

## Käyttöjärjestelmät (5 op), koe 14.6.2019

Kirjoita **jokaiseen** vastauspaperiisi kurssin nimi, pvm, oma nimi, nimikirjoitus ja opiskelijanumero. Ei laskimia. Koeaika 3.5 h. 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] Samanaikaisuus

- a. [3 p] Selitä, mikä on kriittisen vaiheen ongelma ja miksi se on vaikea ratkaista. Kuinka kriittisen vaiheen ongelma ratkaistaan monitorissa?
- b. [3 p] Selitä, mikä on yleinen synkronointiongelma. Monitorissa yleinen synkronointiongelma ratkaistaan ehtomuuttujien (condition variables) avulla. Kuinka ehtomuuttujat ja niiden käyttö eroaa semaforeista ja niiden käytöstä?
- c. [3 p] Juoksurata ja semafori. Juoksurata on 400m pitkä. Ann sekä hänen ystävänsä Bill ja Charlie tulevat sinne usein ja juoksevat 4000m. Ann on sosiaalinen ja odottaa joka kierroksen jälkeen, että jompi kumpi pojista on saanut hänet kiinni (ainakin sama määrä juostuja kierroksia). Pojat ovat kilpailuhenkisiä, eivätkä odota ketään.

Ratkaise syntyvä synkronointiongelma semaforeilla. Anna ratkaisusi muokkaamalla alla olevia juoksijoiden pseudokoodeja. Muista määritellä kaikki käyttämäsi semaforit alkuarvoineen.

Ann	Bill	Charlie
===	====	=====
for (i=1 to 10)	for (i=1 to 10)	for (i=1 to 10)
<juokse kierros>	<juokse kierros>	<juokse kierros>
<synkronoi>	<synkronoi>	<synkronoi>

### 2. [9 p] Lukkiutuminen, muistinhallinta, virtuaalimuisti

- a. [4 p] Oletetaan, että käyttöjärjestelmän prosessi P varaa ensin kriittisen vaiheen kv-1 ja sitten joskus myöhemmin hyvin harvoin myös kriittisen vaiheen kv-2 ennenkuin se lopulta vapauttaa varaamansa kriittiset vaiheet (kv). Joskus taas P varaa ensin kv-2:n ja sitten vähän myöhemmin ehkä myös kv-1:n. Käyttöjärjestelmän prosessi Q toimii samalla tavalla. Anna skenaario (A), jossa P ja Q eivät lukkiudu. Anna skenaario (B), jossa P ja Q lukkiutuvat. Perustelee.
- b. [5 p] Millä tavoin P:n ja Q:n koodia tulisi muuttaa, jotta voitaisiin taata, että lukkiutumista ei voi tapahtua? (Suorituskykyvaatimusten vuoksi ei ole sallittua, että P tai Q aina varaisivat sekä kv-1:n että kv-2:n yhtä aikaa alusta pitäen, koska P ja Q käyttävät lähes aina vain yhtä kv:tä kerrallaan.) Selitä, mitä ongelmia P:n ja Q:n koodin muuttamisessa on. Pitääkö molempia muuttaa? Selitä, miksi skenaariosi B ei nyt voi toteutua. Perustelee, miksi ratkaisusi takaa, että lukkiutumista ei voi koskaan tapahtua.

**KÄÄNNÄ**

3. [9 p] **Muistinhallinta, virtuaalimuisti**

Käytössä on 2-tasoinen sivuttava muistinhallinta, 32-bittiset virtuaaliset tavuosoitteet, 32-bittiset fyysiset keskusmuistiosoitteet ja 4 KB sivut. Sivutaulun alkio on 4 B. Viittaus kohdistuu ohjelman käyttämään muistiosoitteeseen 0x12345678. Osoitteenmuunnoksessa tämä osoite kuvautuu johonkin keskusmuistiosoitteeseen.

- a. [3 p] Kuinka tämä osoitteenmuunnos tehdään sivutaulujen avulla? Kuinka suuria sivutaulut ovat?
- b. [2 p] Oletetaan nyt, että käytössä onkin käänteinen sivutaulu. Kuinka osoitteenmuunnos tehdään käänteisen sivutaulun avulla? Kuinka suuri käänteinen sivutaulu on?
- c. [2 p] Käytössä on myös 256-alkioinen TLB (Translation Lookaside Buffer). Kuinka osoitteenmuunnos tehdään TLB:n avulla ja kuinka TLB nopeuttaa osoitteenmuutoksen tekemistä?
- d. [2 p] Mikä ovat "hutien" TLB-huti (TLB miss), välimuistihuti (cache miss), sivutauluhuti (page fault) karkeat kustannukset?

4. [9 p] **Vuoronanto**

- a. [3 p] Mitä hyötyä aikaviipalevuoronannolla (time slice scheduling) on verrattuna FIFO-vuoronantoon? Mitä haittaa?
- b. [3 p] Miten Rate Monotonic Scheduling (RMS) vuoronantomenetelmä toimii? Mitä tietoja se tarvitsee ja mistä nuo tiedot saadaan? Minkälaiseen tosiaikajärjestelmien laskentaympäristöön RMS on tarkoitettu? Mitä etua RMS:llä on verrattuna useimpiin muihin tosiaikajärjestelmien vuoronantoalgoritmeihin (esim. Earliest Deadline First)?
- c. [3 p] Selitä, mikä on prioriteetin kääntymisongelma ja miksi siitä on haittaa nimenomaan tosiaikajärjestelmissä. Miksi sitä ei ilmene tavallisissa pöytäkonejärjestelmissä? Anna myös yksi ratkaisumenetelmä tähän ongelmaan.