

---

---

Elektrotehnički fakultet u Beogradu  
Katedra za računarsku tehniku i informatiku

*Predmet:* Programiranje u realnom vremenu

*Nastavnik:* Prof. dr Dragan Milićev

*Ispitni rok:* Jun 2008.

*Datum:* 9.7.2008.

*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

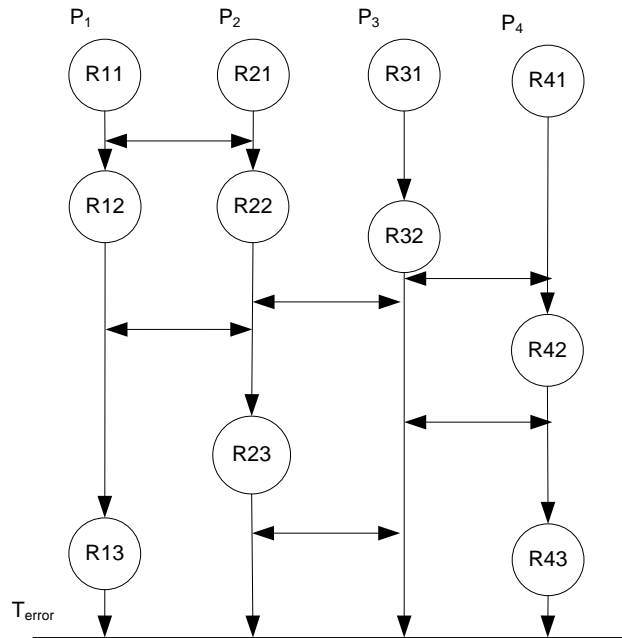
**Ocena:** \_\_\_\_\_ ( \_\_\_\_\_ )

**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 slici je šematski prikazano izvršavanje četiri uporedna procesa  $P_1$  do  $P_4$  sa svojim tačkama oporavka (engl. *recovery points*)  $R_{ij}$  i međusobnom komunikacijom. Objasniti šta se dešava u slučaju nastanka greške u naznačenom trenutku  $T_{error}$ , ako je greška nastala u jednom od ta četiri procesa. Odgovor dati za svaki pojedinačni slučaj od ta četiri, popunjavanjem date tabele, upisujući u nju tačke oporavka u koju treba vratiti procese.



		Proces koji treba vratiti u tačku oporavka			
Otkaz nastao u procesu		P1	P2	P3	P4
	Slučaj 1: P1				
	Slučaj 2: P2				
	Slučaj 3: P3				
	Slučaj 4: P4				

**2. (10 poena) Konkurentno programiranje**

Korišćenjem koncepata procesa i randevua u jeziku Ada, implementirati sistem koji se sastoji od proizvođača, potrošača i ograničenog bafera.

### 3. (20 poena) Realno vreme

U nekom brzom vozu postoji sistem za nadzor ispravnosti kočionog sistema koji aktivira rezervni, sigurnosni kočioni sistem u slučaju otkaza glavnog kočionog sistema. Ovaj kontrolni sistem treba da proveriti da li po aktiviranju glavne kočnice voz počinje da usporava u roku od  $t_d = 0.5$  s od trenutka aktiviranja kočnice. Ovo kašnjenje od  $t_d$  podešeno je prema inerciji voza i osetljivosti senzora ubrzanja/usporenja voza, što znači da rezervnu kočnicu ne treba aktivirati u tom roku čak i ako se ne detektuje usporenje voza. Ako detektuje da voz ne usporava nakon isteka tog roka, ovaj sistem treba da aktivira pomoćni kočioni sistem u roku od dodatnih  $t_c = 0.2$  s nakon tog roka  $t_d$ .

Softver ovog kontrolnog sistema implementiran je kao jedan periodični proces koji je stalno aktivan i koji radi na sledeći način. Ako u nekoj aktivaciji prvi put pronade da je glavna kočnica aktivirana, ne radi ništa u toj i u još nekoliko narednih aktivacija (ukupno njih  $n$ ), kako bi obezbedio protok vremena  $t_d$ , a onda u  $n+1$ -oj aktivaciji, ukoliko pronade da voz i dalje ne usporava a glavna kočnica je još uvek aktivirana, aktivira rezervnu kočnicu.

Na raspolaganju su sledeće funkcije interfejsa prema uređajima:

<code>isMainBreakOn()</code> : Boolean	Vraća True ako je glavna kočnica aktivirana
<code>isDecelerating()</code> : Boolean	Vraća True ako je detektovano usporenje voza
<code>emergencyBreakOn()</code>	Uključuje pomoćnu kočnicu.

(a)(10) Na jeziku Ada implementirati ovaj proces tako da ne pati od kumulativnog plivanja (engl. *cumulative drift*), uzimajući periodu kao simboličku konstantu.

