

Käyttöjärjestelmät, välikoe 1, 8.3.2017

Kirjoita **jokaiseen** vastauspaperiisi kurssin nimi, pvm, oma nimi, nimikirjoitus ja opiskelijanumero. Kuhunkin tehtävään riittää 1-2 sivun vastaus.

Anna kunkin tehtävän vastaus omalla konseptiarkillaan.

Kukin tehtävä vastaa samannumeroista minikoetta. Tehtävä 4 on siis minikoe 4.

HUOM: Palauta jokainen tehtävä omalla konseptiarkillaan oikeaan pinoon!

1. [6 p] Välimuisti.

- [1 p] Anna esimerkit alueellisesta paikallisuudesta ohjelmakoodin koodi- ja data-viitteissä.
- [1 p] Anna esimerkit ajallisesta paikallisuudesta ohjelmakoodin koodi- ja data-viitteissä.
- [2 p] 32 KB välimuisti sisältää 1024 kappaletta 8 sanan (32 tavun) rivejä. Mikä vaikutus välimuistin suorituskykyyn olisi, jos siinä olisikin 256 kappaletta 32 sanan rivejä? Missä tilanteessa tämä olisi parempi vaihtoehto ja miksi?
- [1 p] Laitteistossa on 4 ydintä ja rekisterin viittausaika on 1 ns. Kullakin ytimellä on oma välimuisti (L1). Välimuistin L1 koko on 32KB, saantiaika 2 ns ja osumasuhde 99%. Keskusmuistin koko on 8GB ja saantiaika 60ns (välimuistin riville). Mikä on keskimääräinen muistinviittausaika?
- [1 p] Jatkoa edelliseen. Välimuistiin lisätään toinen taso (L2). Kaikille ytimille yhteisen välimuistin L2 koko on 4 MB, saantiaika 5 ns ja osumasuhde 90%. Mikä nyt on keskimääräinen muistinviittausaika?

2. [6 p] Semaforit.

- [2 p] Meillä on 4-säikeinen ohjelma P, jossa kaikki 4 säiettä kasvattavat yhteisen muuttujan X (alkuarvo 0) arvoa seuraavanlaisella koodilla:

```
Thread T1 (threads T2, T3, T4 similarly)
int a, b;      /* local variables in thread T1 */
...
i1:  a = X;
i2:  a++;      /* a = a+1 */
i3:  X = a;
i4:  b = a;
...
```

Anna skenaario, jossa ohjelman P suorituksen loppuessa muuttujan X arvo on virheellisesti 3.

- [2 p] Jatkoa edelliseen. Kuinka ohjelmaa pitäisi muuttaa, jotta se toimisi aina oikein (X:n arvo lopussa on aina 4)? Muuttujan X arvon lisäys on kriittinen tekijä ja sitä ei saa viivästyttää yhdenkään prosessin vaihdon vertaa.
- [2 p] Editori ja näppäimistön ajuri. Tekstieditori TE lukee merkkipuskurista B merkkejä yksi merkki kerrallaan ja tekee merkin vaatimat muutokset editoitavaan tiedostoon. Näppäimistön laiteajuri DD lukee näppäimistön painallukset (yksi kerrallaan) ja kirjoittaa niitä vastaavat merkit merkkipuskuriin B. Puskuriin B mahtuu 200 merkkiä.

Puskuria käsitellään rutiineilla Put(buf, c) ja c=Get(buf), joiden avulla merkkipuskuriin buf laitetaan ja sieltä otetaan pois merkkejä. Niiden sisäinen samanaikaisuuden hallinta on ratkaistu rutiineissa Put() ja Get().

Anna TE:n ja DD:n välinen synkronointi- ja kommunikointiongelman ratkaisu semaforien ja yhteisessä muistissa olevan puskurin (B) avulla. Esitä ratkaisu TE:n ja DD:n pseudokoodien avulla. Määrittele selkeästi kaikki käyttämäsi semaforit ja muut synkronointiin tarvittavat tietorakenteet alkuarvoineen.

3. [6 p] **Monitori ja lukkiutuminen.**

- a. [2 p] Kuinka monitorin ehtomuuttuja ja sen operaatiot eroavat semaforista ja sen operaatioista?
- b. [2 p] Mitä tarkoittaa käsite "Lampson & Redell signalointisemantiikka"?
Kuinka (ja miksi) se pitää ottaa huomioon monitorin koodissa?
- c. [2 p] Kuinka voidaan toteuttaa useampi mahdollisesti yhtäaikainen kriittinen vaihe siten, että lukkiutuminen ei ole lainkaan mahdollista? Mitään kriittistä vaihetta ei saa pitää varattuna silloin kun sitä ei käytetä. Miten voit todistaa, että ratkaisu on oikein eikä lukkiudu koskaan?

4. [6 p] **Virtuaalimuisti.** 2-tasoisessa sivuttavassa virtuaalimuistissa tavuosoite on 32-bittinen ja sivun koko on 4KB.

- a. [2 p] Kuinka osoitteenmuutos virtuaaliosoitteesta 0x11223344 fyysiseen 0xABCD3344 osoitteeseen tapahtuu?
- b. [2 p] Kuinka kauan (pääpiirteissään) osoitteenmuutos voi eri vaihtoehdoissa kestää?
- c. [2 p] Clock-algoritmi valitsee korvattavan sivun nyt käytössä olevista sivukehyksistä. Miksi Clock on parempi kuin FIFO? Miksi Clock on parempi kuin LRU? Mitä "parempi" tarkoittaa tässä yhteydessä?