

OHJ-4010 Rinnakkaisuus Tentti 15.5.2012

Tentissä ei saa käyttää ylimääräistä kirjallista materiaalia, laskimia, tietokoneita tai muita lunttausvälineitä.

Tentin tarkistaa Johannes Koskinen.

1. Tehtävä

Selitä *lyhyesti* seuraavat käsitteet (1 p/kohta):

- a) poissulkeminen (mutual exclusion)
- b) synkronointi (synchronization)
- c) jakamaton eli atominen operaatio (indivisible or atomic operation)
- d) nälkiintyminen (starvation)
- e) monitori (monitor)
- f) lukkiutuminen (deadlocking)

2. Tehtävä

Mikä on kriittinen alue? Miksi se muodostaa ongelman? Miten sen aiheuttama ongelma voidaan ratkaista? Esitä vastauksesi esimerkkiä apuna käyttäen. (6 p)

3. Tehtävä

Mitä tarkoittaa lukkiutuminen? Mitkä ovat lukkiutumisen ehdot? Lukkiutumisen käsittelyyn on olemassa kolme lähestymistapaa: estäminen (prevention), välttäminen (avoidance) ja laukaiseminen (recovery). Kuvaa lyhyesti kukin termi ja anna lyhyt kuvaus kunkin tavan käyttämisestä. (8 p)

4. Tehtävä

Ohessa on yksi lukitusrutiiniehdocas.

lock_variable : integer := 0;

Lukitusrutiini	Vapautusrutiini
<pre>procedure Lock (my_number : integer) is begin loop while lock_variable = 1 loop null; -- tyhjä silmukka end loop; for j in 0 .. 10000 loop null; -- odotetaan hetki muita end loop; lock_variable := my_number; -- poistutaan, jos meidän vuoro exit when lock_variable = my_number; end loop; end Lock;</pre>	<pre>procedure Unlock is begin lock_variable := 0; end Unlock;</pre>

Vastaa *lyhyesti* seuraaviin kysymyksiin:

- a) Voidaanko algoritmia käyttää sellaisenaan monisuoritinkoneen suorittimien välisenä poissulkemisrutiinina? Jos voi, onko käytöllä joitain rajoituksia? Jos ei voi, tee algoritmista käyttökelpoinen, jos se on mahdollista *pienellä* korjauksella. (2 p)
- b) Voidaanko algoritmia käyttää sellaisenaan yksisuoritinkoneessa prosessien tai säikeiden välisenä poissulkemisrutiinina? Jos voi, onko käytöllä joitain rajoituksia? Jos ei voi, tee algoritmista käyttökelpoinen, jos se on mahdollista *pienellä* korjauksella. (2 p)