

581305-6 Tietokoneen toiminta, 5 op, koe 31.1.2018

Kirjoita **jokaiseen** vastauspaperiisi: oma nimi ja nimikirjoitus, opiskelijanumero ja kurssin nimi. Kuhunkin tehtävään riittää noin 1-2 sivun vastaus. Valitse yksi seuraavista vaihtoehtoista ja mainitse valintasi koepaperissa. Oletusvalinta on (c). Huomaa, että koepaperi on kaksipuolinen.

- (a) Syksyn 2017 luentokurssin minikokeen 4 uusintakuulustelu: tehtävä 4.
(Minikokeita 1-3 ei voi enää uusia)
- (b) Syksyn 2017 luentokurssin kurssikokeen uusintakuulustelu: tehtävät 1-4.
(Tässä uusitaan koko kurssikoe. Vastaa kaikkiin tehtäviin 1-4.)
- (c) Tämä on tavallinen erilliskoe ja kattaa koko kurssin: tehtävät 2-5.

1. [9 p] Konekäskyjen suoritus.

- a. [5 p] Oletetaan, että konekäskyn suorituksen aikana tapahtuu kokonaislukujen nollalla jako - virhetilanne. Miten laitteisto tietää, että virhetilanne on tapahtunut? Missä vaiheessa ja mihin virhetilanne merkitään tapahtuneeksi? Kuinka laitteisto toimii antaakseen käyttöjärjestelmälle suoritusvuoron virhetilanteen käsittelemiseksi? Mitä käyttöjärjestelmä tekee virheen käsittelyn jälkeen?
- b. [4 p] Mikä on laitteiston etuoikeutettu (privileged, kernel, supervisor) käskyjen suoritustila? Mitä kaikkea etuoikeutetussa suoritustilassa voi tehdä, jota ei voi tehdä tavallisessa (user) suoritustilassa? Kuinka etuoikeutettuun suoritustilaan päästään? Miten laitteisto tietää, onko se etuoikeutetussa vai tavallisessa suoritustilassa? Kuinka etuoikeutetusta suoritustilasta päästään tavalliseen suoritustilaan?

2. [9 p] Tiedon esitys ja tarkistus

- a. [3 p] Mikä on kokonaisluvun +23 16-bittinen kahden komplementin Big-Endian esitysmuoto? Entä saman luvun 16-bittinen yhden komplementin Little-Endian esitysmuoto?
- b. [3 p] Mikä on 32-bittisen Big-Endian IEEE-liukuluvun +23.0 esitysmuoto?
- c. [3 p] Mikä on Hamming-koodi? Kuinka se pääpiirteissään toimii edellisen kohdan 32-bittisen liukuluvun tapauksessa? Milloin sitä kannattaa käyttää? Milloin sitä ei kannata käyttää?

3. [9 p] Ohjelman suoritus järjestelmässä ja I/O:n toteutus

- a. [3 p] Oletetaan, että käyttöjärjestelmä on siirtänyt suoritukseen olleen prosessin P valmis suoritukseen (Ready) jonoon, ja antanut suoritusvuoron prosessille S. Kun P pääsee joskus myöhemmin uudestaan suoritukseen, sen suoritus jatkuu täsmälleen samalla tavalla kuin jos sitä ei olisi keskeytetty lainkaan. Kuinka tämä on täsmälleen toteutettu?
- b. [3 p] Miksi DMA I/O on tehokkaampaa kuin keskeyttävä I/O (indirect I/O, interrupt-driven I/O)? Anna kolme eri syytä.
- c. [3 p] Miten laiteajuri kertoo (laiteohjaimella suorittavalle) laiteohjainprosessille, mitä sen pitäisi tehdä seuraavaksi? Miten DMA I/O:ta toteuttava laiteohjainprosessi voi kertoa laiteajurille, että sen saama tehtävä on tehty, vaikka laiteajuri ei ole edes suorituksessa sillä hetkellä?

4. [9 p] Korkean tason kielestä suoritukseen ja Java

- a. [3 p] Mitä korkean tason kielen (esim. C tai Fortran) käännöksessä tapahtuu? Mikä on käännöksen tulos?
- b. [3 p] Mitä linkityksessä tapahtuu? Mikä on linkityksen tulos?
- c. [3 p] Mitä latauksessa tapahtuu? Mikä on latauksen tulos?

(KÄÄNNÄ!)

5. [9 p] Titokone, TitoTrainer ja ttk-91. Funktio $Max(a, b, c, d)$ palauttaa arvonaan suurimman parametrin arvon. Parametrit a ja b ovat arvoparametreja. Parametrit c ja d ovat viiteparametreja. Esimerkiksi,

$Max(1, -4, 23, 18)$ palauttaa arvon 23. Muuttujat x ja y on määritelty pääohjelmatasolla.

a. [3 p] Toteuta ttk-91 symbolisella konekielellä (funktion Max avulla)

$$x = Max(5, x, 77, y).$$

b. [3 p] Toteuta ttk-91 symbolisella konekielellä (funktion Max avulla)

$$y = Max(x+8, x+8765432, x+8, x-y).$$

c. [3 p] Toteuta ttk-91 symbolisella konekielellä funktio Max .

Noudata suositusten mukaista aliohjelmien (funktioiden) kutsumekanismia.

TTK-91 konekielen käskyt ovat: NOP, STORE, LOAD, IN, OUT, ADD, SUB, MUL, DIV, MOD, AND, IR, XOR, SHL, SHR, COMP, JUMP, JNEG, JZER, JPOS, JNNEG, JNZER, JNPOS, JLES, JEQU, JGRE, JNLES, JNEQU, JNGRE, CALL, EXIT, PUSH, POP, PUSHR, POPR, SVC