

ITKA203 -- Käyttöjärjestelmät -- Tentti 15.8.2014

Kevään 2014 kurssi (luennot, demot 1-4 ja oheismateriaali) / Paavo Nieminen <paavo.j.nieminen@jyu.fi>.

HUOM: Tarkastus jaetaan opettajien kesken. Palauta siksi **tasan kolme erillistä vastauspaperia** (ei siis yhtä eikä kahta), joissa jokaisessa on vähintään **nimesi, syntymäaikasi, kurssin tiedot** (koodi ja/tai nimi) sekä **vähintään tyhjä vastaus jokaista tehtävännumeroa kohden**. Jako on seuraava: Tehtävät 1, 2 ja 3 yhteen vastauspaperiin. Tehtävät 4 ja 5 yhteen paperiin. Tehtävä 6 yhteen paperiin.

Kysymys 1/6 (vastauspaperille 1):

Selitä, mitä seuraavat termit tarkoittavat tietokonelaitteiston ja käyttöjärjestelmän yhteydessä:

- a) prosessorin konteksti
- b) prosessorin kontekstin vaihto
- c) lokaalisuusperiaate
- d) signaali (prosessien välisen kommunikoinnin yhteydessä)
- e) kriittinen alue (ohjelmakoodissa)

(1+1+1+1+1=5p)

Kysymys 2/6 (vastauspaperille 1):

- a) Missä toimintatiloissa prosessorin täytyy vähintään pystyä olemaan?
- b) Miksi?

(1+1=2p)

Kysymys 3/6: (vastauspaperille 1)

- a) Anna esimerkki käyttöjärjestelmän toteutuksessa tai konfiguroinnissa välttämättömästä kompromissista, eli kahdesta ristiriitaisesta toiminnallisesta tai laadullisesta tavoitteesta, joita ei voida saada yhtä aikaa optimaalisiksi.
- b) Mihin (tietokonelaitteistossa) tarvitaan keskeytyksiä?

(2+1=3p)

Kysymys 4/6 (vastauspaperille 2):

- a) Kuvaile tuottaja-kuluttaja -ongelma.
- b) Esitä C-ohjelma tai pseudokoodi, jossa tuottaja-kuluttaja -ongelma hoidetaan käyttäen käyttöjärjestelmän (tai alustakirjaston) semaforipalveluita.

(3p)

Kysymys 5/6 (vastauspaperille 2):

Selitä (pseudokoodina tai sanallisesti), miten käyttöjärjestelmä käsittelee sivunvaihtokeskeytyksen (page fault). Oleta että kehyksen valintamenetelmänä on LRU (least-recently-used). Kerro myös, mitä sisäisiä tietorakenteita käyttöjärjestelmä hyödyntää ja miten niitä muutetaan.

(3p)

Kysymys 6/6 (vastauspaperille 3):

- a) Laitoin talon grillijuhlien digikuvat nettiin naapureita varten, mutta he sanovat että "kuvat eivät toimi". Hakemistolistaukseni alku näyttää tältä:

```
-bash-2.05b$ ls -l ~/www/grillikuvat/*.JPG
-rw-----    1 nieminen opis      1027107 Jun 9 16:34 IMG_1940.JPG
-rw-----    1 nieminen opis       836088 Jun 9 16:34 IMG_1942.JPG
```

Mikä on vialla ja miten voin korjata asian?

- b) Olen pääteyhteydessä Linux-palvelimeen, mutta en muista miten unix-komento `whatever` toimii. Miten saan sen selville (tarkistamatta Internetistä)?
- c) Mitä seuraava shell-komentorivi tekee:

```
ls *.jpg > gradu.doc
```