(b)(10) Napisati i objasniti nejednakosti koje predstavljaju uslove za periodu ( $T$ ), vremenski rok ( $D$ ) i broj aktivacija  $n$  u funkciji  $t_d$  i  $t_c$ , a potom odrediti ove parametre ( $T$ ,  $D$  i  $n$ ).

**4. (20 poena) Raspoređivanje i rasporedivost**

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

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

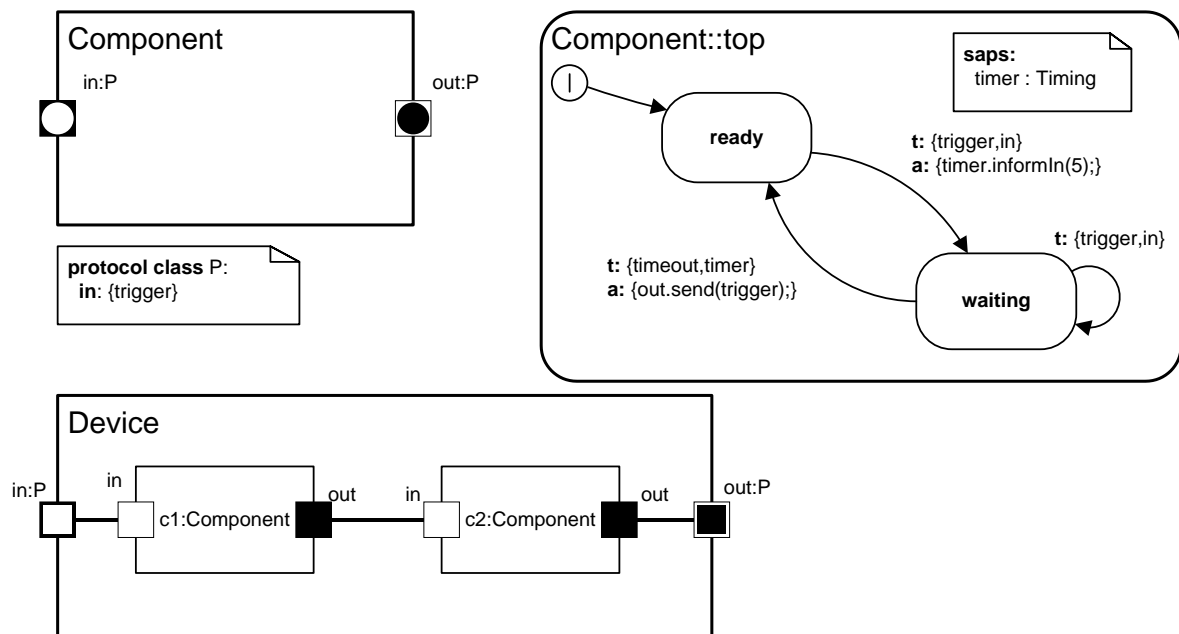
Nacrtati vremenski dijagram raspoređivanja ovog skupa procesa počev od kritičnog trenutka i ispitati njihovu rasporedivost.

(b)(10) Implementirati apstrakciju resursa sa operacijama zauzimanja i oslobađanja po protokolu ICPP na jednoprocesorskom sistemu, pretpostavljajući odgovarajući interfejs apstrakcije procesa.

## 5. (10 poena) ROOM

Na slici je prikazan jedan jednostavan ROOM model. Pretpostavlja se da je trajanje „prelaznog režima“ jednog pojedinačnog aktera između dva „stabilna stanja“, tj. svi prelazi pokrenuti jednim trigerom, uključujući njihove akcije, transfer poruka unutar tih akcija i sve ostale obrade dok taj akter ne pređe u odredišno stanje i bude spreman za obradu eventualne naredne poruke (dakle, jedna „run to completion“ obrada), traje mnogo manje od jedne jedinice vremena kojom se izražava vreme u sistemskim uslugama, kao i da sistem odmah obrađuje postojeće pristigle poruke aktera, čim je u mogućnosti, tj. ne unosi neodređena i nepotrebna kašnjenja, odnosno „bezrazložna“ odlaganja; drugim rečima, svaki akter obrađuje novu pristiglu poruku čim završi tranziciju pokrenutu prethodnom.

U režimu izvršavanja kreirana je jedna instanca d klase `Device` i na njen port `in` „upumpane“ su spolja poruke sa signalom `trigger` u sledećim trenucima vremena: 0, 2, 4, 6 i 8. Odrediti šta će se dešavati na portu `out` aktera `d` (navesti signale poruka i trenutke u kojima se one pojavljuju na ovom portu).



Odgovor: \_\_\_\_\_