

## ITKA203 – Käyttöjärjestelmät – tentti 2016 (MALLITENTTI)

Kevään 2016 kurssin luennot, demot, esimerkkiohjelmat / Paavo Nieminen <paavo.j.nieminen@jyu.fi>

**Tentin ja kurssin arvosteluperusteet 2016:** Tentissä on 48 maksimissaan puolen pisteen arvoista kysymyskohtaa. Ne arvotaan tänä vuonna jopa aivan oikealla arvontakoneella tenttikohtaisesti siten, että (1) jokainen kysymys mittaa yhtä tai useampaa kurssille määriteltyä osaamistavoitetta, (2) puolet kysymyksistä vastaavat kurssin ydinainesta eli välttävää perustietoutta, (3) edistyneemmistä aiheista luennoilla mainitut "perustärpit" ovat mukana joka tentissä ja (4) kaikkien kysymysten yksityiskohdat riippuvat arvonnasta. Joka tentissä on siis osittain erilaiset, mutta mahdollisesti osittain samat kysymykset, arvonnasta riippuen. Kysymyspankia on tarkoitus laajentaa tänä vuonna aivan uusilla kysymyksillä. Uusienkin kysymysten yleinen formaatti tulee olemaan jokin nykyisistä (esim. "väittämiä esimerkkikoodista" tai "yhdistä lauseet") ja ne pyrkivät vastaamaan kysymykseen "osaako opiskelija kurssin käytyään todella ... [+osaamistavoitteen teksti]".

Kysymykset pyrkivät olemaan mahdollisuuksien mukaan soveltavia, joten niissä täytyy olla tarkkaavainen ja miettiä asiat läpi. Koodiesimerkit on luettava läpi ja ymmärrettävä jne. Väittämässä ja monivalinnoissa on suunnitelmallisia "ansoja", joihin on tarkoituskin harhauttaa vastaamaan väärin ne, jotka eivät ole kurssin asioita omaksuneet. Varo siis vaaraa. Tarkoitus on kuitenkin, että oikeasta vastauksesta ei ole mitään epäselvää silloin, kun tiedot löytyvät ja niitä osataan soveltaa.

Mikäli tehtävässä ei mainita poikkeuksista, jokainen oikea vastaus tuo 0.5 pistettä. Kyllä/ei -väittämässä sekä muissa *fifty-fifty* -kysymyksissä *väärä vastaus tuo miinuspisteitä -0.375 pistettä*. Tällä tavoin odotusarvoksi arvaamalla tulee asymptoottisesti 25% maksimipisteistä, aivan kuin normaalissa monivalinnassa, jossa joka neljäs arvaus neljän vaihtoehdon välillä on oikein. Tyhjä vastaus tarkoittaa 0.0 pistettä. Monipuolisemmissa kysymyksissä vääriä vastauskombinaatioita on huomattavasti enemmän kuin 50%, joten niissä ei sakkopisteitä tarvita, koska odotusarvo pisteille on arvaamalla joka tapauksessa lähempänä nolaa kuin edes 25% maksimista. (Havainto, jota en viime vuonna tajunnut tehdä, kun ensimmäistä kertaa uudistin pisteytystä; jospa tämä nyt olisi lähtökohtaisesti hiukan parempi).

Arvosana läpäisyrajalla eli 12 pisteellä on 1 ja täysillä pisteillä 5. Pistealue 12-24 jaetaan tasaväleihin neljällä väliarvolla, jotka pyöristetään lähimpään 0.5 pisteen granulariteetilla ilmaistavaan pistemäärään:

Pisteväli	Välin pituus	Arvosana
[0.0, 12.0)	12p	hylätty
[12.0, 14.5)	2.5p	1
[14.5, 16.5)	2.0p	2
[16.5, 19.0)	2.5p	3
[19.0, 21.5)	2.5p	4
[21.5, 24.0]	2.5p	5

Jos pistemäärä osuu täsmälleen rajalle, luetaan arvosana rajan paremmalta puolelta, esim. 19.0 pistettä riittää arvosanaan 4.

