
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Programiranje u realnom vremenu
Nastavnik: Prof. dr Dragan Milićev
Asistent: dipl. ing. Nemanja Kojić
Ispitni rok: Jun 2010.
Datum: 17.06.2010.

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
<i>Ukupno na ispitu:</i>	_____ /70	<i>Ukupno na projektu:</i>	_____ /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

U nekom industrijskom sistemu visoke pouzdanosti postoje tri redundantna senzora koja mere istu fizičku veličinu. Ove senzore kontrolišu tri serverska procesa s_1 , s_2 i s_3 čiji je interfejs na jeziku Ada definisan na sledeći način:

```
task type Sensor is
  entry read (out value:Real);
  entry reset;
end Sensor;
```

```
s1, s2, s3 : Sensor;
```

Potrebno je realizovati drajverski proces `SensorDriver` čiji je interfejs

```
task SensorDriver is
  entry read (out value:Real);
end SensorDriver;
```

a koji većinskom logikom, uz neegzaktno poređenje sa tolerancijom razlike očitavanja unutar margine `errorMargin`, očitava ove senzore i daje rezultat pozivaocu usluge `read`, pri čemu resetuje onaj senzorski proces koji je dao očitavanje različito od ostala dva, ukoliko je to slučaj. Ukoliko nije moguće izvršiti očitavanje većinskom logikom, treba resetovati sva tri procesa i ponoviti očitavanje.

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 $t_{ad} = 40\text{ms}$. 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 $t_c = 160\text{ms}$ izmeriti ovu veličinu i njenu vrednost ispisati na izlaz. Softver ovog kontrolnog 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

Data su četiri procesa sledećih karakteristika:

<i>Proces</i>	<i>Trenutak aktivacije</i>	<i>D</i>	<i>C</i>	<i>Prioritet za FPS</i>
<i>a</i>	4	10	4	4 (najviši)
<i>b</i>	2	14	6	3
<i>c</i>	2	8	3	2
<i>d</i>	0	8	4	1

Vrednost za vremenski rok D dat je relativno u odnosu na trenutak aktivacije.

Nacrtati vremenski dijagram raspoređivanja ovih procesa za interval od 0 do 18 ako je algoritam raspoređivanja:

a)(10) FPS, sa raspodelom prioriteta koja je data u poslednjoj koloni gornje tabele;

b)(10) EDF.

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 i očitavanje konvertovane vrednosti koje se mogu koristiti na nivou detalja.

Rešenje: