Tosiaikajärjestelmät Kevät 2006

Harjoitukset 3/7 (Ke 29.3.2006)

1) Järjestelmä käyttää sporadisten töiden ajoittamiseen cyclic EDF -algoritmia. Kehyksen koko on 5 aikayksikköä ja hyperperiodissa on 6 kehystä. Ajoitustaulun perusteella lasketut vapaat ajat kehyksissä ovat 1, 0.5, 0.5, 0.5, 1 ja 1.
   a) Oletetaan, että sporadinen työ S_1 (23,1) saapuu kehyksen 1 aikana ja sporadiset työt S_2 (16, 0.8) ja S_3 (20, 0.5) kehyksen 2 aikana. Mihin kehyksiin saapuneet työt ajoitetaan?
   b) Oletetaan, että jaksoton (aperiodic) työ, jonka suoritusaika on 3, saapuu ajanhetkellä 1. Milloin se saadaan valmiiksi, jos vuorottaja ei käytä 'slack stealing' menettelyä?

(Liu 5.2)

2) Järjestelmässä on kolme jaksollista tehtävää (2.5, 1), (4, 0.5) ja (5, 0.75), joien yhteenlaskettu käyttöaste on 0.475.
   a) Järjestelmässä on myös jaksollinen palvelin S(2, 0.5). Järjestelmä käyttää rate-monotonic vuorottajaa.
      i) Oletetaan, että jaksollinen palvelin on sporadinen. Mitkä ovat seuraavien kahden jaksottoman työn vasteajat? Ensimmäinen saapuu ajanhetkellä 3 ja sen suoritusaika on 0.75, toinen saapuu hetkellä 7.5 ja suoritusaika on 0.6.
      ii) Oletetaan, että jaksollinen palvelin onkin osa-aika palvelin (deferrable server). Mitkä silloin ovat noiden jaksottomien töiden vasteajat?
   b) Huomaa, että jaksollisen palvelimen käyttöaste on 0.25. Voimme vaihtaa palvelimen jakson pituutta kunhan pidämme käyttöasteen kiinteästi arvossa 0.25. Miten kohtien i) ja ii) vasteajat muuttuvat, jos jaksollisen palvelimen jakson pituudeksi muutetaan 1.
   c) Voidaanko vasteaikoja parantaa pidentämällä palvelimen jakson pituutta?

(Liu 7.1)

3) Tee tehtävä 2 käyttämällä vuorottajana earliest-deadline-first algoritmia?

4) *Vanha koetehtävä*: Onko seuraavassa taulukossa esitetty tehtäväjoukko ajoitettavissa
   a) RM (Rate Monotonic) tai
   b) EDF (Earliest Deadline First) menetelmällä?
   c) Voidaanko tehtäväjoukkoon vielä lisätä joku työ? Millainen?

   Perustele käyttäen aiemmin opittuja menetelmiä (käyttöaste, ajoitettavuusanalyysi ja simulointi). Jos tehtäväjoukko ei ole a) tai b) kohdissa ajoitettavissa, niin miten sitä pitäisi muuttaa.

Tehtävä	***Saapumisaika***	***Laskennan kesto***	***Jakson pituus***
A	0	2	10
B	0	8	15
C	5	1	20
D	10	3	25
E	15	2	30

ESSEE:

Palauta kirjallisena. Esseistä saa aitoja lisäpisteitä. Jokainen kelvollinen ja viimeistään harjoituksissa palautettu essee on yhden pisteen arvoinen.

Esseeksi käy yhden (tai kahden) sivun mittainen suorasanainen teksti, joka käsittelee artikkelin aihepiiriä.

Tällä kertaa voit valita yhden seuraavista kolmesta artikkelista ja kirjoittaa esseen kyseisen artikkelin herättämien ajatusten perusteella. Yritä olla kirjoittamatta referaattia. Kirjoita mieluummin vaikka oppimispäiväkirjatyyppinen teksti siitä, mikä artikkelissa tai sen aihepiirissä teki vaikutuksen sinuun.

Valitse vain yksi seuraavista artikkeleista:

Giotto-artikkeli käsittelee sulautetun järjestelmän suunnitteluprosessia. Suhteellisen helppolukuinen.

a) Henzinger, T.A., Kirsch, C.M., Sanvido, M.A.A., and Pree, W., *From control models to real-time code using Giotto*. IEEE Control Systems Magazine. Volume 23, Issue 1, Feb. 2003 Page(s):50 – 64. Digital Object Identifier 10.1109/MCS.2003.1172829

Seuraavan artikkelin kuvaa miten olemassa olevaan suunnittelumenetelmään (SDL ja MSC) voidaan lisätä ajoituksiin liittyvää informaatiota analyyseja varten. Saattaa edellyttää joitain pohjatietoja noista menetelmistä.

b) Wang, S., and Tsai, G. *Specification and Timing Analysis of Real-Time Systems*. Real-Time Systems, 28, 69-90, 2004.

Kolmas artikkeli käsittelee ohjelmointikielen tasolla tapahtuvia rajoituksia, jotta ohjelman turvallisuutta (safety) voidaan analysoida jo käännösaikana. Kieli pohjautuu c-kieleen, joten sen tunteminen voi helpottaa artikkelin lukemista. Jos valitset tämän artikkelin, niin mieti lukiessasi miten paljon nuo rajoitteet vaikuttaisivat ohjelmaa kirjoitettaessa.

c) Kowshik, S., Dhurjati, D., and Adve, V. *Ensuring code safety without runtime checks for real-time control systems*. In Proceedings of the 2002 international Conference on Compilers, Architecture, and Synthesis For Embedded Systems (Grenoble, France, October 08 - 11, 2002). CASES '02. ACM Press, New York, NY, 288-297. 2002. DOI=http://doi.acm.org/10.1145/581630.581678