Kurssin läpäisy arvosanalla 1 edellyttää **tentistä** vähintään 12 pistettä. Mahdolliset bonuspisteet lasketaan vasta läpäistyn tentin jälkeen. Jotta ensimmäiseen tenttipäivään ja uusintapäiviin osallistuvat opiskelijat olisivat arvostelun osalta keskenään samanarvoisessa asemassa, **bonuspisteitä saa vain takarajaan 2.6.2016 mennessä palautetuista vapaaehtoisista demoista**. Arvosanaa ei siis voi hinkuttaa ylöspäin jälkikäteen. "matti- ja maijamyöhäset" voivat kuitenkin perustellusta syystä palauttaa *pakollisia* demoja aina kesän viimeiseen uusintaan asti; tämä ei käsittääkseni vaikuta arvosteluun mitenkään eriarvoistavasti. Kesäkuun alusta alkaen pa-

lautuksista tulee sopia etukäteen meilillä tai puhelimitse. Demojärjestelmän päivystäminen todennäköisesti harvenee tai loppuu aivan resurssisyiden vuoksi.

**Yleisiä ohjeita:** Muista merkitä vastauspaperiin oma **nimesi** ja **syntymäaikasi** sekä kurssin nimi. Lisäksi **vastauspaperisi tulee sisältää 48 peräkkäistä numeroitua kohtaa**, joissa on joko tehtävässä pyydetty vastaus tai viiva "–" tyhjän vastauksen merkiksi.

Esimerkkeihin perustuvissa tehtävissä oletetaan, että järjestelmässä ei ole yhtäaikaan muita käyttäjiä, prosesseja, vikoja tai muutakaan, jotka muuttaisivat toimintaa siitä, miltä se esimerkissä suoraviivaisesti näyttää.

Moniselitteisiä kysymyksiä ei ole laitettu mukaan tahallisesti. Mikäli jokin tehtävä on vahingossa sellainen, että vastaus ei olekaan yksikäsitteinen, laita vastauspaperiisi tehtävän kohdalle kommentti, jossa kerrot, miksi mielestäsi näin on. Virheellisiksi osoittautuvat kysymykset poistetaan tämän tenttikerran arvostelusta.

## Numeroidut kysymyskohdat 1–48

**Ohje tehtäviin 1–4:** Yhdistä lauseen loppua vastaava kirjain numeroituun alkuun siten, että kukin lause on totta. Alkuihin on yksikäsitteinen oikea loppu. Jokainen loppu voi sopia useampaan alkuun tai ei yhtenkään.

Lauseiden alut:

Vaihtoehtoiset loput:

- |   |   |
|---|---|
| 1. Keskinäinen poissulku ( <i>Mutex</i> ) | A. liittyy tietokoneiden muistihierarkian perusideaan.                      |
| ...                                       | B. liittyy kilpa-ajotilanteiden (engl. <i>race condition</i> ) ratkomiseen. |
| 2. RAID ...                               | C. liittyy tiedon tallennuksen vikasietoisuuteen.                           |
| 3. Semafori ...                           |   |
| 4. Lokaalisuusperiaate ...                |   |

**Ohje tehtäviin 5–8:** Yhdistä lauseen loppua vastaava kirjain numeroituun alkuun siten, että kukin lause on totta. Alkuihin on yksikäsitteinen oikea loppu. Jokainen loppu voi sopia useampaan alkuun tai ei yhtenkään.

Lauseiden alut:

Vaihtoehtoiset loput:

- |  |  |
|--|--|
| 5. I/O -ohjelmisto (engl. <i>I/O subsystem</i> ) ...                       | A. tarvitaan vain virtuaalikoneeseen asennetussa käyttöjärjestelmässä. |
| 6. Virtuaalimuistin hallinta (engl. <i>virtual memory management</i> ) ... | B. määrittelee nimet/osoitteet koneeseen liitetyille laitteille.       |
| 7. IPC (engl. <i>inter-process communication</i> ) ...                     | C. käsittelee sivukeskeytyksen (engl. <i>page fault</i> ).             |
| 8. Graafinen tiedostoselain (engl. <i>file browser</i> ) ...               | D. tarvitaan bittioperaatioiden, kuten AND ja OR, hoitamiseksi.        |
|  | E. toimittaa signaaleja (esim. apuohjelmalla <i>kill</i> lähetettyjä). |
|  | F. ei ole välttämätön osa nykyaikaista käyttöjärjestelmää.             |

**Tutkittava esimerkki tehtäviin 9–10:** Kurssin luennoilta ja demoista tutussa ympäristössä (Linux,bash) tehty yksittäinen, POSIX-syntaksin mukainen komentorivi:

```
arg arg | grep -i "a b c" | sort -f -r | kissa > cat
```

9. Montako komentoa rivillä on yhteensä?
10. Montako argumenttia rivillä on yhteensä?
11. **Väite:** Tietokonejärjestelmän tuottavuutta (engl. *throughput*) eli aikayksikössä aikayksikössä loppuun saatujen tehtävien määrää voidaan kasvattaa lisäämällä kellokeskeytyksien määrää aikayksikössä. (A=kyllä; B=ei)

**Ohje tehtävään 12:** Yhdistä lauseen loppua vastaavat kirjaimet (*vähintään* yksi, mutta *mahdollisesti* useita) lauseenalun perään siten, että muodostuvat lauseet vastaavat todellisuutta. Vastauksessa on oltava listattuna kaikki todellisuutta vastaavat vaihtoehdot.

Lauseen alku:

12. Kurssilla käsitelty käyttöjärjestelmän rajapintastandardi POSIX (vuoden 2008 versio) määrää, että ...

Vaihtoehtoiset loput (mahdollisesti useita sopivia):

- A. kellokeskeytyksen tulee tapahtua 50, 100 tai 1000 kertaa sekunnissa  
B. shell-komento *c99* käynnistää C99-standardin mukaisen C-kääntäjän  
C. komento *startx* käynnistää X-ikkunointijärjestelmän  
D. Avustus (engl. *Help*) on valikkopalkin oikeanpuoleisin valikko
13. **Väite:** POSIX-säikeitä käyttävä ohjelma täytyy kääntää erikseen yksiytimiselle ja moniytimiselle prosessorille, koska sama säikeitä käyttävä koodi ei voi toimia samanlaisena sekä yksiprosessori- että moniprosessorijärjestelmässä. (A=kyllä; B=ei)

**Ohje tehtäviin 14–17:** Yhdistä lauseen loppua vastaava kirjain numeroituun alkuun siten, että kukin lause on totta. Alkuihin on yksikäsitteinen oikea loppu. Jokainen loppu voi sopia useampaan alkuun tai ei yhteenkään.

Lauseiden alut:

14. Prosessitaulu ...  
15. Prosessielementti (PCB) ...  
16. Sivutaulu ...  
17. Ready-jono ...

Vaihtoehtoiset loput:

- A. liittyy useiden eri prosessien käsittelyyn.  
B. on prosessikohtainen.  
C. ei välttämättä liity prosessien tarpeisiin.
18. **Väite:** Ns. laiteriippuvan I/O-ohjelmiston tehtävänä on mm. aloittaa I/O-laitteelta saapuvan keskeytyksen käsittely. (A=kyllä; B=ei)  
19. **Väite:** Laiteriippuva I/O-ohjelmisto määrittelee laitteille nimeämiskäytännön ja rajapinnan. (A=kyllä; B=ei)  
20. **Väite:** I/O-operaation pyytäminen voi johtaa pyytävän prosessin siirtämiseen blocked-tilaan. (A=kyllä; B=ei)

**Tutkittava esimerkki tehtäviin 21–22:** Kurssin luennoilta ja demoista tutussa ympäristössä (Linux,bash) tehtyjä tuttuja komentoja ja niiden tulosteita (merkistökoodaus on UTF-8):

```
[nieminen@halava esimerkki]$ whoami
nieminen
[nieminen@halava esimerkki]$ ls -la
total 600
drwx-----. 2 nieminen nobody  1024 May 19 11:18 .
drwx-----. 9 nieminen nobody  1024 May 19 11:16 ..
-rwxr--r--. 1 nieminen nobody 586405 May 19 11:20 kayttojarjestelmat.tex
[nieminen@halava esimerkki]$ echo Heippa >> kayttojarjestelmat.tex
[nieminen@halava esimerkki]$
```

