

TIE-02500 Rinnakkaisuus

Tentti 30.10.2017

Tentin vastuuhenkilö: `jyke.savia@tut.fi`

Laskimen käyttö on kiellettyä. Tenttipaperia ei tarvitse palauttaa.

Muista kirjoittaa kaikkiin vastauspapereihin nimesi ja opiskelijanumerosi.

Vastauksessa olet vastaavasi sellaisen ihmisen esittämään kysymykseen, joka tuntee kohtalaisen hyvin ohjelmistotekniikan aihealuetta muutoin paitsi juuri tämän kysymyksen osalta. Mieti etukäteen vastauksesi pääkohdat ja lajittele ne johdonmukaiseen järjestykseen — älä kirjoita yhteen pötköön kaikkea mieleen tulevaa. Muista vastata kaikkiin tehtävän kysymyslauseisiin, sillä täysia pistettä ei voi saada jos kaikkiin kysytyihin asioihin ei ole vastattu. Jos vastaus vaatii ohjelmakoodin kirjoittamista, sen ei tarvitse olla pikkulleen syntaksiltaan oikein. Mikä tahansa johdonmukaisesti käytetty ja yleisessä käytössä olevia ohjelmointirakenteita sisältävä koodin esitysmuoto käy.

Ohjelma 1: Vähennysohjelma

```
namespace moduuli {
    unsigned int Countdown = 42;

    void Vahenna() {
        unsigned int TyoKopio = Countdown;
        TyoKopio = TyoKopio - 1;
        Countdown = TyoKopio;
    }
}
```

1. Ohjelman 1 funktiota `Vahenna()` kutsuu ohjelman käynnistyksen jälkeen kaksi suoritussäiettä. Molempien tulisi vähentää muuttujan `CountDown` arvoa yhdellä, joten odotettu lopputulos on 40. Näin ei ohjelmoijan hämmästykseksi kuitenkaan aina tapahdu.
 - (a) [3 pistettä] Miksi lopputulos voi olla väärä? Selosta yksi suoritusketju, jossa odotettu lopputulos saadaan ja toinen, joka menee pieleen.
 - (b) [3 pistettä] Mitä on suoritussäikeiden poissulkeminen? Miten sen avulla ohjelmassa voitaisiin varmistaa, että odotettu lopputulos saadaan laskennassa aina? Tee tarvittavat muutokset ohjelmaan.

6p.

2. Kerro lyhyesti mitä seuraavat asiat ovat?

10p.

- (a) [1 piste] Rinnakkaisen ohjelman lukkiutuminen (deadlock)
 - (b) [1 piste] Suoritussäikeen nälkiintyminen (starvation)
 - (c) [2 pistettä] Minkälainen operaatio on Barrier-synkronointi?
 - (d) [2 pistettä] Atominen konekäsky. Anna esimerkki tällaisesta käskystä (selosta myös mitä käsky tekee – pelkkä nimi ei riitä)
 - (e) [2 pistettä] rw-lukko (read-write lock). Miten toimii? Missä tilanteissa sitä tyypillisesti voi käyttää rinnakkaisessa ohjelmassa?
 - (f) [2 pistettä] Käänteisprioriteettiongelma (priority inversion problem)?
3. [6 pistettä] "Klassinen" rinnakkaisuuden esimerkki on tuottaja-kuluttaja (producer-consumer). Selosta minkälaisesta ohjelman rakenteesta on kyse? Minkälaisia rinnakkaisuuden ongelmia rakenteen käyttöön liittyy? Hahmottele säieturvallinen ohjelmakoodiratkaisu ongelmaan kun käytössäsi on **semaforit** rinnakkaisuuden hallintaan. (säikeiden luonnin oletetaan tapahtuvan tämän koodin ulkopuolella, joten sitä ei tarvitse ottaa hahmotelmaan mukaan). Arvostelussa ei vähennetä pisteitä syntaksiltaan väärin kirjoitetusta ohjelmakoodista (kunhan merkityksen ymmärtää).