# Käyttöjärjestelmät (5 op), erilliskoe/uusintakoe 17.4.2019

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.
Valitse yksi seuraavista vaihtoehdoista ja mainitse valintasi koepaperissa. Oletusvalinta on (b).

(a) Kevään 2019 luentokurssin minikokeiden 1-4 uusintakuulustelut. Vastaa 1-4 minikokeeseen.
(b) 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.

   b. [3 p] Milloin ja kuinka kriittisen vaiheen ongelma (critical section problem) ratkaistaan puhtaasti ohjelmiston avulla (yhteinen muisti, luku- ja kirjoituskäskyt)?
   Milloin ja kuinka kriittisen vaiheen ongelma ratkaistaan keskeytykset estämällä?
   Milloin ja kuinka kriittisen vaiheen ongelma ratkaistaan erityiskonekäskyjen (esim. Compare&Swap) avulla?

   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 molemmat pojat ovat saaneet 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. [3 p] Käyttöjärjestelmän prosessit P ja Q päivittävät yhteiskäyttöisiä tietorakenteita A ja B. Sekä A:n että B:n päivitysten täytyy tapahtua atomisesti. A:ta suojaa kriittinen vaihe cs-A ja B:tä kriittinen vaihe cs-B.
   Millä tavoin P:n ja Q:n koodi tulisi toteuttaa, jotta voitaisiin taata, että lukkiutumista ei voi tapahtua?
   Suorituskykyvaatimusten vuoksi ei ole sallittua, että P ja Q aina varaisivat sekä cs-A:n että cs-B:n yhtä aikaa alusta pitäen, koska useimmiten P ja Q käyttävät vain jompaa kumpaa tietorakennetta kerrallaan.
   Selitä, mitä ongelmia P:n ja Q:n koodin muuttamisessa on. Miksi ratkaisu ei ole triviaali?
   Perustele, miksi ratkaisusi takaa, että lukkiutumista ei voi koskaan tapahtua.

   b. [3 p] Minkä muistinhallinnan ongelman Buddy-algoritmi ratkaisee ja kuinka se sen tekee?
   Mitä etuja Buddy-systeemistä on verrattuna saman ongelman muihin ratkaisuihin?

   c. [3 p] Miten voidaan tietää, että virtuaalimuistia käytettäessä jollakin prosessilla on liian vähän muistitilaa (sivukehyksiä)?
   Anna jokin menetelmä, jolla dynaamisesti tällaisessa tilanteessa prosessille annetaan lisää muistitilaa.
   Kuvaile menetelmän toiminta pääpiirteissään.

**KÄÄNNÄ**

3. [9 p] **Prosessien/säikeiden vuoronanto**
   a. [1 p] Mihin dataan (vuoronantoalgoritmi) FIFO:n toiminta perustuu ja kuinka tämä data saadaan käyttöön?
      Onko FIFO keskeyttävä (preemptive) algoritmi vai ei?

   b. [2 p] Mihin dataan SPN:n (Shortest Process Next) toiminta perustuu ja kuinka tämä data saadaan käyttöön?
      Onko SPN keskeyttävä algoritmi vai ei?

   c. [2 p] Millä perusteella SPN on parempi kuin FIFO?
      Millä perusteella FIFO on parempi kuin SPN?

   d. [2 p] Tosiaikajärjestelmien (real time systems) vuoronanto perustuu yleensä aikarajoihin (deadlines).
      Selitä, kuinka aikarajavuoronanto (deadline scheduling) yleisesti ottaen toimii?

   e. [2 p] RMS (Rate Monotonic Scheduling) algoritmi välttää täysin useimmat aikarajavuoronantoon liittyvät ongelmat.
      Mihin dataan RMS:n toiminta perustuu ja kuinka tämä data saadaan käyttöön?
      Miksi RMS:ää ei voida käyttää kaikkeen tosiaikajärjestelmien vuoronantoon?

4. [9 p] **Levyjen ja tiedostojen hallinta, pääsynvalvonta**
   a. [3 p] Minkä kovalevyjen (HDD) hallintaan liittyvän ongelman C-SCAN algoritmi ratkaisee ja kuinka algoritmi pääpiirteissään toimii?
      Minkä C-SCAN algoritmin ongelman sen variantti Linus Deadline Scheduler ratkaisee ja kuinka variantin algoritmi pääpiirteissään toimii?
      Minkä C-SCAN algoritmin ongelman sen variantti Linus Anticipatory I/O Scheduler ratkaisee ja kuinka variantin algoritmi pääpiirteissään toimii?

   b. [3 p] Indeksoitu tiedosto (indexed file).
      Montako indeksiä siinä on? Mikä on indeksin (indeksien) koko?
      Anna esimerkki tiedoston käyttömallista (use case), jolloin indeksoitu tiedosto olisi järkevin organisaatiotapa tiedostolle?
      Miksi indeksoidun tiedoston indeksi olisi järkevää toteuttaa B-puuna, eikä tavallisena peräkkäishakuindeksinä?

   c. [3 p] Rooliperustainen järjestelmän pääsynvalvonta (Role-based access control, RBAC). Jussilla on oikeus professorina editoida tutkielma-tietokantaa THESIS ja henkilökunnan jäsenenä lukea sisäisiä tiedotteita tietokannasta BULLETINS.
      Näytä RBAC pääsymatriisi (tai pääsymatriisit), joilla nämä oikeudet on toteutettu järjestelmässä.