

ITKA203 – Käyttöjärjestelmät – tentti 19.5.2017

Kevään 2017 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. 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

Ohje tehtävään 1: Järjestä seuraavat muistikomponentit niiden kapasiteetin mukaan, eli mihin mahtuu eniten dataa kerralla: A=kovalevy, B=keskusmuisti, C=välimuisti, D=rekisteri.

1. Vastauksessasi on neljä järjestettyä kirjainta: isoin muisti ensin, pienin muisti viimeisenä.

Ohje tehtäviin 2–5: Yhdistä lauseen loppua vastaava kirjain numeroituun alkuun siten, että lause on totta. Alkuihin on yksikäsitteinen oikea loppu. Jokainen loppu voi sopia useampaan alkuun tai ei yhteenkään.

Lauseiden alut:

Vaihtoehtoiset loput:

- | | |
|--|---|
| 2. Käyttöjärjestelmän virtuaali-
muistin hallintaosio (engl. <i>virtual
memory management</i>) ... | A. tekee toimenpiteitä jokaisen kellokeskeytyksen yhteydessä. |
| 3. Käyttäjäkohtainen työpöytä
(engl. <i>desktop manager</i>) ... | B. ei ole välttämätön osa nykyaikaista käyttöjärjestelmää. |
| 4. IPC (engl. <i>inter-process commu-
nication</i>) ... | C. tarvitaan ainoastaan, kun suoritetaan virtuaalikoneita. |
| 5. Vuorontaja (engl. <i>scheduler</i>) ... | D. tekee toimenpiteitä sivuvirheen (engl. <i>page fault</i>) yhteydessä. |
| | E. tarjoaa palvelut mm. keskinäiseen poissulkuun. |

Tutkittava esimerkki tehtäviin 6–7: 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 12600 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]$
```

6. **Väite:** komentojen jälkeen tiedoston `kayttojarjestelmat.tex` pituus on pienempi kuin 12600 tavua. (A=kyllä; B=ei)
7. **Väite:** komentojen jälkeen annettava komento `cat kayttojarjestelmat.tex` tulostaisi konsoliin tekstin "`kayttojarjestelmat.tex`" ja rivinvaihdon. (Muistinvirkistys: manuaalin antama synopsis komentolle `cat` on "concatenate files and print on the standard output") (A=kyllä; B=ei)

Ohje tehtävään 8: 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.

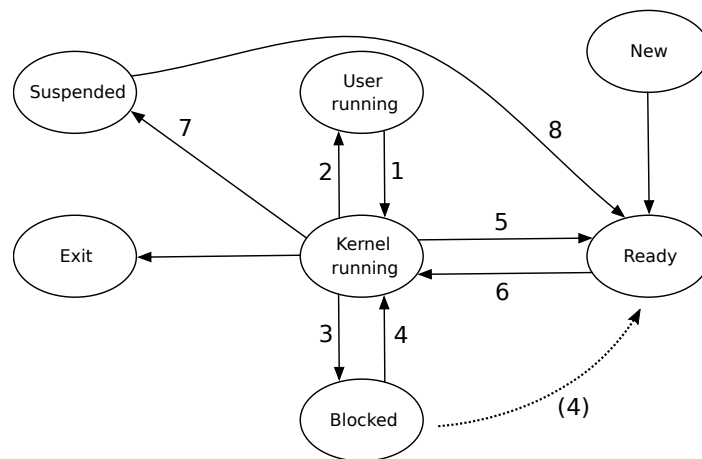
Lauseen alku:

8. Kursilla käsitelty käyttöjärjestelmän rajapintastandardi POSIX (vuoden 2008 versio) määrää, että ...

Vaihtoehdoiset 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. Tiedostovalikko (engl. *File*) on valikkopalkin vasemmanpuoleisin valikko
 D. järjestelmäkutsu *exit()* tapahtuu sijoittamalla rekisteriin RAX luku 60 ja suorittamalla *syscall*-konekielikäskeä

Ohje tehtäviin 9–11: Yhdistä lauseen loppua vastaava kirjain numeroituun alkuun siten, että 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:

9. Tilasiirtymä nro 4 tai (4) tapahtuu, ...
 10. Tilasiirtymä nro 5 tapahtuu, ...
 11. Tilasiirtymä nro 6 tapahtuu, ...

Vaihtoehdoiset loput:

- A. kun ohjelman odottama palvelu, esim. I/O-operaatio, on valmis.
 B. kun käyttöjärjestelmän vuorontaja siirtää suoritukseen toisen prosessin, vaikka nykyinenkin prosessi pystyisi jatkamaan laskemista jo seuraavassa konekielikäskeensä.
 C. kun prosessi aloittaa käyttöjärjestelmän vuorontajalta saamansa aikaikkunan käytön.
 D. kun prosessi on tehnyt pyynnön esim. lukeakseen merkkejä tiedostosta, eikä luettava data ole vielä saapunut fyysiseltä laitteelta toimitettavaksi prosessille saakka.
12. **Väite:** Nykyaikaisen käyttöjärjestelmän yksi päätehtävä on eristää sovellusohjelmat laitteistosta siten, että laitteistoresurssien ohjaus kulkee aina yksinomaan ns. käyttöjärjestelmäkutsurajapinnan (*system call interface*) kautta. (A=kyllä; B=ei)
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 lause on totta. Alkuihin on yksikäsitteinen oikea loppu. Jokainen loppu voi sopia useampaan alkuun tai ei yhteenkään.

Lauseiden alut:

Vaihtoehtoiset loput:

- | | |
|---|---|
| 14. Säie (engl. <i>thread</i>) ... | A. liittyy konekielisen aliohjelmakutsun muotoon. |
| 15. Kehystaulu (engl. <i>frame table</i>) ... | B. liittyy prosessien väliseen kommunikointiin. |
| 16. Sovelluksen binäärirajapinta
(<i>Application Binary Interface</i>) ... | C. liittyy vuoronnukseen. |
| 17. Signaali (esim. POSIXin komen-
nolla kill lähetetty) ... | D. liittyy muistinhallintaan. |

Ohje tehtävään 18: 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:

Vaihtoehtoiset loput (mahdollisesti useita sopivia):

- | | |
|--------------------------------|---|
| 18. Prosessorin keskeytyks ... | A. vaatii prosessorilta vähemmän toimenpiteitä kuin aliohjelmakutsuun siirtyminen (esim. AMD64:n käsky <code>call</code>). |
| | B. voidaan estää sovellusohjelmassa käyttämällä synkronointia. |
| | C. toimii vain moniydinprosessorissa. |
| | D. voi aiheutua käyttäjätilassa toimivan prosessin toimenpiteiden johdosta. |

Tutkittava esimerkki tehtäviin 19–20: Kurssin luennoilta ja demoista tutussa ympäristössä (Linux,bash) tehty yksittäinen, POSIX-syntaksin mukainen komentorivi:

```
arg barg "carg darg" | earg farg | garg --harg | iarg jarg > command
```

19. Montako komentoa shell yrittää käynnistää koko rivin suorittamiseksi?
20. Montako argumenttia rivillä on yhteensä?
21. **Väite:** Tyypillisessä nykyprosessorissa (esim. AMD64) on käyttöjärjestelmätilassa (engl. *kernel mode*) käytettävissä pienempi joukko konekielikäskyjä kuin käyttäjätilassa (engl. *user mode*). (A=kyllä; B=ei)
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)
23. **Väite:** Laitteistoriippumaton I/O-ohjelmiston kerros määrittelee laitteille nimeämiskäytännön. (A=kyllä; B=ei)
24. **Väite:** I/O-operaation pyytäminen aiheuttaa aina pyytävän prosessin siirron suoraan ready-tilaan. (A=kyllä; B=ei)
25. **Väite:** Tietokonejärjestelmän käyttöaste (engl. *utilization*) eli hyödylliseen laskentaan käytettävän ajan määrä paranee, kun lisätään kellokeskeytyksien määrää aikayksikössä. (A=kyllä; B=ei)
26. **Väite:** Prosessin konekieliset käskyt näkyvät samoissa virtuaalimuistiosoitteissa kaikissa prosessin säikeissä. (A=kyllä; B=ei)

Ohje tehtäviin 27–30: Yhdistä lauseen loppua vastaava kirjain numeroituun alkuun siten, että lause on totta. Alkuihin on yksikäsitteinen oikea loppu. Jokainen loppu voi sopia useampaan alkuun tai ei yhteenkään.

Lauseiden alut:

Vaihtoehtoiset loput:

- | | |
|---------------------------------|--|
| 27. Prosessielementti (PCB) ... | A. liittyy ensisijaisesti vuoronnuksen. |
| 28. Sivutaulu ... | B. ilmoitetaan prosessorille muistiosoitteiden automaattista muuntamista varten. |
| 29. Ready-jono ... | C. sisältää kaikki tiedot yksittäisen prosessin tilasta. |
| 30. Blocked-jono ... | |

Tutkittava esimerkki tehtäviin 31–33: Luento-esimerkeistä tuttua C-kielistä ohjelmanpätkeä 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) {
    pthread_mutex_lock(&mymutex);
    for (int i = 1; i <= N; i++){
        summa++;
    }
    pthread_mutex_unlock(&mymutex);
    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 virhekkoodia. (A=kyllä; B=ei)
32. **Väite:** Esimerkin suorituksessa on mahdollista syntyä 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 juumiin. (A=kyllä; B=ei)
34. **Väite:** POSIXin semafori on tietorakenne, joka sisältää etumerkittömän kokonaisluvun ja joukon prosesseja tai säikeitä. (A=kyllä; B=ei)
35. **Väite:** POSIXin semaforikutsun sem_wait(s) suorituksessa semaforin s arvo muuttuu aina. (A=kyllä; B=ei)

Tutkittava esimerkki tehtäviin 36–38: 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 }
```

36. **Väite:** Rivi 4 tapahtuu ainoastaan fork() -kutsun luomassa lapsiprosessissa. (A=kyllä; B=ei)
37. **Väite:** Rivin 5 tullessa suoritukseen on aina olemassa sekä shell että sen luoma lapsiprosessi. (A=kyllä; B=ei)
38. **Väite:** Rivin 11 suorittaminen tapahtuu fork() -kutsun luomassa lapsiprosessissa. (A=kyllä; B=ei)

Tutkittava esimerkki tehtäviin 39–42: Kuvitellaan, että meillä on käytössä yksinkertainen tietokone, jonka virtuaalimuistiosoitteissa on 20 bittiä, joista ensimmäiset 8 ilmoittavat sivunumeron ja loput 12 ilmoittavat tavuindeksin sivun sisällä, eli käytössä on 4 kilotavun sivut. Fyysiset muistiosoitteet ovat 24-bittisiä. Keskumuistista on varattu prosessien käyttöön tasan 8 kehystä fyysistä muistia osoitteissa 0x100000-0x107fff. Heittovaihdon eli swapin avulla käytettävissä olevan muistin kokonaiskoko on 0x1000000 tavua (n. 16 megatavua). Tietorakenteiden tilanne tarkasteluhetkellä on seuraava. Prosessien sivutauluista näytetään vain kartoitetut osat kahden prosessin (PID:t 7 ja 14) osalta.

