

## ITKA203 – Käyttöjärjestelmät – tentti 10.6.2016

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

**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. Mikäli tehtävässä ei mainita poikkeuksista, jokainen oikea vastaus tuo 0.5 pistettä. *Kyllä/ei -väittämissä sekä muissa kysymyksissä, joissa on kaksi vaihtoehtoa (A tai B), väärä vastaus tuo miinus pisteitä -0.375 pistettä*, jotta odotusarvoksi täydellä arvaamisella tulee selkeästi hylätty pistemäärä.

Esimerkkeihin perustuvissa tehtävissä oletetaan, että järjestelmässä ei ole yhtäaikaan muita käyttäjiä, prosesseja, vikoja tai muutakaan, jotka muuttaisivat toimintaa siitä, miltä se esimerkissä suoraviivaisesti näyttää. Oletetaan myös, että kaikki käyttöjärjestelmä- ja alustakirjastokutsut toimivat ilman poikkeuksia tai virheitä.

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 ja muokataan kysymyspankissa yksiselitteisempään muotoon tulevaisuutta varten.

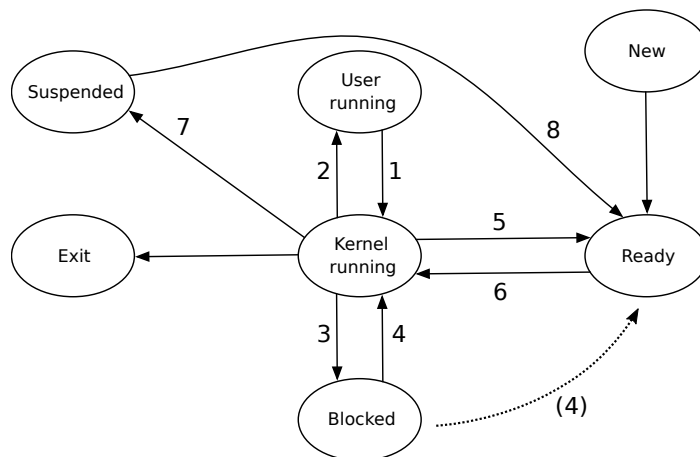
### Numeroidut kysymyskohdat 1–48

**Tutkittava esimerkki tehtäviin 1–2:** Kurssin luennoilta ja demoista tutussa ympäristössä (Linux,bash) tehty yksittäinen, POSIX-syntaksin mukainen komentorivi:

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

1. Montako komentoa rivillä on yhteensä?
2. Montako argumenttia rivillä on yhteensä?

**Ohje tehtäviin 3–5:** 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. Tehtävässä tutkitaan prosessien tilasiirtymäkaaviota, jonka selitetekstit on korvattu numeroilla:



Lauseiden alut:

3. Tilasiirtymä nro 1 tapahtuu, ...
4. Tilasiirtymä nro 5 tapahtuu, ...
5. Tilasiirtymä nro 6 tapahtuu, ...

Vaihtoehtoiset loput:

- A. kun käyttöjärjestelmän vuorontaja siirtää suoritukseen toisen prosessin, vaikka nykyinenkin prosessi pystyisi jatkamaan laskemista jo seuraavassa konekielikäskyssään.
- B. kun prosessi aloittaa käyttöjärjestelmän vuorontajalta saamansa aikaikkunan käytön.
- C. kun prosessille on lähetetty odottelusignaali esim. näppäilemällä pääteyhdydessä Ctrl-Z.
- D. kun prosessori siirtyy keskeytyksen tai käyttöjärjestelmäksun käsittelyyn.

**Ohje tehtäviin 6–7:** 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.

Lauseiden alut:

6. Laiteriippuva I/O -ohjelmisto ...
7. Laitteistoriippumaton I/O -ohjelmisto ...

Vaihtoehtoiset loput (mahdollisesti useita sopivia):

- A. voi tulla suoritukseen sovellusohjelman pyynnön seurauksena.
- B. sisältää laiteohjainten ajurit.
- C. voi tulla suoritukseen prosessorin keskeytyksen seurauksena.
- D. määrittää järjestelmälle yhtenäisen lohkokoon.

