ovog sistema implementiran je kao jedan periodični proces koji je stalno aktivan.

Na raspolaganju su sledeće funkcije interfejsa prema uređajima:

<code>isMeasurable() : Boolean</code>	Vraća True ako je vrednost u mernom opsegu
<code>startAD()</code>	Pokreće A/D konverziju
<code>readAD() : Real</code>	Očitava A/D konvertor.

(a)(10) Na jeziku Ada implementirati ovaj proces tako da ne pati od kumulativnog plivanja (engl. *cumulative drift*) i ima zaštitu od prekoračenja vremenskog roka, uzimajući vremenske parametre kao simboličke konstante.

(b)(10) Napisati i objasniti nejednakosti koje predstavljaju uslove za periodu (T) i vremenski rok (D) ovog procesa u funkciji t_{ad} i t_c , a potom odrediti ove parametre (T i D).

Rešenje:

4. (20 poena) Raspoređivanje i rasporedivost

(a)(10) Dat je sledeći skup nezavisnih periodičnih procesa sa $D = T$ koji se raspoređuju po EDF:

Proces	T	C
a	6	1
b	8	2
c	12	4
d	24	2

Nacrtati vremenski dijagram raspoređivanja ovog skupa procesa počev od kritičnog trenutka i ispitati njihovu rasporedivost. Ukoliko u datom trenutku više procesa ima isti vremenski rok, prednost ima onaj proces koji se do tada izvršavao.

(b)(10) Data su tri periodična, nezavisna procesa sa sledećim karakteristikama:

Proces	T	D	C
a	4	3	1
b	6	2	2
c	18	10	5

Procesima dodeliti prioritete po DMPO i na vremenskom dijagramu prikazati kako se ovi procesi raspoređuju po FPS šemi počev od kritičnog trenutka.

Rešenje:

5. (10 poena) ROOM

Na jeziku ROOM realizovati klasu aktera koji predstavlja ograničeni bafer (*bounded buffer*) kapaciteta N koji prihvata celobrojne vrednosti. Akter poseduje *dataIn* i *dataOut* portove koji, pored potrebnih signala, imaju i signale *full* i *empty* respektivno, kojima se signalizira da je bafer pun odnosno prazan. Takođe, realizovati jednu klasu aktera koja simulira proizvođača (*producer*), kao i jednu klasu aktera koja simulira potrošača (*consumer*).

Rešenje: