

TIE-02500 Rinnakkaisuus

Tentti 14.9.2017

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

Laskimen käyttö on kiellettyä.

Muista kirjoittaa kaikkiin vastauspapereihin nimesi ja opiskelijanumerosi.

Vastauksessa olet vastaavasi sellaisen ihmisen esittämään kysymyksen, 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 täysiä pisteitä ei voi saada jos kaikkiin kysytyihin asioihin ei ole vastattu. Jos vastaus vaatii ohjelmakoodin kirjoittamista, sen ei tarvitse olla pilkkulleen syntaksiltaan oikein. Mikä tahansa johdonmukaisesti käytetty ja yleisessä käytössä olevia ohjelmointirakenteita sisältävä koodin esitysmuoto käy.

```
1 namespace moduuli {
2     unsigned int Countdown = 42;
3
4     void Vahenna() {
5         unsigned int TyoKopio = Countdown;
6         TyoKopio = TyoKopio - 1;
7         Countdown = TyoKopio;
8     }
9 }
```

Ohjelma 1: Vähennysohjelma

1. Ohjelman 1 funktiota `Vahenna()` kutsuu ohjelman käynnistyksen jälkeen kaksi suoritusäiettä. 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 suoritusäikeiden 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?

13p.

- (a) [1 piste] Miksi rinnakkaisuus on tällä hetkellä ohjelmoinnissa tärkeä aihepiiri (ts. miksi jokaisen ohjelmoijan tulisi osata rinnakkaista ohjelmointia)?
- (b) [1 piste] Suoritussäikeen vuoronousu (scheduling)
- (c) [1 piste] Suoritussäikeen nälkiintyminen (starvation)
- (d) [1 piste] Mitä on ohjelmointikirjaston säieturvallisuus (thread safety)?
- (e) [1 piste] Mikä on aktiivinen olio (active object)?
- (f) [1 piste] Minkälainen rinnakkaisuuden hallintamekanismi on futuuri (future) ohjelmointikielissä?
- (g) [2 pistettä] Atominen konekäsky. Mihin niitä käytetään? Anna esimerkki tällaisesta käskystä (selosta myös mitä käsky tekee – pelkkä nimi ei riitä)
- (h) [2 pistettä] Minkälainen operaatio on Barrier-synkronointi?
- (i) [3 pistettä] Semaforilla voi luonnin yhteydessä olla alkuarvona esimerkiksi nolla, yksi tai viisi. Mihin käyttötarkoitukseen kullakin alkuarvolla alustettua semaforia voi käyttää (mikä on oletettu käyttötarkoitus arvon perusteella)?

3. [6 pistettä] Mikä on ehtomuuttuja?

Mitä oheinen ohjelmakoodi (ohjelma 2) tekee? Missä koodissa ilmenee synkronoituminen toisen säikeen kanssa? Miksi `pthread_cond_wait()` -kutsu on `while`-silmukassa?

```
1 struct JononKoko {
2     pthread_mutex_t mutex;
3     pthread_cond_t cond;
4     int val;
5 };
6
7 JononKoko x = { PTHREAD_MUTEX_INITIALIZER,
8                PTHREAD_COND_INITIALIZER, 0 };
9
10 void varaaPaikkaJonossa() {
11     if( pthread_mutex_lock( &x.mutex ) != 0 ) {
12         perror("mutex_lock");
13         exit(1);
14     }
15     x.val--;
16     while( x.val < 0 )
17         if( pthread_cond_wait( &x.cond, &x.mutex ) != 0 ) {
18             perror("cond_wait");
19             exit(1);
20         }
21     /* ... muuta koodia ... */
```

Ohjelma 2: Ehtomuuttuja