

TKT10005 Tietokoneen toiminta, 5 op, koe 31.1.2019

Kirjoita **jokaiseen** vastauspaperiisi: oma nimi ja nimikirjoitus, opiskelijanumero ja kurssin nimi. Kuhunkin tehtävään riittää noin 1-2 sivun vastaus. Ei lisämateriaaleja, ei laskimia. Valitse jompi kumpi vaihtoehdoista (a) ja (b). Mainitse valintasi koepaperissa. Oletusvalinta on (b).

- (a) Syksyn 2018 luentokurssin minikokeiden 1-4 uusintakuulustelut: tehtävät 1, 2, 3 ja/tai 4.
(b) Tämä on tavallinen erilliskoe ja kattaa koko kurssin: kaikki tehtävät 1-4.

1. [9 p] Konekäskyjen suoritus.

- a. [5 p] Minkä vuoksi laitteistossa on keskeytyksiä (poikkeuksia, interrupts, exceptions, faults)? Anna kolme luonteeltaan hyvin eri tyyppistä keskeytystilannetta ja kerro milloin ne tapahtuvat. Kuinka laitteisto havaitsee keskeytyksen ja siirtää kontrollin juuri tuon keskeytyksen käsittelevälle käyttöjärjestelmän osalle? Mitä tapahtuu keskeytyksen käsittelyn jälkeen keskeytyneen ohjelman suoritukselle mainitsemissasi kolmessa eri keskeytystilanteessa?
- b. [4 p] Minkä vuoksi laitteiston etuoikeutettu (privileged, kernel, supervisor) käskyjen suoritustila on olemassa? Mitä kaikkea etuoikeutetussa suoritustilassa voi tehdä, jota ei voi tehdä tavallisessa (user) suoritustilassa? Miten ja milloin suoritustila vaihtuu tavallisesta etuoikeutettuun ja takaisin? Miten laitteisto tietää, onko se etuoikeutetussa vai tavallisessa suoritustilassa?

2. [9 p] Tiedon esitys ja tarkistus

- a. [2 p] Mikä on 16-bittisen kahden komplementin Big-Endian kokonaisluvun -18 esitysmuoto?
- b. [1 p] Mikä on 32-bittisen yhden komplementin Little-Endian kokonaisluvun +18 esitysmuoto?
- c. [3 p] Mikä on 32-bittisen Big-Endian IEEE-liukuluvun -11.25 esitysmuoto?
- d. [3 p] Minkä vuoksi Hamming-koodia käytetään tiedon siirron suojaamiseen muistiväylällä? Montako ylimääräistä johdinta tarvitaan, jos 32-bittisellä muistiväylällä käytetään Hamming-koodia? Kuinka Hamming-koodi pääpiirteissään toimii?

3. [9 p] Ttk-91. Aliohjelma $Init(T, n, val)$ alustaa n -alkioisen taulukon T kaikki alkiot arvoon val . Parametrit n ja val ovat arvoparametreja. Parametri T on viiteparametri. Esimerkiksi, $Init(Tbl, 100, 7654321)$ alustaa 100-alkioisen taulukon Tbl kaikki alkiot arvoon 7654321. Muuttuja x ja 100-alkioinen taulukko Tbl on määritelty pääohjelmatasolla.

- a. [3 p] Toteuta ttk-91 symbolisella konekielellä (aliohjelman $Init$ avulla) 100-alkioisen taulukon Tbl kaikkien alkioiden alustus arvoon 7654321.
- b. [3 p] Toteuta ttk-91 symbolisella konekielellä (aliohjelman $Init$ avulla) 100-alkioisen taulukon Tbl alkioiden 20-29 (alkiot $Tbl[20]$ - $Tbl[29]$) alustus arvoon x .
- c. [3 p] Toteuta ttk-91 symbolisella konekielellä aliohjelma $Init()$.

Noudata suositusten mukaista aliohjelmien (funktioiden) kutsumekanismia.

4. [9 p] Ohjelman suoritus järjestelmässä ja I/O:n toteutus

- a. [3 p] Mikä on prosessin kuvaajassa (PCB, Process Control Block) oleva suoritinympäristö (processor context)? Mihin sitä tarvitaan? Mitä tietoja siinä on? Milloin siellä olevia tietoja päivitetään?
- b. [3 p] Miksi kovalevyjen I/O on yleensä toteutettu DMA I/O:lla eikä keskeyttävällä I/O:lla (indirect I/O, interrupt-driven I/O)? Miksi se on tehokkaampi tapa tehdä I/O:ta?
- c. [3 p] Oletetaan, että DMA I/O:ta toteuttava laiteohjainprosessi on saanut laiteajurilta saamansa tehtävän valmiiksi. Miten laiteohjainprosessi kertoo laiteajurille, että kyseinen tehtävä on tehty, vaikka laiteajuri ei ole edes suorituksessa sillä hetkellä? Mitä laiteajuri tekee sitten?

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