- d) Luettele kolme tarkoitusta, joihin shell-skriptejä voidaan käyttää.
- e) Seuraavan skriptin, nimeltään `skripti.sh`, haluttaisiin toimivan kommenttien mukaisesti, mutta kaikkea ei ole vielä toteutettu (TODO-merkinnät) ja lisäksi tekijälle on tullut joitakin karkeitä virheitä tähänastisessa (`find`-komennon argumentit ovat kuitenkin oikeassa järjestyksessä.) Kirjoita skripti uudelleen niin, että se toimii kommentteissa ilmoitetuin tavoin. Kommentteja ei tarvitse toistaa vastauksessa vaan ainoastaan toimiva skripti.

```
#!/bin/sh
# Tämä skripti listaa C#-lähdekoodit annetusta hakemistosta ja
# tallentaa listan tiedostoon. Käyttöesimerkki:
#   skripti.sh /home/nieminen/koodit listaus.txt
#
# Em. komento tallentaa 'listaus.txt' -nimiseen tiedostoon
# kellonajan ja sillä hetkellä hakemiston '/home/nieminen/koodit'
# alla tiedostojärjestelmässä olevat .cs -päätteiset tiedostot.

# TODO: Tässä pitäisi varmistaa, että käyttäjä on antanut tasan
# kaksi argumenttia ja lopettaa virhetilanteena, jos näin ei ole.

dirname=$1; outfname=$2

# TODO: Pitäisi tarkistaa, että tulostiedostoa ei vielä ole
# olemassa; muuten pitäisi lopettaa kesken virhetilanteena.

# TODO: Pitäisi tarkistaa, että hakemistonimi todella viittaa
# olemassa olevaan hakemistoon eikä esim. normaaliin tiedostoon.
# Muuten pitäisi antaa käyttäjälle ilmoitus ja lopettaa virheenä.

# Laitetaan tiedoston ensimmäiselle riville otsikoksi
# tämänhetkinen kellonaika ja suorittajan oma käyttäjätunnus:
echo Käyttäjät whoami listasi koodit hetkellä date > outfname

# Jatketaan tiedostoa listauksella, joka alkukomentissa luvataan:
find "$dirname" -name "*.cs" > $outfname
```

(1+1+1+1+4 = 8p)

Liite: Yhden sivun bash-luntti

Joitakin bash-shellin ohjelmointirakenteita:

```
muuttuja=57                # tarkka syntaksista: ei välilyöntejä!
muuttuja="eki"; muuttuja="$muuttuja$muuttuja"
read muuttuja              # lukee muuttujaan syöttövirrasta

if EHTO                    # Myös: if EHTO; then ...; \
then                       #       elif EHTO; then ...; fi
...                        #
fi                          #

for muuttuja in LISTA      # muttujan "esittelyssä"/asetuksessa ei $
do
... jotakin $muuttuja jotakin ... # muuttujan käytössä $muuttuja
done

while EHTO; do ... ; ... ; done    # käskyerotin rivinvaihto tai ;

# Tee jotain syötteen kaikille riveille:
while read rivi; do echo "luin: $rivi" ; done < rivit.txt

case $hanen_nimensa in
  kirsimarja)
    echo "Hei Kippe" ;;
  eskomatias)
    echo "Moi E.M." ;;
esac
```

Aliohjelmat mahdollisia&hyödyllisiä, mutta ei käsitellä ITKA203:lla.

Joidenkin ehtojen käyttöä (Välilyönnit merkityksellisiä!):

```
[[ -d TIEDOSTONIMI ]]      # tosi, jos on hakemisto
[[ -f TIEDOSTONIMI ]]      # tosi, jos on tavallinen tiedosto
[[ -w TIEDOSTONIMI ]]      # tosi, jos on olemassa ja kirjoitettavissa
[[ ! -f TIEDOSTONIMI ]]    # tosi, jos ei olemassa tai ei tavallinen
[[ "$muuttuja" -le "57" ]] # tosi, jos muuttuja <= 57 (myös lt,gt,ge)
```

Argumenttien käyttö: (yli 9 arg mahd., mutta ei käsitellä ITKA203:lla)

```
echo "Tämän skriptin nimi on $0. Eka argumentti $1, neljäs $4"
echo "Argumenttien määrä on $#"
```

Joitakin sisäänrakennettuja toimintoja:

```
exit VIRHEKOODI           # koodi 0 tarkoittaa onnistumista
cd                         # vaihtaa skriptin työhakemistoa
echo                       # kaiuttaa tekstiä
let muuttuja=$muuttuja+3   # (perusaritmetiikkaa)
```

Erikoismerkkejä (tuttuja interaktiivisesta käytöstä; toimii skripteissä):

```
< > >> | 'KOMENTO'
```