21. **Väite:** komentojen jälkeen tiedoston *kayttojarjestelmat.tex* pituus on pienempi kuin 586405 tavua. (A=kyllä; B=ei)  
22. **Väite:** komentojen jälkeen annettava komento *echo ./kayttojarjestelmat.tex* tulostaisi konsoliin "Heippa" (A=kyllä; B=ei)  
23. **Väite:** Historiallisesti merkittävän ENIAC-tietokoneen (valmistettu vuonna 1946) mukana toimitettiin käyttöjärjestelmä nimeltä Multics, jossa oli jo mukana monia nykyisten käyttöjärjestelmien ominaisuuksia. (A=kyllä; B=ei)  
24. **Väite:** Prosessin konteksti (rekisterien tila) on sen kaikille säikeille yhteinen. (A=kyllä; B=ei)  
25. **Väite:** POSIXin semafori on tietorakenne, joka sisältää etumerkittömän kokonaisluvun ja joukon prosesseja tai säikeitä. (A=kyllä; B=ei)  
26. **Väite:** POSIXin semaforikutsun *sem\_wait(s)* suorituksessa semaforin *s* arvo muuttuu aina. (A=kyllä; B=ei)

**Tutkittava esimerkki tehtäviin 27–29:** luento-esimerkeistä tuttua C-kielistä ohjelmanpätkeä muistuttava koodi (POSIXin pthread-kirjastoa hyödyntävä)

```
#define N 1000
pthread_mutex_t mymutex = PTHREAD_MUTEX_INITIALIZER;
uint64_t summa = 0;
void * saikeen_koodi(void *v) {
    int i;
    for (i = 1; i <= N; i++){
        pthread_mutex_lock(&mymutex);
        summa++;
    }
    return NULL;
}

int main(int argc, char *argv[]) {
    pthread_t saieA, saieB;
    pthread_create(&saieA, NULL, saikeen_koodi, NULL);
    pthread_create(&saieB, NULL, saikeen_koodi, NULL);
    pthread_join(saieA, NULL);
    pthread_join(saieB, NULL);
    if (summa==2000){
        return 0; // homma toimi
    } else {
        return 1; // homma ei toiminut
    }
}
```

27. **Väite:** Esimerkin main() palauttaa aina 0:n (A=kyllä; B=ei)
28. **Väite:** Esimerkin ohjelma ratkaisee kuluttaja-tuottaja -ongelman kahdella säikeellä (A=kyllä; B=ei)
29. **Väite:** Esimerkin ohjelma voi aiheuttaa lukkiutumistilanteen (engl. *deadlock*), vaikka sen aliohjelmakutsut onnistuisivat ilman ongelmia (A=kyllä; B=ei)
30. **Väite:** Microkernel -tyyppinen käyttöjärjestelmä pyrkii suorittamaan palveluita mahdollisimman paljon käyttöjärjestelmätilassa (engl. *kernel mode*). (A=kyllä; B=ei)

**Tutkittava esimerkki tehtäviin 31–32:** C-mäisenä pseudokoodina tehty ehdotus rengaspuskuria käyttävän Kuluttaja-tuottaja -ongelman ratkaisemiseksi semaforipalvelua käyttäen. Yhteistä muistia käytetään vain puskuriooperaatioissa.

```

tuottaja(){
    while(true) { // tuotetaan loputtomiin
        tuota(); // tehdään yksi datapätkä
        wait(EMPTY);
        wait(MUTEX);
        siirra_puskuriin();
        signal(MUTEX);
        signal(EMPTY);
    }
}

kuluttaja(){
    while(true) { // kulutetaan loputtomiin
        wait(FULL);
        wait(MUTEX);
        lue_puskurista();
        signal(MUTEX);
        signal(FULL);
        kuluta(); // käytetään yksi datapätkä
    }
}

main(){
    EMPTY=tee_semafori( PUSKURIN_KOKO );
    FULL=tee_semafori( 0 );
    MUTEX=tee_semafori( 1 );
    kaynnista_saie(tuottaja);
    kaynnista_saie(kuluttaja);
}

```

31. **Väite:** Esitetty koodi ratkaisee kuluttaja-tuottaja -ongelman oikeellisesti (A=kyllä; B=ei)  
32. [Edellinen kohta on poikkeuksellisesti 1.0p arvoinen; tämä kohta jää tyhjäksi]

