ITKA203 Käyttöjärjestelmät, kesä 2007
Tenttitärpit

Tässä on suurin piirtein kaikki vuodesta 2003 alkaen kurssin tenteissä koskaan kysytyt kysymykset, ja mukana on myös muutama uusi. Jos osaat vastata jokaiseen, osaat koko Käyttöjärjestelmät -kurssin sisällön. Jos osaat vastata vain osaan, puutteita tiedoissa on, ja arvosana riippuu siitä, mitä kohtia satun missäkin tentissä kysymään. Muutoksia aiempiin:

- Olen yrittänyt täsmentää kysymyksiä ja jakaa niitä pienempiin osiin, koska esseetyyppisten pitkien vastausten tarkastaminen ja tasapuolinen arvostelu on vaikeaa.
- Olen yrittänyt tehdä kysymyksistä soveltavampia ja kysyä varsinaisen artefaktitoteutuksen lisäksi syitä ja päämääriä.
- C-shell -skriptitehtävät vaihdettu Bourne Shelliksi.

Jokaisessa kysymyksessä on pistemäärä, jonka arvoiseksi se laajuudeltaan katsotaan. Jokaiseen tenttikertaan tulen valitsemaan näiden joukosta erilaisia tehtäviä siten, että pistesumma on 24. Aihepiirit jakautunevat tentissä suurin piirtein seuraavasti: 6 pistettä laitteiston toimintaa ja konekieliohjelmointia, 6 pistettä shellin & unix-komentojen & skriptien käyttöä ja 12 pistettä käyttöjärjestelmätoteutuksiin liittyvää.

HUOM: Luvataan, että tämä on lopullinen tenttitärppikokoelma, mutta varaan silti oikeuden tehdä pieniä muutoksia, erityisesti:

- Voin muuttaa tehtävästä luvattua pistemäärää ylöspäin, jos alkaa tuntua että se on haastavampi tai aikaavievämpi vastata kuin mitä tässä on arvioitu. (Kuitenkin eri tenttikerrat tulevat olemaan pisteytykseltään tasapainossa)
- Jotkut tehtävät jäivät aiempaan, mielestäni vaikeasti tarkasteltavaan otsikkomuotoon. Saatan kysyä niitä samaan tapaan täsmällisiin alakohtiin jaettuina kuin muutkin tässä olevat uudet kysymysmuodot. Uusia aiheita en kuitenkaan tässä mainittujen ulkopuolelta tule kysymään. Aiheesta kuin aiheesta (tässä olevista) sinun on varauduttava vastaamaan tiiviisti ja erillisesti seuraaviin: Miksi asia on tärkeä, eli mitä sillä saavutetaan. Mitä asiaan liittyy, eli sovellusesimerkkejä tai vastaavia. Miten asia pääpiirteissään on toteutettu.
- Toteutettavat Shell-skriptit voivat olla erilaisia (mutta vaikeudeltaan saman tasoisia) kuin tässä esitetyt

Onnea matkaan, ja iloa lukemiseen. Huomioikaa materiaalit ja linkit kurssin nettisivulta. Lopullinen materiaalikokoelma ja sisältörajaus julkaistaan 25.6.2007 klo 23:59 mennessä (tai parin tunnin viiveellä ennen tiistaiaamua). Paperikopiot saatavilla (kurssilaisille ilmaiseksi) saman viikon loppupuolella, viimeistään ensimmäisessä demo neljän tilaisuudessa.

1 Tietokone fyysisenä laitteena, Von Neumann-arkkitehtuuri, C-ohjelmointi

Otsikossa mainitut aihepiirit ovat "esitietoa" ymmärryksen pohjaksi. Niitä ei kysytä tentissä useinkaan suoraan, mutta ...

- hyvissä vastauksissa käytät sujuvasti sanoja 'prosessori', 'rekisteri', 'käskyosoiterekisteri', 'pino-osoitin', 'väylä', 'oheislaite' (tai 'I/O -laite'), 'muisti', jne...
- ymmärrät mitä nämä sanat tarkoittavat nykytietokoneessa (ja se näkyy vastauksestasi, jos se on "hyvä")
- niissä kysymyksissä, joissa edellytetään lyhyen C-ohjelman ymmärtämistä tai tuottamista, syntaksin pilkut eivät ole tärkeimpiä; kannattaa kirjoittaa kommentteja, joilla selität alkuperäisen ohjelmoinnillisen tarkoituksen — jos se on oikein, ei C:n syntaksilla ole niin väliä.

2 Konekieli, suoritus, aliohjelmat, keskeytykset

Kysymys:

a) Mitä rekisterejä pitää prosessorissa vähintään olla?

b) Mihin kutakin välttämättömistä rekistereistä käytetään?

c) Mitä rekisterit fyysisen toteutuksen kannalta ovat?

(yht. 3 p)

Kysymys: Ajatellaan pinokehysmallia pääpiirteissään.

a) Mitkä ensinnäkin ovat edellytykset aliohjelmien (tai metodien) hyödylliselle käytölle ohjelmoinnissa: mitä aliohjelman pitää kyetä tekemään ja mitä siellä pitää voida olla? Riittää käsitellä *muuttujien käyttöä* (sekä kutsuvan että kutsuttavan ohjelman tarpeiden mukaan) ja *suoritusjärjestystä* (kutsuvan ohjelman tarpeiden mukaan).

b) Miten nämä saadaan toteutumaan konekielelle käännetyssä ohjelmassa: mitkä ohjelmakäännöksen ja laitteiston osat ovat tekemisissä keskenään, missä vaiheessa suoritusta ja miten?

(Täsmennys: Saat selittää b-kohdassa minkä tahansa tuntemasi tavan, jolla a-kohdassa tunnistamasi tavoitteet toteutuvat riittävästi. Luennoilla ja varsinaisessa monisteessa käsiteltiin klassinen pinokehysmalli; uudessa materiaalissa esiteltiin pinokehysmallin eräs moderni laajennos. Voit käyttää rekistereille kurssimateriaalista tuttuja x86-arkkitehtuureille tyypillisiä rekisterinimiä.)

(3 p)

(Huom: Em. täsmennetty kysymys korvaa tentissä usein olleen ympäripyöreän kysymyksen "Aliohjelman suoritusperiaate konekielitasolla, ts. selvitä pinon käyttö ja paluuarvon välittäminen.")

Kysymys: Selitä, mitä prosessori tekee (yhden suoritussyklin aikana), kun se suorittaa

a) aliohjelmakutsun

b) paluun aliohjelmasta.

(1 p)

Kysymys: Mihin tarvitaan keskeytyksiä? (1 p)

Kysymys: Selvitä keskeytyskäsittelyn perusmekanismeja.

a) Miksi ylipäätään keskeytyskäsittely tehdään?

b) Kerro kolme esimerkkitilannetta keskeytyksestä: Kuka keskeytti, miksi ja millä tavoin.

c) Olipa syy keskeytykseen mikä tahansa, niin mitä operaatioita prosessori tekee keskeytyksenhoitoon siirtymisessä? Aloita kertomus tilanteesta juuri ennen keskeytystä, ja päätä se tilanteeseen, jossa käskyosoiterekisteri osoittaa jo varsinaisen käsittelijäkoodin ensimmäistä käskyä.

d) Miten keskeytyksenhoidosta paluu tapahtuu? Aloita kertomus tilanteesta, jossa prosessori ei vielä tiedä, milloin paluu tapahtuu, ja päätä se tilanteeseen, jossa käskyosoiterekisteri ensimmäistä kertaa osoittaa jo jonnekin muualle kuin käsittelijäkoodiin.

(4 p)

(Huom: Em. täsmennetty kysymys korvaa tentissä usein olleen ympäripyöreän kysymyksen "Selvitä keskeytyskäsittelyn perusmekanismi; ts. keskeytyksenhoitoon siirtyminen ja sieltä paluu")

Kysymys: Oletetaan, että käytössä on jokin sellainen prosessori, jossa on 16-bittinen rekisteri nimeltä `CX`. Pinoa täytetään muistissa normaalisti "alaspäin". Muistipaikan koko on 8 bittiä. Tarkastellaan seuraavan assemblerkielisen ohjelmapalasen suoritusta:

```
    mov   CX, 10       ; aseta laskurin arvoksi 10
@alku:
    push  CX           ; vie laskurin sisältö pinoon
    dec   CX           ; vähennä laskuria yhdellä
    jnz   @alku        ; jatka, jos CX ei mennyt nollaksi
```

(Syntaksi ihan vähän erilaista kuin kurssimateriaalissa; idea selvinnee kommenteista.)

a) Mikä kaikki on muuttunut (prosessorissa, muistissa? missä siellä?), kun ohjelma on suoritettu? Pyri huomioimaan kaikki asiat sillä oletuksella, että suorituksen aikana ei tullut yhtään keskeytystä.

b) Mitä on muistipaikassa [SP+8] ohjelman suorituksen jälkeen? (SP on pino-osoitinrekisteri, syntaksi [SP+8] tarkoittaa samaa kuin kesän 2007 kurssimateriaalissa 8(%SP))

c) Selitä, mitä kaikkea prosessori tekisi välittömästi (ennen seuraavan suoritussyklin alkua), jos `push CX` -komennon suorituksen jälkeen tulisi keskeytyspyyntö laitteistolta (eli kuvaile FLIH-toiminta). Selvitä prosessorin mahdollisista toimintatiloista riippuvat erityistapaukset tekemättä mitään oletuksia, ja kerro mistä kohtaa prosessori jatkaa suoritusta `push CX`:n jälkeen missäkin erityistapauksessa.

(noin 4p, eli ehkä 1+1+2)

Kysymys: Selitä kussakin seuraavista tilanteista tapa, jolla tietokoneen prosessori määrittää uuden sisällön käskyosoiterekisterille ("IP"). Jos IP voi joistakin, mahdollisesti vaihtelevista, olosuhteista riippuen saada vaihtoehtoisia arvoja, kerro missä olosuhteissa mikäkin vaihtoehto valitaan.

1) normaalin käskyn suoritus (ei keskeytystä)

2) ehdollinen hyppykäsky (ei keskeytystä)

3) aliohjelmakutsu (ei keskeytystä)

4) paluu aliohjelmasta (ei keskeytystä)

5) keskeytyspyyntö on saapunut laitteelta

6) suoritettava käsky on ohjelmoitu keskeytys (`trap`/`int`/`svc`)

7) `tas` ("test and set") -komento moniprosessorikoneessa

(noin 6/7 oikein = 6p)

Kysymys: Luettele kahdeksan käyttöjärjestelmän palvelua, joita kutakin voisi vastata yksi käyttöjärjestelmäkutsu. (1 p)

Kysymys: Miten käyttöjärjestelmän palveluiden käyttö tapahtuu? Eli miten tietyn palvelun käyttö näyttäytyy **a)** sovellusohjelmoijan kannalta, **b)** prosessorilaitteiston kannalta, **c)** käyttöjärjestelmän toteutuksen kannalta. (2 p)

3 Virtuaalikoneen käsite

Kysymys: Anna esimerkki virtuaalikonehierarkiasta (2 p).

4 Käyttöjärjestelmän tehtävät ja rakenne

Kysymys: Käyttöjärjestelmän rakenteesta:

a) Luettele neljä merkittävää käyttöjärjestelmän osaa

b) kuvaile yhdellä lauseella kunkin a-kohdassa mainitsemasi osan merkitystä kokonaisuuden kannalta

c) mainitse esimerkki yhdestä kunkin a-kohdassa mainitsemasi osan toimintaan liittyvästä käyttöjärjestelmäkutsusta (eli selitä korkeintaan yhdellä lauseella, mitä tarkoitusta varten kutsu on; sen nimellä ei ole mitään väliä).

(2 p)

Kysymys: Millaisiin korkeimman tason moduuleihin modernin moniajokäyttöjärjestelmän lähdekoodi esimerkiksi kannattaisi jakaa? Miksi? (2 p)

Kysymys: Mitkä ovat mielestäsi käyttöjärjestelmän neljä tärkeintä tehtävää 2000-luvun palvelintietokoneessa. (2 p)

Kysymys: Mikä on microkernel-malli? Mitkä sen hyödyt olisivat, ja miksi toisaalta tällä hetkellä ei ole paljon microkernel-toteutuksia? (2 p)

5 Käyttöjärjestelmäkutsut, keskeytysten käsittely

Ks. kohta 2

6 Prosessi, prosessien vuorottelu, synkronointi ja kommunikointi

Kysymys: Mikä on prosessitaulu ja mitä se sisältää. (2p)

Kysymys:

a) Mikä on prosessielementti (”PCB”)?

b) Kuka tarvitsee PCB:tä ja mihin tarkoituksiin?

c) Mitä kaikkea PCB sisältää? (2 p)

Kysymys: Missä eri tiloissa prosessi voi olla? Mitkä siirtymät tilasta toiseen ovat mahdollisia, ja milloin? (2 p)

