

Nimi	Nimikirjoitus	Opiskelijanumero	Pisteet

## Käyttöjärjestelmät, minikoe 3, 29.2.2016 (6p)

Kirjoita vastauksesi tälle koepaperille kunkin tehtävän kohdalle. Huomaa, että koepaperi on 2-puolinen.

- a) [3 p] Monitori. Prosessi P on toteutettu 4 säikeellä: A, B1, B2 ja B3. Säikeet Bi tekevät alustuksensa ja odottavat koodin kohdassa Binit, kunnes säie A on saanut tehtyä oman alustuksensa loppuun koodin kohdassa Ainit. Säie A odottaa sitten kohdassa Ainit, kunnes kaikki säikeet Bi ovat saaneet laskentansa valmiiksi kohdassa Bdone. Tämän jälkeen kaikki säikeet suorittavat koodinsa loppuun. Prosessien välisen synkronointiongelman ratkaisu on toteutettu monitorilla Sync. Prosessien A ja Bi pseudokoodit ovat

<u>A</u>	<u>B1, B2, B3</u>
...	...
Sync.Ainit();	Sync.Binit();
...	...
	Sync.Bdone();
	...

Anna monitorin Sync ja sen metodien Ainit(), Binit() ja Bdone() ratkaisu pseudokoodina. Oletusarvoisesti käytössä on Hoaren *signal-and-wait* signalointisemantiikka. Käyttäessäsi jotain muuta signalointisemantiikkaa mainitse asiasta.

- b) [1 p] Hajautetussa ympäristössä on usea prosessi, joista kukin voi haluta kriittisessä vaiheessa päivittää kolme eri tietokantaa eri palvelimilla ajan tasalle. Kuinka tämä hajautetun järjestelmän kriittisen vaiheen ongelma voidaan ratkaista?
- c) [1 p] Käyttöjärjestelmän säikeet P, Q, R ja S voivat olla suorituksessa samanaikaisesti. Ne kaikki käyttävät 7 kriittistä aluetta (CS1-CS7). Yleensä kriittisiä alueita on vain yksi kerrallaan käytössä. Joskus harvoin CS2:n yhteydessä tarvitaan myös alueita CS1 tai CS3. Joskus harvoin CS3:n tai CS6:n yhteydessä tarvitaan myös alueita CS2 tai CS5.  
Kuinka voidaan estää, että kyseiset säikeet eivät koskaan lukkiudu? Voitko todistaa ratkaisun oikeellisuuden? (Ratkaisun tulee sallia se, että prosessi yleensä tarvitsee ja varaa vain yhden kriittisen alueen käyttöönsä.)
- d) [1 p] Mitä tietoja ja mitä tietorakenteita lukkiutumisen havaitsemisalgoritmi (DDA) tarvitsee?