8. **Väite:** Prosessin kaikilla säikeillä on aina sama sisältö jokaisessa prosessorin rekisterissä (esim. AMD64:n RIP:ssä ja RAX:ssä) (A=kyllä; B=ei)

**Ohje tehtävään 9:** 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:

9. 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

**Ohje tehtäviin 10–13:** 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:

10. Prosessitaulu ...
11. Prosessielementti (PCB) ...
12. Sivutaulu ...
13. Ready-jono ...

Vaihtoehtoiset loput:

- A. liittyy useiden eri prosessien käsittelyyn.
- B. on prosessikohtainen.
- C. ei välttämättä liity prosessien tarpeisiin.

14. **Väite:** Tietokonejärjestelmän tuottavuutta (engl. *throughput*) eli aikayksikössä loppuun saatujen tehtävien määrää voidaan kasvattaa lisäämällä kellokeskeytyksien määrää aikayksikössä. (A=kyllä; B=ei)

**Ohje tehtäviin 15–18:** 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:

15. Keskinäinen poissulku (engl. *mutual exclusion, Mutex*) ...
16. FLIH (*first-level interrupt handling*) ...
17. Semafori (*semaphore*) ...
18. IP-rekisteri (*instruction pointer*) ...

Vaihtoehtoiset loput:

- A. on osa prosessorilaitteiston toimintaa.
- B. liittyy tietokoneiden muistihierarkian perusideaan.
- C. on tietoverkkoon liitetyn laitteen osoite.
- D. liittyy kilpa-ajotilanteiden (engl. *race condition*) ratkomiseen.

**Tutkittava esimerkki tehtäviin 19–20:** Kurssin luennoilta ja demoista tutussa ympäristössä (Linux,bash) tehtyjä tuttuja komentoja ja niiden tulosteita (merkistökoodaus on UTF-8):

```
[nieminen@halava tenttikys]$ whoami
nieminen
[nieminen@halava tenttikys]$ ls -l
total 16
-rw-r--r--. 1 nieminen users 10740 May 17 22:34 kayttojarjestelmat.tex
-rw-r--r--. 1 nieminen users 4 May 17 22:35 moi
[nieminen@halava tenttikys]$ cat moi > kayttojarjestelmat.tex
[nieminen@halava tenttikys]$
```

19. **Väite:** komentojen jälkeen tiedoston `kayttojarjestelmat.tex` pituus on pienempi kuin 10740 tavua. (A=kyllä; B=ei)
20. **Väite:** komentojen jälkeen annettava komento `cat moi` tulostaisi konsoliin tekstin "kayttojarjestelmat.tex" ja rivinvaihdon. (A=kyllä; B=ei)

**Ohje tehtävään 21:** Järjestä seuraavat muistikomponentit niiden nopeuden mukaan: A=kovalevy, B=keskusmuisti, C=välimuisti, D=rekisteri.

21. Vastauksessasi on neljä järjestettyä kirjainta: nopein ensin, hitain viimeisenä.
22. **Väite:** Pää tavoite ohjelmistokerrosten ja niiden välisten rajapintojen määrittelyssä on estää kilpailijoita toteuttamasta tuotteita, jotka toimivat samalla tavoin kuin valmistajan oma tuote. (A=kyllä; B=ei)

**Ohje tehtävään 23:** 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:

23. Prosessorin keskeytys ...

Vaihtoehtoiset loput (mahdollisesti useita sopivia):

- A. toimii vain moniydinprosessorissa.
- B. on yhtä nopea toimenpide kuin aliohjelmakutsuun siirtyminen (esim. AMD64:n käskyn `call` suorittaminen).
- C. voi aiheutua käyttäjätilassa toimivan prosessin toimenpiteiden johdosta.
- D. voidaan estää sovellusohjelmassa käyttämällä synkronointia.

24. **Väite:** Tyypillisen käyttöjärjestelmän lähdekoodin määrä on ollut laskusuhdanteessa 1970-luvulta alkaen, johtuen kyseisen vuosikymmenen merkittävästä keksinnöistä koodin rakenteeseen liittyen. (A=kyllä; B=ei)

**Ohje tehtäviin 25–28:** 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:

25. I/O -ohjelmisto (engl. *I/O subsystem*) ...
26. Käyttöjärjestelmän virtuaali-muistin hallintaosio (engl. *virtual memory management*) ...
27. IPC (engl. *inter-process communication*) ...
28. Graafinen tiedostoselain (engl. *file browser*) ...