Kysymys: Täydennä seuraavan ”pseudo-C -kielisen” ohjelman kommentit; luekomento()-aliohjelma lukee parametreilleen arvot. pid on kokonaisluku. (3p)

```
/* Tämä on minimalistinen esimerkkitoteutus ohjelmasta, jollaista
 * tyypillisesti nimitetään ... [VASTAUKSESI a-kohta].
 *
 * Ohjelman avulla sen käyttäjä voi ... [VASTAUKSESI b-kohta]
 */
```

```
while(true){

  luekomento(komento, parametrit);

  pid = fork();    /* pid:hen sijoitetaan fork()-funktion
                    * paluuarvo. Tuo kyseinen fork() vastaa
                    * UNIX-järjestelmässä käyttöjärjestelmäkutsua,
                    * jonka tehtävä on ... [VASTAUKSESI c-kohta].
                    */

  if (pid > 0) {

    /* Suorituksen saapuessa tähän lohkoon, tiedetään
     * käyttöjärjestelmän tilasta, että ... [VASTAUKSESI d-kohta]
     * ja tästä prosessista että ...  [VASTAUKSESI e-kohta]
     */

    status = wait(); /* Tämän rivin käyttöjärjestelmäkutsun wait()
                      * tehtävä on ... [VASTAUKSESI f-kohta]
                      */

  } else if (pid == -1) {


    /* Suorituksen saapuessa tähän lohkoon, tiedetään
     * käyttöjärjestelmän tilasta, että ... [VASTAUKSESI g-kohta]
     * ja tästä prosessista että ...  [VASTAUKSESI h-kohta]
     */

    exit(1)         /* Tämän rivin käyttöjärjestelmäkutsun exit()
                     * tehtävä on ... [VASTAUKSESI i-kohta]
                     */

  } else {


    /* Suorituksen saapuessa tähän lohkoon, tiedetään
     * käyttöjärjestelmän tilasta, että ... [VASTAUKSESI j-kohta]
     * ja tästä prosessista että ...  [VASTAUKSESI k-kohta]
     */
```

```
    exec(komento, parametrit);

    /* Edellisen rivin käyttöjärjestelmäkutsun exec() tehtävä on
     * ... [VASTAUKSESI l-kohta]
     */

    /* Niinpä ollen seuraavan rivin suoritus
     * ... [VASTAUKSESI m-kohta]
     */
    printf("%d",pid);

  }
}
```

Kysymys: Seuraavaa aliohjelmaa voidaan mahdollisesti ajaa useammassa kuin yhdessä säikeessä, ja viestikanava `statuskan` on säikeille yhteinen. Toistaiseksi kukaan ei ole miettinyt synkronointikysymyksiä ohjelmistossa, johon aliohjelma liittyy.

a) Millainen ongelma tässä voi tulla, ja missä tilanteessa?

b) Kerro, miten käyttäisit semaforia ongelman ratkaisemiseen. (Koodia ei tarvitse kirjoittaa – kunhan selität)

(2 p)

```
/* Tulostaa toimintatilan statuskanavaan tekstinä.
 * Käytämme tasan 10 merkin mittaisia statuskoodeja,
 * joiden perusteella koneisto säätyy uuteen tilanteeseen.
 */
tulosta_status(char teksti[]){
  int i;
  for (i=0;i<10;i++){
    fputc(teksti[i], statuskan);  /* merkin tulostus virtaan */
  }
}
```

Kysymys:

a) Semafori on eräs tyypillinen käyttöjärjestelmän ylläpitämä tietorakenne. Mitä tietoja sellaisessa on? Anna tiedoille symboliset nimet, joita voit käyttää c-kohdan selityksessä.

b) Mihin tarkoitukseen semaforia käytetään? Nimeä (tai kuvaile, jos et muista nimiä) kaksi tyypillistä ongelmaa, jotka voidaan ratkaista käyttämällä apuna semaforeja.

c) Miten on toteutettu kaksi semaforin varsinaiseen käyttöön tarvittavaa käyttöjärjestelmän palvelua? Mieluiten kirjoita selkeä pseudokoodi kummankin palvelun toteutuksesta. (Keskity siis varsinaiseen käyttöön ja jätä huomioimatta semaforin luominen, käyttöönotto ja tuhoaminen.)

(2 p)

Kysymys: Selitä käsite "kriittinen alue". (1 p)

Kysymys:

a) Kuvaile poissulkemisongelma: millaisissa tilanteissa se tulee kyseeseen, mitä voi käsittelemättömänä aiheuttaa

b) Esitä C-ohjelma tai pseudokoodi, jossa poissulkemisongelma ratkaistaan käyttämällä käyttöjärjestelmän semaforipalveluita. (2 p)

Kysymys:

a) Kuvaile perinteinen tuottaja-kuluttaja -ongelma

b) Esitä C-ohjelma tai pseudokoodi, jossa tuottaja-kuluttaja -ongelma ratkaistaan käyttämällä käyttöjärjestelmän semaforipalveluita.

(3 p)

Huom: Em. täsmennetyt kysymykset korvaavat tentissä usein olleen ympäripyöreän kysymyksen "Semafori ja sen käyttö synkronoinnissa/poissulkemisessa"

Kysymys: Mitä ovat signaalit, ja miten niitä käytetään (2 p)

Kysymys: Viestinvälitys send- ja receive -kutsuilla. (2 p)

Kysymys: Selitä prosessin ja säikeen ero. (1 p)

Kysymys: Esimerkiksi mihin tarkoitukseen sovellusohjelmissa voisi hyödyntää säikeitä? (1 p)

7 Muistinhallinta

Kysymys: Tässä riittää käsittely kurssimateriaalin kuvailemassa tarkkuudessa. (Jos tiedät enemmän, se on hyvä, mutta älä yritä kirjoittaa sitä romaania tenttikonseptiin).

a) Mihin nykyaikaisen käyttöjärjestelmän ja laitteiston muodostamaan kokonaisuuteen liittyvät käyttöjärjestelmän tietorakenteet, joiden nimet ovat "sivutaulu" ja "kehystaulu"?

b) Mikä siinä on perusidea?

c) Kuvaile päämääriä ja tarpeita, joihin kyseinen kokonaisuus pyrkii vastaamaan. Millä tavoin tilanne siis on parempi kuin jos sitä ei olisi.

d) Mikä on sivutaulu, ja mitä tietoja se sisältää?

e) Montako sivutaulua ainakin on olemassa kullakin hetkellä?

f) Mikä on kehystaulu, ja mitä tietoja se sisältää?

g) Montako kehystaulua ainakin on olemassa kullakin hetkellä?

h) Kuvaile, missä tilanteissa ja mihin tarkoituksiin näiden eri taulujen eri tietoja tarvitaan.

(4 p)

Huom: Em. täsmennetty kysymys korvaa tentissä usein olleen ympäripyöreän kysymyksen "Sivuttavan virtuaalimuistin periaate ja merkitys. Selvitä myös, millaisia tauluja virtuaalimuisti edellyttää ja miten se niitä käyttää (4 p)"

8 Oheislaitteiden ohjaus, ajuriohjelmistot

Kysymys: Merkin lukeminen päätteeltä on esimerkki I/O-operaatiosta. Kerro, miten luku tapahtuu: mitkä kaikki ohjelmisto- ja laitteisto-osat osallistuvat sen toteuttamiseen missäkin vaiheessa, ja mitä niiden kunkin vastuulla on?. Aloita kertomus hetkestä, jolloin käyttäjän ohjelma tekee operaatiopyynnön, ja lopeta siihen, kun operaatio on valmis ja käyttäjän ohjelma jatkuu taas. (2 p)

Kysymys: Megatavun mittaisen JPG-kuvan lukeminen kovalevyltä muistiin myöhempää dekompressointia ja tulostamista varten on esimerkki I/O-operaatiosta. Kerro, miten luku tapahtuu: mitkä kaikki ohjelmisto- ja laitteisto-osat osallistuvat sen toteuttamiseen missäkin vaiheessa, ja mitä niiden kunkin vastuulla on?. Aloita kertomus hetkestä, jolloin käyttäjän ohjelma tekee operaatiopyynnön, ja lopeta siihen, kun operaatio on valmis ja käyttäjän ohjelma jatkuu taas. (2 p)

Kysymys: Miksi tarvitaan erikseen laiteriippuva ja laiteriippumaton I/O-ohjelmisto? (1 p)

9 Tiedostojärjestelmät

Kysymys:

a) UNIX-tiedostojärjestelmän toteutus (Millaisia tietorakenteita UNIX-käyttöjärjestelmässä tätä varten on, ja mitä ne sisältävät).

b) Mitä vaiheita ja sisäisiä operaatioita käyttöjärjestelmän palvelussa täytyy käydä läpi, että levyltä löytyisi tiedosto nimeltä `/home/jokuope/valokuvat/kurssi.jpg`

(2 p)

Kysymys: UNIX-tiedostojärjestelmän toteutus. Selvitä myös, miten suoritetaan tiedoston haku nimen perusteella. (2 p)

Kysymys: Mitkä ovat päätavoitteet RAID-järjestelmän (Redundant Array of Independent Disks) käytössä? Mitä eri keinoja RAID tarjoaa tavoitteiden saavuttamiseksi? (RAID-tasojen numeroita ei tarvitse muistaa; selitä vaan ne käytetyt keinot lyhyesti ja ymmärrettävästi jossakin järjestyksessä.) (2 p)

Kysymys: On tunnettua, että käyttäjän kovalevyn tiedot voivat olla vaarassa esimerkiksi sähkökatkon sattuessa, ja ulkoisen muistilaitteen tiedot vaarantuvat esim. jos laite irrotetaan väärään aikaan USB-portista. Miksi näin on? Mitä suojakeinoja käyttöjärjestelmäohjelmisto voi tähän asiaan tarjota? (2 p)

10 Unix, ”Unix-työkalut” ja shell-komentorivit

Näihin tehtäviin saatetaan jakaa tehtäväpaperin ohessa lisämateriaalia, kuten otteita englanninkielisistä man-sivuista.

Kysymys: Selitä mikä on interaktiivinen shell. (1 p)

Kysymys: Mitä seuraava shell-komentorivi tekee: `ls *.jpg > gradu.doc` (1 p)

Kysymys: Mitä seuraava skripti tekee (hipsukat ovat ”backtick” -merkkejä)::

```
#!/bin/sh
echo ‘$0‘
```

(1 p)

Kysymys: Tee Bourne Shell -skripti, joka testaa onko argumenttina annettu hakemisto olemassa. Esimerkiksi skripti voitaisiin ajaa komennolla `loytyyko hake/alihake/` ja tulostus olisi ”kylla” tai ”ei” tilanteen mukaan. (1 p)

Kysymys: Laajenna edellistä siten, että parametrina voisi olla vaikka kuinka monta hakemistoa, esim. ”loytyyko hake/ali1/ hake/ali2/” kertoisi kummankin hakemiston kohdalla ”kylla” tai ”ei”. Viimeistään tässä versiossa tarkista, onko yhtään parametria. Jos ei ole yhtään, skriptin tulee tulostaa virheilmoitus ja loppua virhekoodin kera. (2 p)

Kysymys: Tee Bourne shell -skripti, joka lukee päätteeltä yhden rivin tekstiä ja lisää rivin sitten tiedoston `~/ajatuksia.txt` loppuun. Skriptin tulee ensin erikseen tarkistaa, onko kyseinen tiedosto olemassa ja onko siihen kirjoitusoikeus. Jos ei, niin molemmissa tapauksissa on tulostettava virheilmoitus ja lopetettava skripti ennen rivin lukemista päätteeltä. (2 p)

TAI VASTAAVIA SIMPPELEITÄ BOURNE SHELL (.sh) -SKRIPTEJÄ.

Kysymys: Laitoin talon grillijuhlien digikuvat nettiin naapureita varten, mutta he sanovat, että ”kuvat eivät toimi”. Hakemistolistaukseni näyttää tältä:

```
-bash-2.05b$ ls -l ~/www/grillikuvat/*.JPG
total 3964
-rw-------    1 nieminen opis      1027107 Jun  9 16:34 IMG_1940.JPG
-rw-------    1 nieminen opis       836088 Jun  9 16:34 IMG_1942.JPG
-rw-------    1 nieminen opis      1101871 Jun  9 16:34 IMG_1943.JPG
-rw-------    1 nieminen opis      1066770 Jun  9 16:34 IMG_1945.JPG
```

Selitä, mikä on vialla ja miten voin korjata asian. (1 p)

Kysymys: Tee Bourne shell -skripti (tai one-liner) jolla saan muunnettua kaikkien grillikuvien tiedostopäätteet pieniksi kirjaimiksi (eli .JPG → .jpg). (1 p)

Kysymys: Mitä voi tehdä grep-apuohjelmalla? (1 p)

Kysymys: Mitä voi tehdä find-apuohjelmalla? (1 p)