

# TIE-02500 Rinnakkaisuus

Tentti 18.9.2014

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

Laskimen käyttö on kiellettyä.

Kirjoita selkeällä käsialalla.

Muista kirjoittaa kaikkiin vastauspapereihin nimesi ja opiskelijanumerosi.

Vastauksessa olet vastaavasi sellaisen ihmisen esittämään kysymykseen, joka tuntee kohdalaisen hyvin ohjelmistotekniikan aihealuetta muntoin 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 täysiä pisteitä ei voi saada jos kaikkiin kysytyihin asioihin ei ole vastattu. Jos vastaus vaatii ohjelmakoodin kirjoittamista, sen ei tarvitse olla pilkulleen syntaksiltaan oikein. Mikä tahansa johdonmukaisesti käytetty ja yleisessä käytössä olevia ohjelmointirakenteita sisältävä koodin esitysmuoto käy.

1. [4 pistettä] Suoritussäikeellä (thread) on omaa tilaa ja yhteistä tilaa. Mitä nämä tarkoittavat ohjelmoijan näkökulmasta katsottuna?
2. Kerro lyhyesti mitä seuraavat asiat ovat? 8p.
  - (a) [1 piste] Suoritussäikeen vuoronnus (scheduling)
  - (b) [1 piste] Rinnakkaisen ohjelman lukkiutuminen (deadlock)
  - (c) [2 pistettä] Ohjelmointikirjaston rutiinin säieturvallisuus.
  - (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?
3. Semafori ohjelmointikielissä. 7p.
  - (a) [1 piste] Minkälainen rakenne semafori on ohjelmoijan näkökulmasta?
  - (b) [1 piste] Miten semafori pitää huolen poissulkemisesta?
  - (c) [1 piste] Miten semaforin avulla voi odottaa toisen säikeen tehtävän valmistumista (synkronointi)?
  - (d) [4 pistettä] Miten semafori voidaan toteuttaa käyttöjärjestelmän koodissa?