

OHJ-4010 Rinnakkaisuus Tentti 4.2.2008

Tentissä ei saa olla mukana mitään materiaalia. Tentin tarkistaa Hannu-Matti Järvinen.

1. Tehtävä

Ohessa on kuusi vaihtoehtoa lukija-kirjoittajaongelmassa esiintyvän kirjoittajan nälkiintymisen ratkaisemiseksi. Käytössä ovat semaforit `write_mutex`, `mutex` ja `extra`, jotka kaikki on alustettu arvoon 1. Lukijoilla on lisäksi yhteinen kokonaislukumuuttuja `read_count`, joka on alustettu nolaksi. Vaihtoehtoiset algoritmit lukijan varausalgoritmeiksi ovat A-C ja vapautusalgoritmeiksi D-F. Lisäksi on annettu kirjoittajan lukitus- ja vapautusalgoritmit.

A	B	C
<pre>extra.p(); extra.v(); mutex.p(); if read_count = 0 then read_count := read_count+1; write_mutex.p(); end if; mutex.v();</pre>	<pre>extra.p(); mutex.p(); extra.v(); read_count := read_count+1; if read_count = 1 then write_mutex.p(); end if; mutex.v();</pre>	<pre>extra.p(); mutex.p(); read_count := read_count+1; if read_count = 1 then write_mutex.p(); extra.v(); end if; mutex.v();</pre>

D	E	F
<pre>p(mutex) read_count := read_count-1; if read_count = 0 then write_mutex.v(); end if; mutex.v();</pre>	<pre>p(mutex) read_count := read_count-1; if read_count = 0 then write_mutex.v(); else extra.v() end if; mutex.v();</pre>	<pre>p(mutex) if read_count = 1 then write_mutex.v(); end if; read_count := read_count-1; mutex.v();</pre>

Kirjoittajan algoritmit	
varaus	<pre>extra.p(); write_mutex.p();</pre>
vapautus	<pre>extra.v(); write_mutex.v();</pre>

OK

read_count korruptoitunut, oikea järj. pettä

kirj. voi nälkiintyä

Muodostetaan lukijan lukitusalgoritmeista (A, B, C) ja vapautusalgoritmeista (D, E, F) algoritmiparit AD, AE, AF, BD, BE, BF, CD, ~~CE~~, CG. Mitkä näistä yhdistelmistä toimii oikein, mitkä väärin? Semaforien odotetaan toimivan siten, että jonoa puretaan saapumisjärjestyksessä. Mikäli yhdistelmä toimii oikein ja tehokkaasti, siitä riittää perusteluksi OK. Mikäli algoritmissa on tehokkuusongelma tai virhe, se pitää perustella lyhyesti malliin "ratkaisu lukkiutuu joskus".

2. Tehtävä

Selitä *mahdollisimman lyhyesti* seuraavat käsitteet:

- a. poissulkeminen (mutual exclusion)
- b. synkronointi (synchronization)
- c. lukkiutuminen (deadlocking)
- d. nälkintyminen (starvation)
- e. kriittinen alue (critical section)
- f. jakamaton eli atominen operaatio (indivisible or atomic operation)

3. Tehtävä

Toteuta tuottaja-kuluttajaongelma käyttäen Adan tehtäviä ja Adan kohtaamismekanismeja (semaforit eivät siis ole sallittuja). Tee siis tehtävien Tuottaja, Kuluttaja ja Puskuri määrittely- ja runko-osat.

4. Tehtävä

Aterioivia filosofeja voi olla pöydän ympärillä vaihteleva määrä.

- a. Mikä on pienin määrä filosofeja, jolla lukkiutuminen voi tapahtua? Anna esimerkki tilanteesta, ja selitä, miksi vähemmällä määrällä filosofeja lukkiutumista ei voi esiintyä.
- b. Mikä on pienin määrä filosofeja, jolla nälkintyminen voi tapahtua? Anna esimerkki tilanteesta, ja selitä, miksi vähemmällä määrällä filosofeja nälkiintymistä ei tapahdu.