

OHJ-4010 Rinnakkaisuus Tentti 26.11.2007

Tentissä ei saa olla mukana mitään materiaalia. Tentin tarkistaa Jyke Jokinen.

1. Tehtävä

Ohessa on kolme suorittimien väliseen lukitukseen tarkoitettua algoritmia. Toimivatko algoritmit yleisesti tai jossain rajoitetussa tilanteessa? Jos algoritmi ei toimi lainkaan, niin miksi ei? Jos toimii, niin onko algoritmien käytöllä jotain rajoituksia?

A	B	C
<pre>turn : integer := 0; procedure lock (me : integer) is begin loop while turn /= me loop null; end loop; turn := me; for j in 0..1000 loop null; end loop; exit when turn = me; end loop; end lock; procedure unlock is begin turn := 0; end unlock;</pre>	<pre>flags : array (0..1) of boolean; turn : integer := 0; procedure lock (me : integer; she : integer) is begin flags(me) := true; turn := she; while flags(she) and turn /= me loop null; end loop; end lock; procedure unlock(me : integer) is begin flags(me) := false; end unlock;</pre>	<pre>flags : array (0..1) of boolean; turn : integer := 0; procedure lock (me : integer; she : integer) is begin flags(me) := true; while flags(she) loop if turn = she then flags(me) := false; while turn = she loop null; end loop; flags (me) := true; end if; end loop; end lock; procedure unlock(me : integer; she : integer) is begin turn := me; flags(she) := false; end unlock;</pre>

2. Tehtävä

Mitä tarkoittaa lukkiutuminen? Lukkiutumisen ehdot. Miten lukkiutumisen voi *estää* (yksi tapa riittää)?

3. Tehtävä

Lukija-kirjoittajaongelmassa on yhtenä osa-ongelmana se, että kirjoittajat voivat nääntyä nälkään. Toteuta Adalla tehtävä (task) resurssi, joka valvoo yhteisen muistialueen käyttöä. Lukijat ja kirjoittajat pyytävät resurssi-tehtävältä pääsyä muistialueelle ja käytettyään muistialuetta ilmoittavat käytön päättymisestä. Resurssi-tehtävän tulee huolehtia siitä, että kirjoittajat eivät nälkiinny, ja lukija-kirjoittajaongelman alkuperäiset ehdot säilyvät. (Muista kirjoittaa myös resurssi-tehtävän määrittelyosa.)

4. Tehtävä

Ohessa on kolme eri versiota tuottaja-kuluttajaongelmasta. Niissä oletetaan, että käytettävissä ovat semaforit free, full, mutex1 ja mutex2. Kaikissa versioissa ei kaikkia semaforeja ole käytössä, mutta niille voidaan aina olettaa seuraavat alustusarvot: full = 0, free = size, mutex1 = 1 ja mutex2 = 1. Kaikille versioille yhteinen määrittelyosa on seuraava:

```
type T; -- Mikä tahansa sijoituksen salliva tyyppi
size : constant integer := 100;
buf : array (1..size) of T;
in_index : integer := 1; out_index : integer := 1;
```

A	B	C
<pre>task tuottaja is item : T; begin loop -- tuota item mutex1.P; buf (in_index) := item; in_index := in_index mod size + 1; full.V; end loop; end; task kuluttaja is item : T; begin loop full.P; item := buf (out_index); out_index := out_index mod size + 1; mutex1.V; -- kuluta item end loop; end;</pre>	<pre>task tuottaja is item : T; begin loop -- tuota item free.P; buf (in_index) := item; in_index := in_index mod size + 1; full.V; end loop; end; task kuluttaja is item : T; begin loop full.P; mutex2.P; item := buf (out_index); out_index := out_index mod size + 1; mutex2.V; free.V; -- kuluta item end loop; end;</pre>	<pre>task tuottaja is item : T; begin loop -- tuota item mutex1.P; buf (in_index) := item; in_index := in_index mod size + 1; mutex1.V; full.V; end loop; end; task kuluttaja is item : T; begin loop full.P; mutex1.P; item := buf (out_index); out_index := out_index mod size + 1; mutex1.V; -- kuluta item end loop; end;</pre>

Kaikki toteutukset ovat alkuperäisen yleistoteutuksen muunnelmia. Vastaa kukin toteutuksen kohdalla seuraaviin kysymyksiin: Voidaanko toteutusta käyttää jossain tilanteissa? Mitkä ovat ne rajoitteet, joilla toteutusta voidaan käyttää, tai miksi toteutus on täysin käyttökelvoton?