Drugi međuispit iz predmeta ***Sustavi za rad u stvarnom vremenu***

11. 5. 2010.

1. (2 boda) Što pokazuje/definira formula: $lub(U) = m\left(2^{\frac{1}{m}} - 1\right)$? Uz koje uvjete?

Formula definira najveću gornju granicu procesorske iskoristivosti za sustav od m zadataka uz koju će sustav sigurno biti rasporediv, ako se raspoređivanje radi RMPA metodom.

2. (2 boda) *Opći kriterij raspoređivanja* zadan je s dvije nejednakosti (formule *(i)* i *(ii)*). Što svaka od nejednakosti zasebno provjerava?

$$(i) \quad D_i + \sum_{j=1}^{i-1}\left(\left\lceil \frac{T_i}{T_j} \right\rceil - \left\lceil \frac{D_i}{T_j} \right\rceil\right) c_j \geq T_i$$

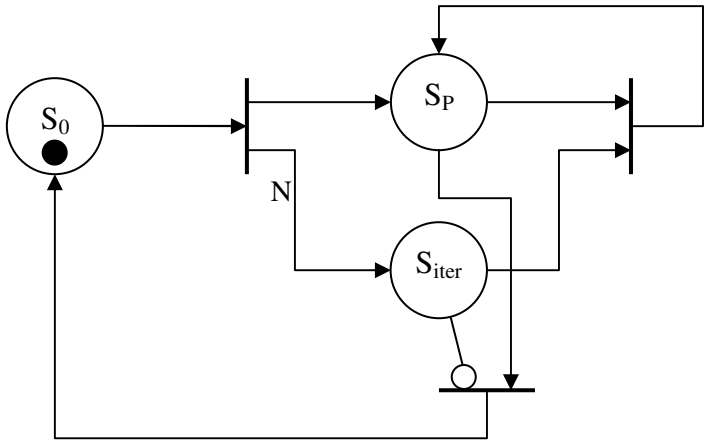
$$(ii) \quad \sum_{j=1}^{i}\left\lceil \frac{D_i}{T_j} \right\rceil c_j \leq D_i$$

(i) nakon Di nema mjesta za zadatak "i", svo se vrijeme "potroši" na zadatke većeg prioriteta
(ii) prije Di ima vremena za obavljanje svih zadataka većeg prioriteta, i to onoliko puta koliko se puta javljaju u tom intervalu, kao i za sam zadatak "i"

3. (2 boda) Zašto je *problem inverzije prioriteta* „problem“ za RT sustave?

Problem je stoga što se zadatak većeg prioriteta ("bitniji zadatak") odgađa zbog zadataka manjeg prioriteta, i to ne samo zadatka koji trenutno ima traženo sredstvo, već i ostalih koji i ne moraju imati nikakve veze s tim sredstvom na kome se zadatak većeg prioriteta blokira.

4. (2 boda) Dio svog posla ciklički zadatak obavlja u petlji s *N* prolaza (izvođenje petlje se ponavlja!). Nacrtati Petri-mrežu koja to prikazuje. Koristiti slijedeće oznake: početno stanje (prije i poslije petlje) je S_0, a stanje u petlji S_P. Po potrebi dodati i ostala stanja i prijelaze.



5. (2 boda) Grafičkim postupkom provjeriti je li zadani sustav zadataka rasporediv:
   T_1 = 2 ms, c_1 = 1 ms
   T_2 = 3 ms, c_2 = 1 ms
   T_3 = 12 ms, c_3 = 2 ms .

Rasporediv (iz grafičkog rješenja bi se to vidjelo).

6. (4 boda) *Općim kriterijem raspoređivanja* provjeriti je li zadani sustav zadataka rasporediv:
   $T_1 = 10$ ms, $c_1 = 5$ ms
   $T_2 = 15$ ms, $c_2 = 5$ ms
   $T_3 = 20$ ms, $c_3 = 1$ ms
   $T_4 = 30$ ms, $c_4 = 1$ ms.

Nije rasporediv! τ_3 nije rasporediv u kritičnom slučaju, formula (ii) to i pokazuje za sve točke raspoređivanja, uključujući i zadnju za 20 ms.

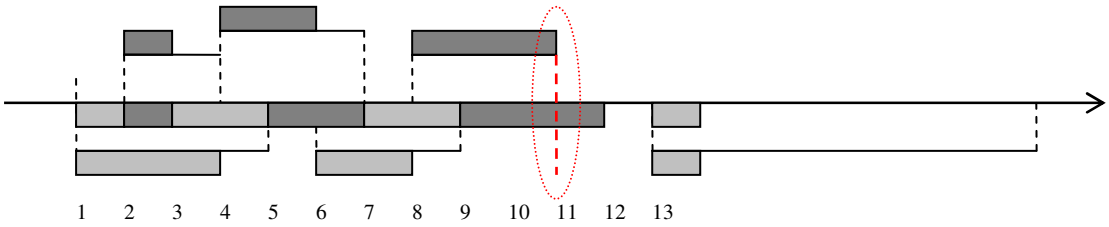
7. (3 boda) U sustavu koji koristi *DDS* (raspoređivanje prema krajnjim trenucima završetka) pojavljuju se događaji za koja su poznata i vremena izračunavanja, kao i krajnji trenuci završetka:

   (*vrijeme pojave*, *potrebno procesorsko vrijeme*, *krajnji trenutak dovršetka*) = (t_i, c_i, k_i) =

   { (1, 3, 5), (2, 1, 4), (4, 2, 7), (6, 2, 9), (8, 3, 11), (13, 1, 20) }.

   Provjeriti je li slijed događaja rasporediv *DDS* postupkom.

Nije rasporediv, događaj (8, 3, 11) ne stigne se obaviti do 11. jedinice vremena:



8. (3 boda) Dvije dretve obavljaju kôd prema slici. Navedite nekoliko problema zadanog kôda i kako se (bar neki od njih) mogu riješiti u ovom primjeru (predložite izmjene)? Namjera korištenih sinkronizacijskih mehanizama je osiguravanje ispravnih vrijednosti varijabli pri čitanju i pisanju, a zbog mogućeg paralelnog rada.

```
v1 = 0;
v2 = 0;
br[1] = 0;
br[2] = 0;

povećaj(i)
{
   br[i]++;
}

reset(n)
{
   čekajBSem(n);
   v1 = v2 = 0;
   PostaviBSem(n);
}
```

```
Dretva 1
{
   petlja {

      //radi nešto ...

      čekajBSem(k1);
      v1++;
      ako v1 > 10 && v1 > v2 {
         povećaj(1);
         reset(k2);
      }
      PostaviBSem(k1);

      //radi nešto ...
   }
}
```

```
Dretva 2
{
   petlja {

      //radi nešto ...

      čekajBSem(k2);
      v2++;
      ako v2 > 10 && v2 > v1 {
         povećaj(2);
         reset(k1);
      }
      PostaviBSem(k2);

      //radi nešto ...
   }
}
```

Problemi:

- *u dretvi 1, v2 se provjerava bez zaštite ("v1 > v2"); ekvivalentno s dretvom 2 za v1*
- *pozivom "reset" zaključava se i onaj drugi semafor, tj. pokušava se zaključati - može se pojaviti problem potpunog zastoja*

Rješenje:

- *(najjednostavnije) koristiti samo jedan binarni semafor ili monitor ili drugi mehanizam međusobnog isključivanja nad cijelim segmentom: od povećavanja do reseta;*