

ITKA203 Käyttöjärjestelmät -- kesä 2011 -- tenttitärpit

Yleistä tenttitärpeistä

Jokaisessa kysymyksessä on pistemäärä, jonka arvoiseksi se laajuudeltaan katsotaan. Jokaiseen tenttikertaan tulen valitsemaan tässä lueteltujen tehtävien joukosta erilaisia siten että pistesumma on 24. Aihepiirit jakautunevat tentissä suurin piirtein seuraavasti: 6 pistettä laitteiston toimintaa ja konekieliohjelmointia, 6 pistettä shellin, unix-komentojen ja skriptien käyttöä ja loput 12 pistettä käyttöjärjestelmän ytimen eri osa-alueisiin liittyviä kysymyksiä. Arvosana läpipääsyn rajalla eli 12 pisteellä on 1 ja täysillä pisteillä 5. Demoista saadut bonuspisteet huomioidaan pohjalle myös läpipääsyn mielessä.

HUOM: Jotkut kysymykset ovat luonteeltaan muuttuvaisia:

- Koodit (assembly- ja bash-ohjelmat), joista kysymykset tehdään, voivat olla erilaiset mutta vaatimuudeltaan vastaavat kuin tässä on esimerkinomaisesti annettu. Niihin pystyy vastaamaan, jos on tutustunut kirjalliseen materiaaliin (mukaanlukien demot) sekä esiteltyihin esimerkkiohjelmiin. Soveltamiskykyä (toivottavasti) jonkin verran edellytetään.
- Jotkut tehtävät ovat yhä yleisessä “esseevastausta odottavassa” muodossa, koska en ole vielä keksinyt, kuinka ne voisi jakaa täsmällisempiin alakohtiin, joiden vastauksista tulee lyhyitä (lyhyisiin on nopea vastata, jos tietää, ja ne on helppo tarkastaa onko oikein vai ei). Jos hyvää jakoa ei löydy, pyri vastaamaan esseessäsi lyhyesti ja ytimekkäästi seuraaviin asioihin: (1) Miksi kysyttävä asia on tärkeä, eli mitä sillä saavutetaan. (2) Mitä asiaan liittyy käytännön tasolla, eli esim. jokin sovellusesimerkki on hyvä tapa osoittaa ymmärrys tästä. (3) Miten asian voi käyttöjärjestelmässä pääpiirteittäin toteuttaa, eli esittele tietorakenteet ja riittävät pseudokoodit.
- Tässä mainittujen “selitä mitä tarkoittaa X” -tyyppisten kysymysten lisäksi saatan kysyä muitakin luentomateriaalissa boldattuja termejä! Eli koetetaan tälläkin tavoin motivoida tutustumaan kurssilla esiteltyyn käsitteistöön; näissä tenttitärpeissä ei olisi järkeä mainita erillisenä kysymyksenä kaikkia termejä (yli 100 kpl; kappas, näköjään kurssista saa irti aika paljonkin alan sanastoa, jolla on sitten helpompi kommunikoida jatkossa!). Pidän järjen kädessä, eli kysyn vain tärkeimmistä käsitteistä, joiden tärkeys toivottavasti on tullut kurssilla ilmi.

Kysymysten jaottelu mukaillee tässä kesän 2011 kirjallisen materiaalin otsikointia. (P.S. Olen nähnyt erään “ulkoa opeteltavien vastausten” listan vastaaviin kysymyksiin... se oli asiasisällöltään kokolailla päin p:ttä, joten nöyrästi suosittelen vaan miettimään ihan itse ja kysymään epäselvistä ennen tenttiä:))

Contents

- Yleistä tenttitärpeistä
- Tietokonelaitteisto
- Konekielisen ohjelman suoritus
- Käyttöjärjestelmän kutsurajapinta
- Prosessi ja prosessien hallinta
- Yhdenaikaisuus, prosessien kommunikointi ja synkronointi
- Muistinhallinta
- Oheislaitteiden ohjaus
- Tiedostojärjestelmä
- Käyttöjärjestelmän suunnittelusta
- Unixin ja shellin käyttö ja skriptit

Tietokonelaitteisto

Von Neumann -arkkitehtuuri on “esitieto” eikä sinänsä Käyttöjärjestelmät-kurssin asia, eli laitteistosta sinänsä ei tulla kysymään paljonkaan. Kuitenkin ymmärrät laitteiston rakenteen ja käsitteet (materiaalissa boldatut sanat), joita käytät monissa vastauksissasi, jos ne ovat niin sanotusti hyviä vastauksia. Seuraavat kysymykset ovat kuitenkin sisällön kannalta olennaisia:

Kysymys:

- a) Mitä rekisterejä prosessorissa täytyy vähintään olla?
- b) Mihin kutakin näistä välttämättömistä rekistereistä käytetään?
- c) Mitä muita muistikomponentteja tietokoneessa on, ja mitkä näiden merkittävät erot ovat (ts. selitä “muistihierarkia”)?

(1+1+2 = 4p)

Kysymys:

- a) Missä toimintatiloissa prosessorin täytyy vähintään pystyä olemaan?
- b) Miksi?
- c) Miten tämä vaikuttaa prosessorin käskykannan ja rekisterien käyttöön?

(1+1+1 = 3p)

Kysymys:

- a) Mikä on prosessorin lippurekisteri?
- b) Anna kolme erilaista esimerkkiä tilanteista, joissa lippurekisterin sisältö muuttuu (siis milloin muuttuu ja millä tavoin).

(1+1 = 2p)

Kysymys:

Miten käskyosoiterekisteri IP muuttuu seuraavissa tilanteissa:

- a) siirtokäsky, jossa rekisterin sisältö kopioidaan toiseen
- b) ehdoton hyppykäsky
- c) ehdollinen hyppykäsky
- d) aliohjelmakutsu
- e) paluu aliohjelmasta
- f) ohjelmoitu keskeytys (int/trap/svc)

(2p)

Kysymys:

Selitä termi “käskykanta-arkkitehtuuri”.

(1p)

HUOM: Tällaisia yhden pisteen arvoisia termin selityksiä voi tulla mistä tahansa materiaalissa **täten korostetusta** termistä. Vastauksena ei odoteta romaania, vaan lyhyt ja ytimekäs, yhden pisteen arvoinen selvitys.

Konekielisen ohjelman suoritus

Kysymys:

- a) Esitä prosessorin nouto-suoritussykli (mukaanlukien keskeytyskäsitely) kaaviona
- b) selitä lyhyesti, mitä syklin missäkin vaiheessa tapahtuu.

(2p)

Kysymys:

- a) Mihin eri alueisiin (käyttötarkoituksen mielessä) ohjelman tyypillisesti käyttämä muisti jakaantuu?
- b) Selitä, mikä on virtuaalimuistiavaruus.
- c) Selitä, mikä on “osoitin” ja mitä se yksinkertaisimmillaan tarkoittaa konekielessä.

(3p)

Kysymys:

- a) Mitkä rekisterit x86-64 -arkkitehtuurissa liittyvät läheisimmin pinomuistin käyttöön? (jos et muista nimiä, kerro rekisterien merkitykset)
- b) Mitä pinomuistille ja prosessorin rekistereille tapahtuu, kun x86-64 -arkkitehtuurin mukaisessa ohjelmassa tapahtuu `call` käskyn suoritus eli aliohjelmaan siirtyminen?
- c) Vastaavasti `ret` -käskyn kohdalla eli aliohjelmasta paluu?
- d) Jos aliohjelma tarvitsee paikallisia muuttujia, miten sen pitää käyttää a-kohdassa mainitsemasi rekisterejä?

(4p -- edellyttää tarkkaavaista vastausta, jossa huomioit sopivassa kohtaa kaikki osallistuvat rekisterit (myös RIP) sekä muistat kertoa kaikki niissä tapahtuneet muutokset!)

Kysymys:

Selitä termi “pinokehys” eli “aktivaatitietue”.

(1p)

Kysymys:

Oletetaan, että x86-64 -prosessori (jossa on 64-bittiset rekisterit, ja muistiosoitteet ovat 8-bittisten tavujen osoitteita) suorittaa seuraavan ohjelmakoodin (HUOM: koodi voi vaihdella tenttien välillä; se on aina jotakin suurin piirtein yhtä yksinkertaista ja kommentoitu rivi riviltä suomeksi; tarvitaan siis tarkkaavaisuutta, suttupaperia ja käsitys siitä kuinka prosessori ja muisti toimivat):

```
    movq  $10, %rcx    # aseta laskurin arvoksi 10
@alku:
    pushq %rcx        # vie laskurin sisältö pinoon
    dec   %rcx        # vähennä laskuria yhdellä
    jnz   @alku       # alkuun, jos CX ei mennyt nolllaksi
```

- a) Mikä kaikki on muuttunut prosessorissa ja muistissa (missä siellä?) kun ohjelmanpätkä on suoritettu loppuun? Pyri huomioimaan kaikki tapahtuneet asiat sillä oletuksella, että yhtään keskeytystä ei tapahdu suorituksen aikana.
- b) Mitä on muistipaikassa `16(%rsp)` suorituksen jälkeen?
- c) Selitä mitä prosessori tekisi välittömästi (ennen seuraavan suoritussyklin alkua), jos `pushq %rcx` -käskyn jälkeen tulisi keskeytyspyyntö laitteistolta (eli kuvaile FLIH -toiminta pääpiirteissään, olennaisilta osiltaan). Selvitä prosessorin mahdollisista toimintatiloista riippuvat erityistapaukset tekemättä mitään oletuksia, ja kerro mistä kohtaa prosessori jatkaa suoritusta käskyn jälkeen missäkin näistä erityistapauksista.

(suurin piirtein $1+1+2 = 4$ p)

Kysymys:

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

- a) siirtymisen aliohjelmaan (`call`)
- b) paluun aliohjelmasta (`ret`)

(1p)

Käyttöjärjestelmän kutsurajapinta

Kysymys:

Mihin tarvitaan keskeytyksiä?

(1p)

Kysymys:

Selvitä keskeytyskäsitteilyn perusmekanismeja:

- a) Miksi ylipäättään keskeytyksiä tarvitaan?
- b) Kerro kolme esimerkkiä keskeytyksestä: mistä keskeytys tuli, miksi, ja millä tavoin?
- c) Olipa syy keskeytykseen mikä tahansa, niin mitä prosessori tekee keskeytyksen tultua (kuvaile siis FLIHin tärkeimmät osuudet)?
- d) Selitä jokin käyttöjärjestelmän keskeytyskäsitteilytoimenpide (sopivalla tarkkuudella, joko sanallisesti tai pseudokoodina).

(4p)

Kysymys:

Luettele (nimeltä) kahdeksan käyttöjärjestelmän palvelua, joita kutakin voisi vastata yksi käyttöjärjestelmäkutsu.

(2p)

Kysymys:

Miten käyttöjärjestelmän palveluiden käyttö tapahtuu? Eli miten palvelun käyttö näyttää:

- a) sovellusohjelmoijan lähdekoodissa
- b) prosessorilaitteiston toimenpiteenä
- c) käyttöjärjestelmän toteutuksessa.

(2p)

Prosessi ja prosessien hallinta

Kysymys:

Mitä käyttöjärjestelmän yhteydessä tarkoitetaan sanalla “prosessi”?

(1p)

Kysymys:

Mikä on käyttöjärjestelmän prosessitaulu ja mitä se sisältää?

(2p)

Kysymys:

- Mikä on prosessielementti (PCB)?
- Kuka tai mikä käyttää prosessielementtiä ja mihin?
- Mitä prosessielementti sisältää?

(2p)

Kysymys:

Voit käsitellä luennoilla tai materiaalissa esitetyssä laajuudessa, oppikirjan esittelemänä konkreettisina esimerkkeinä, tai siinä laajuudessa kuin käsittääksesi on välttämätöntä järkeväen moniajon toteutumiseksi:

- Missä eri tiloissa prosessi voi olla?
- Milloin tapahtuu mikäkin siirtymä tilojen välillä?

(1+2 = 3p)

Kysymys:

Täydennä seuraavan “pseudo-C-kielisen” ohjelman kommentit. Aliohjelma luekomento() lukee arvot parametreilleen, pid on kokonaisluku:

```
/* 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 käyttäjärjestelmäkutsua,
        * jonka tehtävä on ... [VASTAUKSESI c-kohta].
        */
if (pid > 0) {
    /* Suorituksen saapuessa tähän lohkoon, tiedetään
    * käyttäjärjestelmän tilasta, että ... [VASTAUKSESI d-kohta]
    * ja tästä prosessista, että ... [VASTAUKSESI e-kohta]
    */
    status = wait(); /* Tämän rivin käyttäjärjestelmäkutsun wait()
        * tehtävä on ... [VASTAUKSESI f-kohta]
        */
} else if (pid == -1) {
    /* Suorituksen saapuessa tähän lohkoon, tiedetään
    * käyttäjärjestelmän tilasta, että ... [VASTAUKSESI g-kohta]
    * ja tästä prosessista että ... [VASTAUKSESI h-kohta]
    */
    exit(1)          /* Tämän rivin käyttäjärjestelmäkutsun exit()
        * tehtävä on ... [VASTAUKSESI i-kohta]
        */
} else {
    /* Suorituksen saapuessa tähän lohkoon, tiedetään
    * käyttäjärjestelmän tilasta, että ... [VASTAUKSESI j-kohta]
    * ja tästä prosessista että ... [VASTAUKSESI k-kohta]
    */
    exec(komento, parametrit);
    /* Edellisen rivin käyttäjärjestelmäkutsun exec() tehtävä on
    * ... [VASTAUKSESI l-kohta]
    */
    /* Niinpä ollen seuraavan rivin suoritus
    * ... [VASTAUKSESI m-kohta]
    */
    printf("%d",pid);
}
}

```

(4p)

Kysymys:

- a) Selitä mitä tarkoittaa “konteksti”
- b) Selitä mitä tarkoittaa “kontekstin vaihto”

(2p)

Kysymys:

Mikä on säikeen ja prosessin ero?