Vaihtoehtoiset loput:

- A. ei ole välttämätön osa nykyaikaista käyttöjärjestelmää.
  - B. käsittelee sivukeskeytyksen (engl. *page fault*).
  - C. toimittaa signaaleja (esim. apuohjelmalla *kill* lähetettyjä).
  - D. määrittelee nimet/osoitteet koneeseen liitetuille laitteille.
  - E. tarvitaan vain virtuaalikoneeseen asennetussa käyttöjärjestelmässä.
  - F. tarvitaan bittioperaatioiden, kuten AND ja OR, hoitamiseksi.
29. **Väite:** POSIXin semafori on binäärinen lippumuuttuja. (A=kyllä; B=ei)
  30. **Väite:** POSIXin semaforikutsun `sem_post(s)` suorituksessa semaforin `s` arvo muuttuu aina. (A=kyllä; B=ei)

**Tutkittava esimerkki tehtäviin 31–33:** Luento-esimerkeistä tuttua C-kielistä ohjelmanpätkää muistuttava koodi (POSIXin pthread-kirjastoa hyödyntävä; mahdollisesti "rikottu" jollain selkeällä tavalla tai sitten ei). Oletetaan tehtävässä, että kaikki kutsut aina onnistuvat, eikä mitään ulkopuolisia vaikutuksia ole:

```
#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++){
        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
    }
}
```

31. **Väite:** Esimerkin `main()` palauttaa aina 0:n, mikäli sen suoritus ylipäätään onnistuu loppuun saakka ilman esteitä ja kaikki sen sisältämät aliohjelmakutsut onnistuvat ilman virhekoodia. (A=kyllä; B=ei)
32. **Väite:** Esimerkissä on periaatteessa mahdollista tilanne, jossa säikeessä A muuttuja `i==123` ja säikeessä B muuttuja `i==456` samaan aikaan. (A=kyllä; B=ei)
33. **Väite:** Esimerkin ohjelma voi aiheuttaa lukkiutumistilanteen (engl. *deadlock*), jossa sen suoritus jää jumiin. (A=kyllä; B=ei)
34. **Väite:** P.J. Denningin ym. 1970-luvulla kehittelemä lokaalisuusperiaate (engl. *principle of locality*) on pohjana sille, että konekieliset aliohjelmat voivat kutsua itseään rekursiivisesti. (A=kyllä; B=ei)

**Tutkittava esimerkki tehtäviin 35–36:** C-mäisenä pseudokoodina tehty ehdotus rengaspuskuriä käyttävän tuottaja-kuluttaja -ongelman ratkaisemiseksi POSIX-tyyppistä semaforipalvelua käyttäen. Yhteistä muistia käytetään puskuriopeeraatioissa. Semafori EMPTY mallintaa rengaspuskurin vapaata, kirjoitettavissa olevaa tilaa ja FULL mallintaa kirjoitettua mutta toistaiseksi käsittelemätöntä tilaa.

```
tuottaja() {
    while(true) { // tuotetaan loputtomiin
        tuota(); // tehdään yksi datapätkä
        sem_wait(EMPTY);
        sem_wait(MUTEX);
        siirra_puskuriin();
        sem_post(MUTEX);
        sem_post(EMPTY);
    }
}

kuluttaja() {
    while(true) { // kulutetaan loputtomiin
        sem_wait(FULL);
        sem_wait(MUTEX);
        lue_puskurista();
        sem_post(MUTEX);
        sem_post(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);
}
```

35. **Väite:** Semaforien alustus on oikeellinen ongelman ratkaisemiseksi. (A=kyllä; B=ei)  
36. **Väite:** Semaforien käyttö säikeissä on oikeellinen ongelman ratkaisemiseksi. (A=kyllä; B=ei)

**Tutkittava esimerkki tehtäviin 37–39:** luennolla ja monisteessa esitetyn kaltainen minimalistinen shell-ohjelma (kommentit poistettu, rivit numeroitu, näytetty vain olennainen toimintopätkä):

```
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 }
```

37. **Väite:** Rivi 10 tapahtuu fork() -kutsun luomassa lapsiprosessissa. (A=kyllä; B=ei)  
38. **Väite:** Rivin 8 suorittaminen lopettaa koko minimalistisen shellin suorituksen. (A=kyllä; B=ei)

39. **Väite:** Rivin 12 suorittaminen lopettaa koko minimalistisen shellin suorituksen. (A=kyllä; B=ei)

**Tutkittava esimerkki tehtäviin 40–43:** 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 fyysistä muistia osoitteissa 0x100000-0x107fff. Virtuaalimuistin kokonaiskoko on 0x1000000 tavua (n. 16 megatavua). Tietorakenteiden tilanne tarkasteluhetkellä on seuraava. Prosessien sivutauluista näytetään vain kartoitetut osat kahden prosessin (PID:t 2 ja 7) osalta.