Prosessin (PID=7) sivutaulun kartoitetut rivit:

virt. sivu	fyys. sivu	muist. (P)	dirty (D)	diskIndex
0x02	0x000	0	0	0x0022
0x03	0x101	1	1	0x0008
0x04	0x000	0	0	0x0004
0x2e	0x104	1	0	0x0abc
0x7f	0x107	1	0	0x0007

Prosessin (PID=14) sivutaulun kartoitetut rivit:

virt. sivu	fyys. sivu	muist. (P)	dirty (D)	diskIndex
0x02	0x100	1	0	0x0010
0x03	0x000	0	0	0x00f2
0x2e	0x106	1	1	0x00bb
0x7f	0x102	1	1	0x00bc

Järjestelmän kehystaulu:

fyys. sivu	omistajan PID	omistajan PTE#	aikayksiköt edellisen käyttökerran jälkeen
0x100	14	0x02	19
0x101	7	0x03	75
0x102	14	0x7f	80
0x103	234	0x45	8
0x104	7	0x2e	12
0x105	876	0x45	4
0x106	14	0x2e	1
0x107	7	0x7f	9

39. Prosessori suorittaa prosessia 7, ja käskyosoiterekisterin sisällöksi tulee 0x2e2e2. Keskeytystä ei ole pyydetty. Mistä fyysisestä osoitteesta prosessori lähtee noutamaan seuraavaa konekielikäskyä?
40. Mistä virtuaaliosoitteesta prosessi 14 saisi käyttöönsä tavun, joka sijaitsee kovalevyllä indeksissä 0xf2345 heittovaihto-osion / -tiedoston (engl. *swap space*) alusta lukien?
41. Prosessi 14 suorittaa hyppykäskyn aliohjelmaan muistiosoitteessa 0x03456. Tapahtuuko prosessorissa sivuvirhe / sivunvaihtokeskeytys? (A=kyllä; B=ei)
42. Prosessissa 234 aiheutuu sivunvaihtokeskeytys. Korvausalgoritmi on LRU. Onko jonkin sivun sisältö tallennettava levyllä? (A=kyllä; B=ei)

Tutkittava esimerkki tehtäviin 43–45: Kurssin luennoilta tutulla symbolisella konekielellä (GNU assembler; AT&T -syntaksi; AMD64; Linux-järjestelmä) kirjoitettu, rivi riviltä kommentoitu kokonainen ohjelma. HUOM: suoritus alkaa osoitteesta "_start", kuten oikeassa GNU-työkaluilla tuotetussa ohjelmassa.

```
ali:
    subq    $2,%rcx    # vähennä 2 rekisterin RCX arvosta
    movq    %rcx,%rdi  # kopioi rekisterin RCX sisältö rekisteriin RDI
    ret

_start:
    # Aloituspaikan symbolinen osoite on "_start"
    movq    $22,%rcx   # sijoita luku 22 rekisteriin RCX
    call    ali        # kutsu aliohjelmaa osoitteessa ali
    call    ali        # kutsu aliohjelmaa osoitteessa ali
    inc     %rdi       # lisää 1 rekisterin RDI arvoon
    movq    $60,%rax   # Valmistele rajapinnan mukainen exit()-kutsu
    syscall          # Toteuta exit(KOODI), missä KOODI on RDI:n arvo
```

43. Kuinka monta konekielikäskyä ohjelmanpätkän jälki sisältää, ts. kuinka monta käskyä prosessi suorittaa siitä alken, kun käyttöjärjestelmä siirtää kontrollin nimettyyn aloituspisteeseen "_start"?
44. Mikä on rekisterin RCX sisältö ohjelmanpätkän suorituksen lopuksi?
45. Mikä on rekisterin RDI sisältö ohjelmanpätkän suorituksen lopuksi?

Tutkittava esimerkki tehtäviin 46–47: 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
```

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

Tutkittava esimerkki tehtävään 48: Yksi vakiokirjasto (otsikkotiedosto "stdio.h") käytävä C99-koodirivi:

```
printf("%d", 0x101);
```

48. Mitä koodirivin suorittaminen tulostaa? (speksistä: "*argument is converted to signed decimal notation*")

Vapaaehtoinen vapaa sana

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