**Tutkittava esimerkki tehtäviin 33–35:** luennolla ja monisteessa esitetyn kaltainen minimalistinen shell-ohjelma (kommentit poistettu, rivit numeroitu, näytetty vain olennainen toimintopätkä, mahdollisesti "rikkottu" jollain selkeällä tavalla tai sitten ei):

```

1  while(true){
2      luekomento(komento, argumentit);
3      pid = fork();
4      if (pid > 0) {
5          status = wait();
6      } else if (pid == -1) {
7          ilmoita("fork() epäonnistui!");
8          exit(1);
9      } else {
10         exec(komento, argumentit);
11         ilmoita("Komentoa ei voitu suorittaa!");
12         exit(1);
13     }
14 }

```

33. **Väite:** Rivi 5 tapahtuu fork() -kutsun luomassa lapsiprosessissa. (A=kyllä; B=ei)  
34. **Väite:** Rivin 8 suorittaminen lopettaa koko minimalistisen shellin suorituksen. (A=kyllä; B=ei)  
35. **Väite:** Rivin 12 suorittaminen lopettaa koko minimalistisen shellin suorituksen. (A=kyllä; B=ei)

**Tutkittava esimerkki tehtäviin 36–39:** Kuvitellaan, että meillä on käytössä yksinkertainen tietokone, jonka virtuaalimuistiosoitteissa on 20 bittiä, joista ensimmäiset 8 ilmoittavat sivunumeron ja loput 12 ilmoittavat tavuindeksin sivun sisällä. Fyysiset muistiosoitteet ovat 24-bittisiä. Keskusmuistista on varattu prosessien käyttöön tasan 8 kehystä fyysisissä osoitteissa 0x100000-0x107000. Virtuaalimuistin kokonaiskoko on 0x1000000 tavua (n. 16 megatavua). Tietorakenteiden tilanne tarkasteluhetkellä on seuraava. Prosessien sivutauluista näytetään vain kartoitetut osat.

Prosessin (PID=2) sivutaulun kartoitetut rivit:

virt. sivu	fyys. sivu	muist. (P)	dirty (D)	diskIndex
0x02	0x100	1	0	0x0010
0x03	0x000	0	0	0x0022
0x7f	0x101	1	1	0x00bc

Prosessin (PID=7) sivutaulun kartoitetut rivit:

virt. sivu	fyys. sivu	muist. (P)	dirty (D)	diskIndex
0x02	0x000	0	0	0x0004
0x03	0x102	1	0	0x0008
0x04	0x000	0	0	0x00f2
0x2e	0x104	1	1	0x006c
0x7f	0x107	1	1	0x0007

Järjestelmän kehystaulu:

fyys. sivu	omistajan PID	omistajan PTE#	aikayksiköt käyttökerran jälkeen	edellisen jälkeen
0x100	2	0x02	19	
0x101	2	0x7f	155	
0x102	7	0x03	199	
0x103	234	0x45	8	
0x104	7	0x2e	12	
0x105	876	0x45	4	
0x106	14	0x5e	8	
0x107	7	0x7f	9	

36. Mihin fyysiseen muistiosoitteeseen kohdistuisi prosessin 2 tekemä kirjoitus virtuaalimuistiosoitteeseen 0x7f123?
37. Mihin fyysiseen muistiosoitteeseen kohdistuisi prosessin 7 tekemä luku virtuaalimuistiosoitteesta 0x7f123?
38. Prosessi 2 suorittaa hyppykäskyn aliohjelmaan muistiosoitteessa 0x02345. Tapahtuuko prosessorissa sivuvirhe / sivunvaihtokeskeytys? (A=kyllä; B=ei)
39. Prosessissa 7 aiheutuu sivunvaihtokeskeytys. Korvausalgoritmi on LRU. Onko jonkin sivun sisältö tallennettava levyille? (A=kyllä; B=ei)
40. **Väite:** Unixin i-solmuihin (engl. *inode*) perustuvassa tiedostojärjestelmässä tiedoston nimi sisältyy tiedoston i-solmuun. (A=kyllä; B=ei)
41. **Väite:** Unixin i-solmuihin (engl. *inode*) perustuvassa tiedostojärjestelmässä tiedoston viimeisin muutos aika sisältyy tiedoston i-solmuun. (A=kyllä; B=ei)

