

## Käyttöjärjestelmät (5 op), koe 14.8.2019

Kirjoita **jokaiseen** vastauspaperiisi kurssin nimi, pvm, oma nimi, nimikirjoitus ja opiskelijanumero. Ei laskimia. Koepaperi on kaksipuolinen. Kuhunkin tehtävään riittää noin 1-2 sivun vastaus. Tämä on tavallinen erilliskoe ja kattaa koko kurssin. Vastaa kaikkiin tehtäviin 1-4.

### 1. [9 p] Säikeet, samanaikaisuus

- a. [3 p] Anna pseudokoodiesimerkki kriittisen vaiheen ongelmasta. Anna esimerkkiisi liittyvä skenaario (A), jossa koodi toimii oikein. Anna esimerkkiisi skenaario (B), jonka lopputulos on virheellinen. Selitä, miksi se on virheellinen.
- b. [2 p] Anna kriittisen vaiheen ongelman semafori-ratkaisu. Selitä, miksi skenaariosi (B) ei nyt ole mahdollinen.
- c. [4 p] Selitä, mikä on tuottaja-kuluttaja ongelma äärettömän kokoisen puskurin tapauksessa. Minkälaisessa tilanteessa ratkaisu voi perustua äärettömän kokoiseen puskuriin, vaikka käytännössä kaikki muistialueet ovat äärellisen oloisia? Anna semaforien avulla toteutettu (pseudokoodi) tuottaja-kuluttaja ongelman ratkaisu äärettömän kokoisen puskurin tapauksessa.

### 2. [9 p] Lukkiutuminen

- a. [3 p] Selitä, mikä on Aterioivien filosofien ongelma. Anna siihen lukkiutuva skenaario. Selitä, mistä lukkiutuminen aiheutuu scenaariossasi.
- b. [3 p] Miten lukkiutumisen havaitsemisalgorithmi (DDA, Deadlock Detection Algorithm) pääpiirteissään toimii? Milloin ja kuinka sitä käytetään lukkiutumisiongelman ratkaisuun?
- c. [3 p] Käyttöjärjestelmän ytimen koodissa lukkiutuminen usein ehkäistään ennalta sillä tavoin, että kriittiset vaiheet varataan aina tietyssä järjestyksessä, esimerkiksi aakkosjärjestyksessä "A B C D". Miksi tämä menetelmä estää lukkiutumisen kaikissa skenaarioissa? Missä tilanteessa järjestys "C B D A" olisi parempi kuin järjestys "A B C D"? Perustele.

**KÄÄNNÄ**

3. [9 p] **Virtuaalimuisti**

- a. [2 p] Sivun koko on 4KB. Kuinka viitattu tieto löytyy keskusmuistista yksitasoista sivuttavaa virtuaalimuistia käyttäen?  
Käytä esimerkkinä koodissa viitattua tavuosoitetta 0x11223344. Mihin keskusmuistin osoitteeseen viittaus kohdistuu?
- b. [3 p] Mitä tarkoittaa "ruuhkautuminen" (trashing)? Mistä se aiheutuu? Mitä haittaa siitä on? Kuinka se havaitaan käytännössä?  
Milloin ja miten siitä voi toipua tai sen voi välttää? Milloin siitä ei voi toipua ja mitä silloin tehdään?
- c. [4 p] Clock ja PFF (Page Fault Frequency) algoritmeja voidaan kumpaakin käyttää sivunpuutoskeskeytyksen yhteydessä. Mitkä ongelmat kyseiset algoritmit ratkaisevat ja kuinka algoritmit pääpiirteittäin toimivat?

4. [9 p] **Tosiakajärjestelmien vuoronanto (real time scheduling)**

- a. [3 p] Miksi tosiaikajärjestelmissä ei voi yleensä käyttää samoja vuoronantomenetelmiä (esim. FIFO tai prioriteettipohjaiset ratkaisut) kuin tavallisissa järjestelmissä? Esimerkiksi, miksi FIFO olisi huono vuoronantomenetelmä tosiaikajärjestelmissä?
- b. [3 p] Rate Monotonic Scheduling (RMS) on kuitenkin prioriteettipohjainen vuoronanto menetelmä nimenomaan tosiaikajärjestelmiin. Milloin sitä voidaan käyttää ja kuinka se toimii? Kuinka prioriteetit määräytyvät?
- c. [3 p] Mikä on realiaikajärjestelmissä joskus esiintyvä prioritetin kääntymisongelma (priority inversion)? Kuinka se voidaan ratkaista? (Yksi ratkaisumenetelmä riittää.)