(1p)

Yhdenaikaisuus, prosessien kommunikointi ja synkronointi

Kysymys:

Luettele (nimeltä) neljä erilaista prosessienvälisen kommunikoinnin (IPC) menetelmää.

(1p)

Kysymys:

Mitä ovat signaalit, ja miten niitä käytetään?

(2p)

Kysymys:

Miten tapahtuu viestinvälitys `send()` ja `receive()` -käyttöjärjestelmäkutsuilla?

(2p)

Kysymys:

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

- Millainen ongelma tässä voi tulla, ja missä tilanteessa?
- Kerro, miten käyttäisit semaforia ongelman ratkaisemiseen (koodia ei ole pakko kirjoittaa, kunhan selität ratkaisuidean).

Koodi:

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

(1+2 = 3p)

Kysymys:

- Millainen tietorakenne on semafori (siis mitä tietoja se sisältää)?

- b) Nimeä (tai kuvaile, jos et muista nimiä) kaksi erilaista tarkoitusta, joihin semaforia voi käyttää.
- c) Miten on toteutettu kaksi tärkeintä semaforiin liittyvää palvelua eli `wait()` ja `signal()`? Mieluiten kirjoita pseudokoodi; vähintäänkin kuvaile toimenpiteet tarkoin.

(2p)

Kysymys:

Selitä käsite “kriittinen alue”.

(1p)

Kysymys:

- a) Selitä poissulkuongelma: millaisissa tilanteissa se voi tulla eteen, ja mitä se esimerkiksi voi käsittelemättömänä aiheuttaa.
- b) Esitä C-ohjelma tai pseudokoodi, jossa poissulku tehdään käyttäen käyttöjärjestelmän semaforipalveluita.

(2p)

Kysymys:

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

(3p)

Kysymys:

- a) Anna esimerkki tilanteesta, jossa voi tapahtua lukkiutuminen eli deadlock -tilanne.
- b) Täsmennä, missä olosuhteissa esimerkissäsi tapahtuu lukkiutuminen.

(2p)

Muistinhallinta

Kysymys:

Tässä riittää käsittely kurssimateriaalin kuvailemassa laajuudessa (jos tiedät enemmän, se on hyvä, mutta älä yritä kirjoittaa sitä novellia tähän).

- a) Mihin nykyaikaisen käyttöjärjestelmän ja laitteiston muodostamaan kokonaisuuteen liittyvät käyttöjärjestelmän tietorakenteet nimeltä “sivutaulu” ja “kehystaulu”?
- b) Mikä tämän kokonaisuuden tavoite on? Siis miksi se on kehitetty?
- c) Mikä on sivutaulu, ja mitä tietoja se sisältää?
- d) Montako sivutaulua ainakin on olemassa kullakin hetkellä?
- e) Mikä on kehystaulu, ja mitä tietoja se sisältää?
- f) Montako kehystaulua ainakin on olemassa kullakin hetkellä?
- g) Kuvaile, missä tilanteissa ja mihin tarkoituksiin näiden eri taulujen eri tietoja tarvitaan?

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

Kysymys:

- a) Mitä tarkoittaa muistihierarkia?
- b) Mitä tarkoittaa lokaalisuusperiaate muistinhallinnan yhteydessä?

(2p)

Kysymys:

Selitä (pseudokoodina tai sanallisesti), miten käyttöjärjestelmä käsittelee sivunvaihtokeskeytyksen (page fault). Oleta että kehysten valintamenetelmänä on LRU (least-recently-used).

(3p -- edellyttää vastausta, jossa kerrotaan, mitä tietorakenteita käytetään ja miten; ts. ei tosiaan saa täysiä pisteitä vastaamalla että “vaihtaa sivun”!!)

Oheislaitteiden ohjaus

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 seuraavasta käskystä. (Tyyli vapaa - kaaviot, pseudokoodit, essee käyvät).

(3p)

Kysymys:

Megatavun mittaisen JPG-kuvan lukeminen kovalevyiltä 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 seuraavasta käskystä. (Tyyli vapaa - kaaviot, pseudokoodit, essee käyvät).

(3p)

Kysymys:

Selitä DMA (direct memory access) -järjestelmän periaate sekä sen edut ja haitat.

(2p)

Kysymys:

Miksi tarvitaan erikseen laiteriippuva ja laiteriippumaton I/O -ohjelmisto?

(1p)

Tiedostojärjestelmä

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 kovalevyiltä löytyisi tiedosto nimeltä:
`/home/jokuope/valokuvat/kurssi.jpg`

(3p)

Käyttöjärjestelmän suunnittelusta

Kysymys:

Luettele (ja selitä lyhyesti) tavoitteita käyttöjärjestelmän suunnittelussa erityisesti prosessorin käyttöön ja vuoronnuksen liittyen.

(2p)

Kysymys:

- a) Selitä, mitä tarkoitetaan “reaaliaikajärjestelmällä”. Anna esimerkki sellaisesta.
- b) Mitä erityisiä tavoitteita reaaliaikajärjestelmälle voidaan asettaa?

(3p)

Kysymys:

- a) Miten microkernel -käyttöjärjestelmä eroaa monoliittisesta?
- b) Mitä etuja microkernel -mallissa on?

(2p)

Kysymys:

Mitkä ovat käyttöjärjestelmän päätehtävät?

(1p)

Kysymys:

Kuvaile jokin hierarkkisesti tai lähes hierarkkisesti rakennettu käyttöjärjestelmätoteutus karkeimpien moduulien tasolla.

(1p)

Unixin ja shellin käyttö ja skriptit

Kysymys:

- a) Selitä, mikä on “interaktiivinen shell”.
- b) Mikä on shell-skripti?

(1p)

Kysymys:

Luettele kolme tarkoitusta, joihin shell-skriptejä voidaan käyttää.

(1p)

Kysymys:

Mitä seuraava shell-komentorivi tekee:

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

(1p)

Kysymys:

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

```
#!/bin/sh  
echo ‘$0‘
```

b) Mikä on lopputulema tämän skriptin ajamisesta?

(1+1=2p) (Vastaus pääsi lipsahtamaan luennon 14 loppupuolella, joten pidetään kysymys tämän vuoden valikoimassa, vaikkei näitä aiheita käsitelty kurssilla aiemmin. Vinkki: ÄLÄ kokeile tätä koneella, jossa on muita käyttäjiä, koska sen jälkeen susta ei tykkää juuri kukaan! :-))

Kysymys:

Tee bash-skripti, joka testaa onko argumenttina annettu hakemisto olemassa. Esimerkiksi skripti voitaisiin ajaa komennolla:

```
loytyyko hake/alihake
```

ja tulostus olisi kyllä tai ei sen mukaan kumpi on tilanne.

(1p)

Kysymys:

Tee bash-skripti, joka lukee päätteeltä yhden rivin tekstiä ja lisää sen tiedoston ~/ajatuksia loppuun. Skriptin tulee ensin tarkistaa, onko kyseinen tiedosto olemassa ja onko siihen kirjoitusoikeus. Jos ei, niin molemmissa tapauksissa on tulostettava tilannetta kuvaava virheilmoitus ja lopetettava skripti ennen kuin päätteeltä luetaan mitään.

TAI VASTAAVIA SIMPPELEITÄ BOURNE AGAIN SHELL (bash) SKRIPTEJÄ!

(2p)

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
```

Mikä on vialla ja miten voin korjata asian?

(1p)

Kysymys:

Tee pienoinen bash-skripti, jolla saa apuohjelmaa `convert` käyttäen tehtyä kaikista oleskeluhakemistossa olevista `.JPG` -päätteisistä kuvista 60 pikseliä leveät versiot joiden nimi on mallia `pieni_IMG_1943.JPG` Ei tarvitse tehdä mitään oikeellisuustarkistuksia. Ohje `convertin` käyttöön:

```
-resize 60x LÄHDETIEDOSTO KOHDETIEDOSTO
```

(1p)

Kysymys:

Tee pieni bash-skripti, jolla saa muunnettua kaikkien oleskeluhakemistossa olevien .JPG -päätteisten tiedostojen nimet päättymään .jpg eli pienin kirjaimin. Komento, joka tulostaa argumenttina annetun tiedostonimen alkuosan, toimii seuraavasti:

```
basename .JPG JOTAKIN.JPG
```

Komento, jolla tiedoston nimi muutetaan, on:

```
mv -p ALKUPERAINEN_NIMI UUSI_NIMI
```

Vinkki: tarkkana hipsukoiden ym. kanssa.

(1p)

Kysymys:

Mitä voi tehdä apuohjelmalla `grep`?

(1p)

Kysymys:

Mitä voi tehdä apuohjelmalla `find`?

(1p)