**Tutkittava esimerkki tehtäviin 42–43:** Kurssin esimerkeistä tuttua Linuxille käännettyä C-ohjelmaa ajetaan x86-64 -arkkitehtuurilla, ja debuggerilla on nähtävissä seuraavat hetkelliset tiedot

Rekistereitä:

```
RIP (käsky) 0x0000000000400504
RSP (huippu) 0x00007fffffffddcc0
RBP (kanta) 0x00007fffffffddcf0
```

Muistin sisältöä (kaikki muuttujat 64-bittisiä eli 8-tavuisia):

```
0x7fffffffdd08: 0x0000000100000000
0x7fffffffdd00: 0x00007fffffffde38
0x7fffffffddcf8: 0x0000000000400647
kanta --> 0x7fffffffddcf0: 0x00007fffffffdd50
0x7fffffffddce8: 0x0000000000000001
0x7fffffffddce0: 0x0000000000000000
0x7fffffffddcd8: 0x0000000000000000
0x7fffffffddcd0: 0x0000000000000000
0x7fffffffddcc8: 0x0000000000000000
huippu --> 0x7fffffffddcc0: 0x0000000000000000
```

42. Mikä on kutsupinossa nykyistä aktivaatiota *edeltävän* aktivaation pinokehyksen kannan osoite? Ilmoita heksalukuna.
43. Mikä tulee olemaan RSP-rekisterin sisältö siinä vaiheessa, kun tämä meneillään oleva aliohjelmakäyttö on jossain vaiheessa loppunut ja siihen sisältyvä käsky `ret` on viimeisimpänä suoritettu.

**Tutkittava esimerkki tehtäviin 44–46:** Kurssin luennoilta tutulla symbolisella konekielellä (GNU assembler) kirjoitettu, rivi riviltä kommentoitu ohjelmanpätkä:

```
_start:
    movq    $4,%rcx    # sijoita luku 4 rekisteriin RCX
    movq    $0,%rdi    # sijoita luku 0 rekisteriin RDI
silimu:
    addq    %rcx,%rdi # laske sisältöjen summa RCX + RDI
                    # ja sijoita tulos rekisteriin RDI
    dec    %rcx      # vähennä 1 rekisterin RCX arvosta
    jnz    silimu    # hyppää, jos edellisen käskyn tulos ei ollut 0
```

44. Kuinka monta konekielikäskyä ohjelmanpätkän jälki sisältää, ts. kuinka monta käskyä sen suorittamisen alusta loppuun vaatii?
45. Mikä on rekisterin RCX sisältö ohjelmanpätkän suorituksen lopuksi?
46. Mikä on rekisterin RDI sisältö ohjelmanpätkän suorituksen lopuksi?

**Ohje tehtävään 47:** Yhdistä lauseen loppua vastaavat kirjaimet (*vähintään* yksi, mutta *mahdollisesti* useita) lauseenalun perään siten, että muodostuvat lauseet vastaavat todellisuutta. Vastauksessa on oltava listattuna kaikki todellisuutta vastaavat vaihtoehdot.

Lauseen alku:

47. Prosessorin keskeytys ...

Vaihtoehtoiset loput (mahdollisesti useita sopivia):

- A. on yhtä nopea toimenpide kuin aliohjelmakutsuun siirtymisen (esim. AMD64:n käsky `call`)
  - B. toimii vain moniydinprosessorissa
  - C. ei voi aiheutua käyttäjätilassa toimivan prosessin toimenpiteiden johdosta
  - D. mahdollistaa monen ohjelman yhdenaikaisen suorittamisen
48. **Väite:** RAID 1 -järjestelmä perustuu tiedon peilaamiseen identtisenä useammalle kovalevylle (varma fakta; varsinainen väite seuraa). Tämä parantaa toimintavarmuutta, mutta erityisesti tiedon lukeminen

RAID 1 -levyjärjestelmästä on hitaampaa kuin yksittäistä kovalevyä käyttämällä. (A=kyllä; B=ei)

### **Vapaaehtoinen vapaa sana**

Halutessasi voit antaa loppuun palautetta tai kommentoida muuten kurssia tai tenttiä. Vastauksen muoto on vapaa, eikä se vaikuta arvosteluun.