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
0x2e	0x106	1	0	0x00bb
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	1	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 edellisen käyttökerran jälkeen
0x100	2	0x02	19
0x101	2	0x7f	150
0x102	7	0x03	16
0x103	234	0x45	8
0x104	7	0x2e	12
0x105	876	0x45	4
0x106	2	0x2e	175
0x107	7	0x7f	9

40. Mihin fyysiseen muistiosoitteeseen kohdistuisi prosessin 7 tekemä kirjoitus virtuaalimuistiosoitteeseen 0x7f123?
41. Mistä virtuaaliosoitteesta prosessi 7 saisi käyttöönsä tavun, joka sijaitsee kovalevyllä indeksissä 0xf2345 heittovaihto-osion / -tiedoston (engl. *swap space*) alusta lukien?
42. Prosessi 2 suorittaa hyppykäskyn aliohjelmaan muistiosoitteessa 0x2e07f. Tapahtuuko prosessorissa sivuvirhe / sivunvaihtokeskeytys? (A=kyllä; B=ei)
43. Prosessissa 876 aiheutuu sivunvaihtokeskeytys. Korvausalgoritmi on LRU. Onko jonkin sivun sisältö tallennettava levyllä? (A=kyllä; B=ei)

**Tutkittava esimerkki tehtäviin 44–46:** Kurssin luennoilta tutulla symbolisella konekielellä (GNU assembler; AT&T -syntaksi) kirjoitettu, rivi riviltä kommentoitu ohjelmanpätkä:

```
_start:
    movq    $3,%rcx    # luku 3 rekisteriin RCX
    movq    $6,%rdi    # luku 6 rekisteriin RDI
silimu:
    inc     %rdi       # lisää 1 rekisterin RDI arvoon
    dec     %rcx       # vähennä 1 rekisterin RCX arvosta
    jz     ulos        # hyppää, jos edellisen käskyn tulos oli 0
    jmp    silimu      # hyppää aina
ulos:
    dec     %rdi       # vähennä 1 rekisterin RDI arvosta
```

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

**Tutkittava esimerkki tehtäviin 47–48:** Kurssin esimerkeistä tuttua Linuxille käännettyä C-ohjelmaa ajetaan x86-64 -arkkitehtuurilla, ja debuggerilla on nähtävissä seuraavat hetkelliset tiedot

Rekistereitä:

```
RIP (käsky) 0x000000000400504
RSP (huippu) 0x00007fffffffddcc0
RBP (kanta) 0x00007fffffffdcf0
```

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

```
0x7fffffffdd08: 0x0000000100000000
0x7fffffffdd00: 0x00007fffffffde38
0x7fffffffdcf8: 0x000000000400647
kanta --> 0x7fffffffdcf0: 0x00007fffffffdd50
0x7fffffffddce8: 0x0000000000000001
0x7fffffffddce0: 0x0000000000000000
0x7fffffffddcd8: 0x0000000000000000
0x7fffffffddcd0: 0x0000000000000000
0x7fffffffddcc8: 0x0000000000000000
huippu --> 0x7fffffffddcc0: 0x0000000000000000
```

47. Kuinka monta tavua muistia on varattu nykyisen aktivaation väliaikaisten muuttujien (eli. lokaalit muuttujat ja tarvittavat paikalliset kopiot parametreista) säilyttämistä varten? Ilmoita *kymmenjärjestelmän* lukuna.
48. Mikä tulee olemaan RIP-rekisterin sisältö siinä vaiheessa, kun tämä meneillään oleva aliohjelmanaktivaatio on jossain vaiheessa loppunut ja siihen sisältyvä käsky `ret` on viimeisimpänä suoritettu.

## Vapaaehtoinen vapaa sana

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