
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Programiranje u realnom vremenu

Nastavnik: Prof. dr Dragan Milićev

Asistent: Bojan Furlan

Ispitni rok: Jun 2013.

Datum: 4.6.2013.

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

Na raspolaganju je bibliotečna funkcija

```
void assert(bool);
```

koja služi za detekciju greške proverom logičke ispravnosti. Ona prima argument logičkog tipa. Ako je vrednost argumenta `true`, funkcija prosto vraća kontrolu pozivaocu, inače signalizira postojanje greške podsystemu za oporavak od otkaza.

Ugraditi pozive funkcije `assert()` na potrebna mesta u kodu sledeće funkcije, kako bi taj kod bio što pouzdaniji, odnosno otporniji na otkaze:

```
void insert (int* a, int size_of_a, int* len_of_a, int x, int at) {  
    for (int i=*len_of_a-1; i>=at; i--) a[i+1]=a[i];  
    a[at]=x;  
    (*len_of_a)++;  
}
```

Rešenje:

2. (10 poena) Konkurentno programiranje

Na programskom jeziku Aada, korišćenjem mehanizma randevua, napisati program koji obezbeđuje sinhronizaciju prolaska vozila kroz pametnu raskrsnicu. Vozilima je na raskrsnici zabranjeno skretanje, tako da je potrebno obezbediti samo da se kroz raskrsnicu u svakom trenutku vozila kreću samo u jednom od dva moguća pravca. Vozila dolaze sa severa, juga, istoka i zapada, a maksimalan broj vozila koj se može nalaziti unutar raskrsnice je N . Napisati i kod procesa koji predstavljaju vozila. Ne treba rešavati potencijalni problem izgladnjivanja, već samo maksimizovati protok vozila kroz raskrsnicu.

Rešenje:

3. (20 poena) Realno vreme

(a)(10) Na jeziku Ada napisati kod za periodičan proces bez kumulativnog plivanja (*cumulative drift*) koji u svakoj svojoj aktivaciji treba da pošalje praznu poruku serverskom procesu pozivom njegovog ulaza `Server.send()`, a potom da sačeka na odgovor od servera na svom ulazu `receive()`, s tim da čekanje na odgovor bude vremenski ograničeno na 200 ms radi zaštite od izgubljenih poruka.

(b)(10) Na raspolaganju je bibliotečna funkcija:

```
Time clock();
```

koja očitava sat realnog vremena i vraća broj milisekundi proteklih od početka izvršavanja programa. Korišćenjem samo ove funkcije, realizovati apstrakciju „štoperice“ sa sledećim interfejsom:

```
class Timer {
public:
    Timer ();
    void start();
    void stop();
    void reset();
    Time elapsed();
};
```

Nakon kreiranja, štoperica stoji i ne meri proteklo vreme. Štoperica (tj. merenje vremena pomoću nje) se može pokrenuti (`start()`), privremeno zaustaviti (`stop()`), ponovo pokrenuti (`start()`) itd. proizvoljno mnogo puta, tako da štoperica meri kumulativno proteklo vreme za koje je bila pokrenuta. Nakon zaustavljanja, ukupno izmereno proteklo vreme može se očitati (`elapsed()`), ili se štoperica može „resetovati“ (`reset()`).

Rešenje:

4. (20 poena) Raspoređivanje i rasporedivost

Data su tri procesa sledećih karakteristika:

<i>Proces</i>	<i>Trenutak aktivacije</i>	<i>Prioritet</i>	<i>Izvršavanje</i>
<i>a</i>	2	3 (najviši)	ECE
<i>b</i>	3	2	EE
<i>c</i>	0	1	ECCCE

Svako slovo u opisu izvršavanja predstavlja određenu vrstu izvršavanja tokom jedne jedinice vremena:

E – sopstveno izračunavanje bez interakcije sa drugim procesima;

C – boravak u međusobno isključivoj zajedničkoj kritičnoj sekciji.

Nacrtati vremenski dijagram raspoređivanja ovih procesa za jednu aktivaciju ako je algoritam raspoređivanja:

a)(10) Obični FPS, bez nasleđivanja prioriteta (*priority inheritance*);

b)(10) FPS sa nasleđivanjem prioriteta (*priority inheritance*).

Rešenje:

5. (10 poena) ROOM

Na jeziku ROOM realizovati klasu aktera koji obavlja periodično očitavanje A/D konvertora tehnikom pomeranja periode (*period displacement*) i svaku očitanu vrednost šalje na svoj izlazni port. Sa drugog svog porta ovakav akter može da prima signale za uključivanje, isključivanje i promenu periode očitavanja na novu zadatu vrednost. Na raspolaganju su funkcije za pokretanje konverzije `startAD()` i očitavanje konvertovane vrednosti `readAD()` koje se mogu koristiti na nivou detalja.

Rešenje: