

ATK tähtitieteessä

765307A-01 – osa Tähtitieteen tutkimusprojekti 1 -kurssia!

Heikki Salo (heikki.salo@oulu.fi, MA307) ja

Aku Venhola (aku.venhola@oulu.fi, MA303)

Aiheet;

- IDL-ohjelmointi (HS)
- LaTeX-dokumenttien luominen (HS)
- Python-ohjelmointi (AV)

Luentokalvot, esimerkkiohjelmat saatavilla [www-sivulla](http://www.oulu.fi/astronomy/TUTKIMUSPROJEKTI2021/)
<https://www.oulu.fi/astronomy/TUTKIMUSPROJEKTI2021/>

(käyttäjätunnus=testi, salasana= testi)

ATK-osuuden (sisältää kotiharjoitustyön) jälkeen alkaa varsinainen tutkimusprojekti. Kurssin arvosana määräytyy kotiharjoitustyön (20%) ja tutkimusprojektin (80%) perustella.

Aikataulu

Luennot (YL124)

07.09.21	ti	12-16	Linux, emacs (HS)
08.09.21	ke	12-16	IDL
14.09.21	ti	12-16	IDL
15.09.21	ke	10-14	IDL
21.09.21	ti	12-16	IDL
22.09.21	ke	10-14	IDL-kotiharjoitustyö, varsinaisen tutkimusprojektin aihevaihtoehtojen esittely
28.09.21	ti	12-16	Python (AV)
29.09.21	ke	10-14	Python (AV)
05.10.21	ti	12-14	Python (AV)
06.10.21	ke	10-12	Python (AV)

HUOM!

Tämä ei ole varsinainen ohjelmointikurssi!

Tämän kurssin tarkoituksena on esitellä työkaluja, jotka ovat tähtitieteessä hyödyllisiä, ja joita ei välttämättä käsitellä ohjelmointikursseilla.

Nämä ovat niitä työkaluja, joita käytetään tähtitieteen kurssien harjoitustöissä tai demonstraatioissa. Esim.

Taivaanmekaniikan kurssi (IDL, syksy 2021)

Aurinkokunnan dynamiikka (IDL, kevät 2022)

Galaksit (python, syksy 2022)

...

Kurssilla käsitellään suppeasti:

- Linux-käyttöjärjestelmä: yleisiä ominaisuuksia, komentorivin käyttö, muutamia ohjelmia
- IDL-ohjelmointikielen perusteita (Interactive Data Language). Pienten ohjelmien laatiminen, miten kuvat pdf-tiedostoon
- LaTeX ladontajärjestelmä raporttien ja työselostusten luontia varten. Esim. pdf-kuvien liittäminen
- Python-ohjelmointikielen perusteita ja tähtitieteellisiä ohjelmia

Lisämateriaali

- Linux.fi wikisivustot (<http://linux.fi/wiki>)
- IDL:
 - Aiempien kurssien materiaali (hs.pdf vuodelta 1990!)
 - The IDL Astronomy User's Library
<https://idlastro.gsfc.nasa.gov/>
 - Lista tähtitieteen IDL-sivustoista
https://idlastro.gsfc.nasa.gov/other_url.html
 - Esim: <http://www.idlcoyote.com/gallery/index.html>
- LaTeX: Tobi Oetiker's *The not so Short Introduction to LaTeX2e* <https://tobi.oetiker.ch/lshort/>

Tähtitieteilijöiden käyttämiä ohjelmistoja

(lähde: Pertti Rautiaisen luentokalvot)

Momcheva & Tollerud 2015 tekivät kyselyn tähtitieteilijöille heidän käyttämistään ohjelmistoista (1142 vastausta):

- *90% vastaajista oli kirjoittanut ainakin osan käyttämistään ohjelmista itse.*
- 8% oli saanut laajamittaista koulutusta ohjelmistokehityksestä.
- 43% ei ollut saanut mitään ohjelmistokoulutusta!
- Työkalujen Top 10 ovat: *Python, shell-skriptit, IDL, C/C++, Fortran, IRAF, taulukkolaskenta, HTML/CSS, SQL ja Supermongo*

Python #1, IDL #3, Matlab on #11

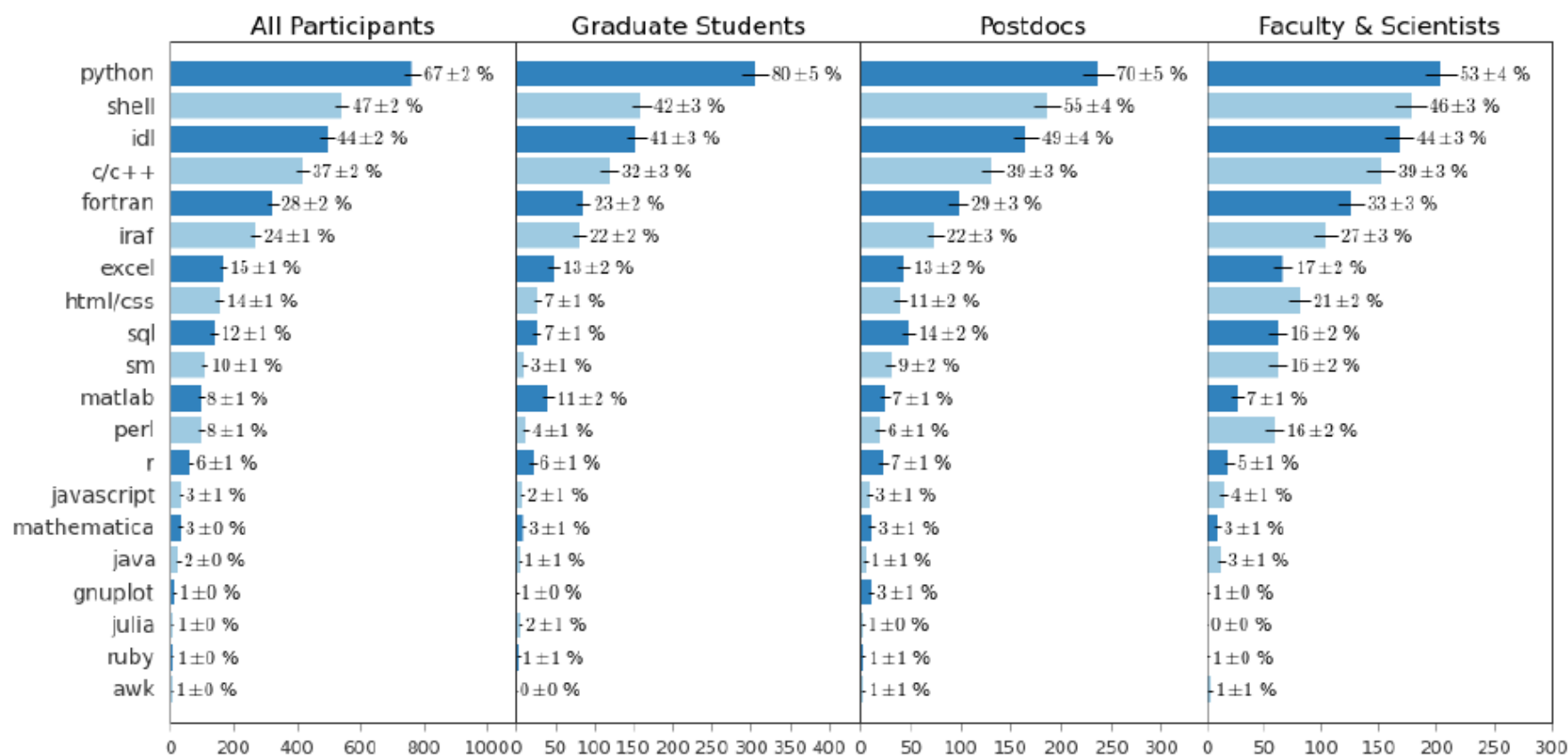


FIG. 10.— Responses to the prompt “Select any of these that you regularly use in your research”, sub-divided by career stage. The options listed included: IDL, IRAF, Python, C, Fortran, Perl, Javascript, Julia, Matlab, Java, R, SQL, Shell Scripting, STAN, Figaro, Ruby, HML/CSS, Supermongo (labeled “sm”), and Excel or other spreadsheets (labeled “excel”). Respondents could add additional tools not listed using an “Other” box. Among the tools in this plot, four items were added by respondents: C++, Mathematica, gnuplot and awk. Note that the x axis varies between panels.

Joitakin kommentteja

Tulokset eivät riippuneet voimakkaasti uravaiheesta tai maantieteellisestä sijainnista – tähtitieteilijät näyttävät muodostavan maailmanlaajuisen yhteisön.

Monet listalla olevista työkaluista eivät ole varsinaisesti matemaattisia tai data-analyysissä käytettäviä ohjelmistoja, mutta ne ovat silti oleellinen osa esim. havaintodatan käsittelyprosessia.

Pythonin käyttö jatkuvasti kasvamassa!

Tekstinkäsittely

Tieteelliset artikkelit: LaTeX-koodia, joka kirjoitetaan jollakin editorilla.

Editorit: ohjelmia joilla voi käsitellä ns. pelkkiä ASCII-tekstitiedostoja, esim. ohjelmalistauksia, skriptejä, asetustiedostoja.

Unix-maailmassa käytettyjä editoreja ovat esim. *Emacs*, *pico*, *gedit* ja vi

Matemaattisen tekstin tuottaminen esim. LibreOfficen kaavaeditorilla on turhan työlästä – eikä jälkikään ole kovin kaunista.

Monet tieteelliset lehdet toivovat, että käsikirjoitukset tehdään käyttäen niiden omaa LaTeX-makropakettia.

LaTeX on saatavissa käytännössä kaikkiin käyttäjäjärjestelmiin.

LaTeX-koodi on puhdasta ASCII-tekstiä, jonka saa aina auki.

LaTeX:illa tehtyihin dokumentteihin voi liittää kuvia useissa formaateissa.

Ohjelmointi

Tärkeimmät ohjelmointikielet tähtitieteessä ovat Fortran ja C/C++.

Ilmaisia kääntäjiä: gcc, gfortran.

Tulkkaavien kielten suosio on kasvussa: IDL, Python, Matlab ja Mathematica ovat näistä yleisimpiä.

Suurta numeronmurskaustehoa tarvittaessa turvaudutaan edelleen Fortraniin ja C:hen.

Havaintojen käsittely ja esittäminen

Useita tähtitieteellisten havaintojen käsittelyyn suunniteltuja paketteja: IRAF, MIDAS, AIPS jne.

Tulosten visualisoinnissa IDL (Interactive Data Language) on tehokas yleistyökalu.

Kuvien käsittelyssä käytettäviä ohjelmia ovat myös Gimp ja Adobe Photoshop.

Unix/Linux tunnissa

Unix on tietokoneen käyttöjärjestelmä, tarkemmin sanottuna monen käyttäjän moniajokäyttöjärjestelmä.

Tärkeimpiä kaupallisia Unixeja olivat AIX (IBM), HP-UX (HP), Solaris (Sun) jne.

Suluissa oleva kohta osoittaa, että ne olivat yleensä laitevalmistajan omia variantteja, olkoonkin että loppukäyttäjälle erot olivat yleensä pieniä.

Yhtenä Unix-varianttina voidaan pitää myös Linuxia, joka on avoimeen koodiin perustuva käyttöjärjestelmä joka on saatavissa useille laiteympäristöille.

Jos mennään halkomaan hiuksia, niin Linux on vain käyttöjärjestelmän ydin, suuri osa muista ohjelmista on peräisin esim. GNU-projektista. Joskus käytetäänkin nimitystä GNU/Linux.

Käytettävän käyttöjärjestelmäversion saa näkyviin komennolla:

`uname -a`

Linux-yleistä

- Monia eri jakelupaketteja:
Fedora linux, Ubuntu, Mint, Debian
- Perinteisesti pidetty vaikeana:
voidaan käyttää samalla tavalla työpöydällä kuin
Windows tai Mac OS X
-valittavana monia eri graafisia käyttöliittymiä, jotka luovat
työpöytäympäristön (Gnome, KDE...)
- Tieteellisessä käytössä yleinen (open source, ilmainen)

Työpöytäympäristö

Vaikka Unixia voidaan käyttää pelkässä tekstipääteyhteudessa, niin nykyään useimmiten käytetään apuna ikkunointijärjestelmää.

Unixissa/Linuxissa ikkunointijärjestelmänä on yleensä X11 (yleensä puhutaan X:stä).

Karuhkon perus-X:än sijaan nykyään on yleensä X11:n päällä pyörivänä graafisena työpöytäympäristönä Gnome, KDE, Unity tai esim. XFCE.

Loggautuminen Unix/Linux-koneeseen

Login-prompti joko graafisena käyttöliittymänä tai komentorivillä:

`login:`

Tai

`Username:`

Käyttäjätunnuksen ilmoittamisen jälkeen kone kysyy salasanaa:

- mieluusti vähintään 8 merkkiä (nykyisin minimi 12)
- sekaisin isoja ja pieniä kirjaimia+numeroita
- ei sanakirjasta
- ei kuviota näppäimistöllä.

Jos kone ei käynnistä automaattisesti graafista käyttöliittymää, niin tämän voi tehdä komennolla `startx`.

Koneesta poistuminen tapahtuu komennolla `exit` tai (tai tekstiyhteyden tapauksessa myös `logout`).

Jos käyttää graafista käyttöliittymää, niin `exit` komentoriviltä annettuina tarkoittaa vain kyseisestä terminaalista poistumista, tämän lisäksi täytyy loggautua erikseen pois valikosta käsin.

HUOM: Linux:ssa isot ja pienet kirjaimet merkitsevät eri asioita!

Yhteydet muihin koneisiin

Yhteydet muihin Unix-koneisiin kannattaa tehdä käyttäen ssh-ohjelmaa, joka kryptaa käytettävän viestiliikenteen.

Jos käyttäjätunnus logattavassa koneessa on sama, niin voi kirjoittaa `ssh kone`.

Jos käyttäjätunnus on toinen, niin sitten voi loggautuakseen kirjoittaa `ssh kone -l tunnus`.

Tai vaihtoehtoisesti `ssh tunnus@kone`

X-ikkunoinnin forwardointi ei ole välttämättä automaattisesti päällä
→ jos haluaa käyttää toisessa koneessa sijaitsevia graafisia sovelluksia, yhteys otettava seuraavasti: `ssh -X tunnus@kone`.

Unix/Linux-koneiden etäkäyttö onnistuu myös Windowsille tehdyistä ssh-toteutuksista. Graafisten ohjelmien käyttö onnistuu esim. Cygwinin xterminaalista käsin.

Komentotulkki

Kun Unix-koneeseen ollaan loggauduttu, niin käyttäjän kirjoittamia komentoja käsittelee ns. komentotulkki (engl. shell).

Yleisiä komentotulkkeja ovat *bash*, *tcsh*, *csch* ja *zsh*.

Näistä yleisin on nykyään *bash*, joka on myös oletuksena yl124-luokan koneissa. Harjoituksissa käytettävä materiaali pohjautuu suurelta osin *tcsh*-tulkkiin, jonka saa käyttöön kirjoittamalla **tcsh**

Kun käyttäjä loggautuu koneeseen, käyttöjärjestelmä suorittaa joitakin alustustiedostoja.

tcsh:ssa näitä ovat mm. *.login* ja *.cschrc*

*bash*issa taas *.bashrc*.

Jos käyttäjällä ei ole omia versioitaan ko. tiedostosta, niin silloin käyttöön otetaan tiedostojen oletusversioita.

Alustustiedostossa määritellään hakupolkuja ja uusia komentoja:

Esimerkkinä **alias**, eli lyhennysmerkintä vähän monimutkaisemmalle komennolle (tässä tapauksessa tiedostot listaava `ls`-komento kahdella parametrilla):

`.cshrc`:ssa tämän voisi tehdä rivillä

```
alias ll 'ls -al'
```

`.bashrc`:ssä taas rivillä

```
alias ll='ls -al'
```

`.cshrc` tai `.bashrc`-tiedostossa voi olla myös polkuasetuksia (ohjelmakirjastojen sijainti jne.).

Jos koneeseen on asennettu useampi shell, niin sitä voi tilapäisesti vaihtaa kirjoittamalla esim. `bash`:issä `tcsh`

(huom: vaikuttaa ainoastaan ikkunaan, jossa tehty)

Hakemistorakenne ja tiedostot

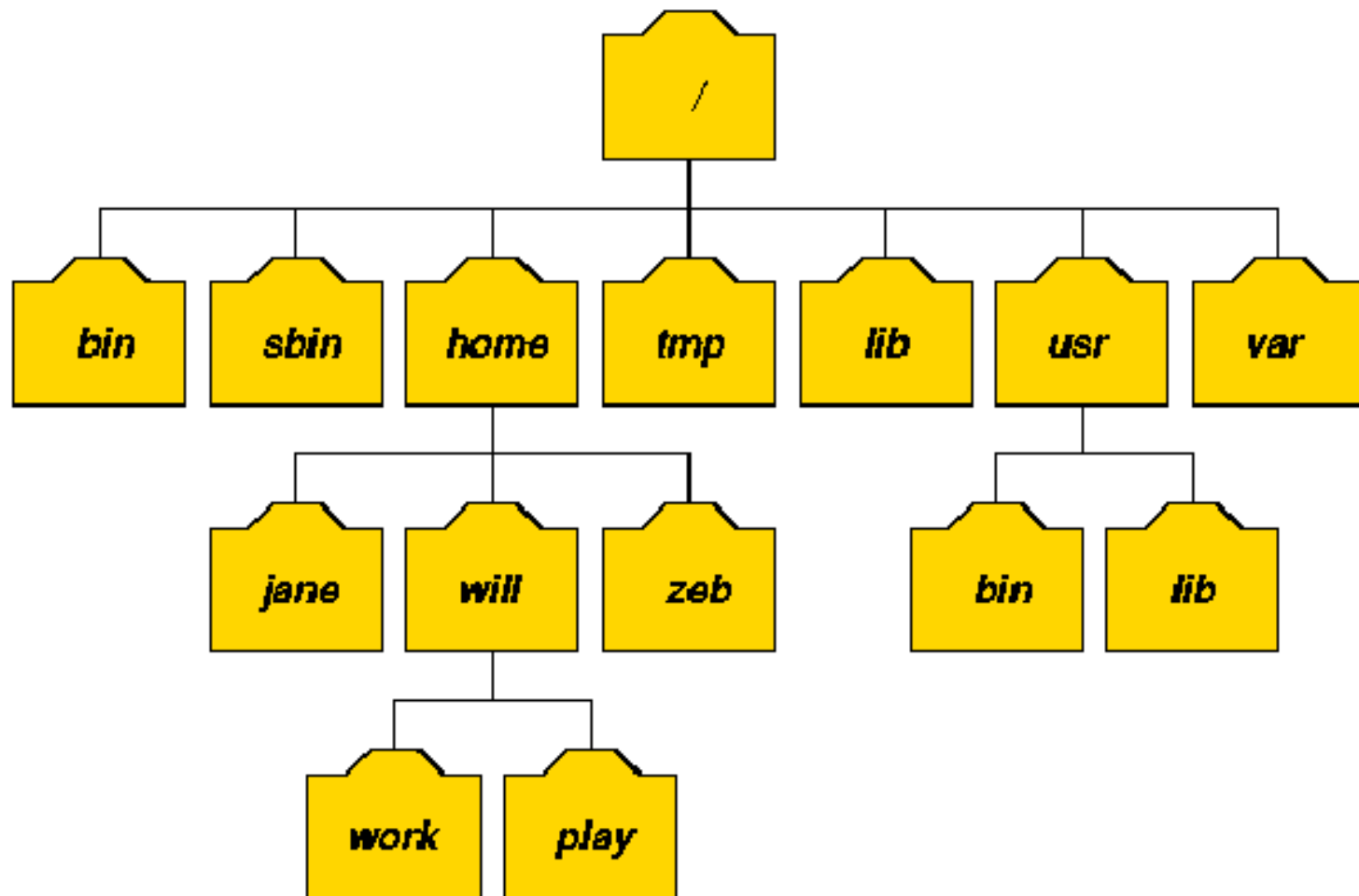
Tieto sijaitsee kiintolevyllä olevissa tiedostoissa.

Käytön selkeyttämiseksi tiedostot on järjestetty hakemistorakenteeseen.

Levyllä on siis erilaisia hakemistoja, joissa voi olla tiedostoja ja edelleen alihakemistoja...

Komennolla **df -h** näkee koneen päähakemistot (ja niiden koot, sekä vapaana olevan tilan).

Hakemistorakenne



Kuva : <http://www.doc.ic.ac.uk/~wjk/UnixIntro/Lecture2.html>

Hakemistorakenne ja tiedostot

Komennolla `pwd` näkee oman sijaintinsa hakemistorakenteessa.

Listauksen hakemiston sisällöstä saa komennolla `ls`.

Pelkän `ls` -komennon antama informaatio on niukkaa, ja enemmän saa komennolla `ls -al`, missä komennolle `ls` annetaan argumentteina optiot `a` ja `l`.

`ls -alrh` → koot, luontipäivät, uusin alimmaisena

Lisätietoa komennosta ja sen argumenteista saa kirjoittamalla `man komento`.

Myös **apropos aihepiiri**

(ja etsimällä verkosta)

Uuden alihakemiston voi luoda komennolla `mkdir` olettaen, että käyttäjällä on kirjoitusoikeus kyseiseen paikkaan hakemistorakenteessa.

Tyhjän hakemiston voi tuhota komennolla `rmdir`.

Yksittäisen tiedoston voi tuhota komennolla `rm`, esimerkiksi
`rm tempfi.`

Hakemistorakenteessa on myös paikkoja, joihin tavallisella käyttäjällä ei ole kirjoittamisoikeutta.

Hakemistosta toiseen pääsee käyttämällä komentoa `cd`.

Kotihakemistoon pääsee pelkällä `cd` komennolla.

Hakemistorakenteen “juureen” pääsee komennolla `cd /`

- Hakemistossa yksi taso ylöspäin komennolla
cd ..
- **Lyhennysmerkintöjä**
nykyinen hakemisto **.**
nykyisen hakemiston yläpuolella oleva hakemisto **..**
käyttäjän kotihakemisto **~**
- Esim: **cd ~/atktati** ja
cd /koti/hsalo/atktati
vievät samaan hakemistoon

Muita tärkeitä komentoja

Tiedoston nimen voi muuttaa toiseksi käyttäen mv-komentoa:

```
mv tiedosto1 tiedosto2
```

Tätä komentoa voi käyttää myös tiedoston siirtämiseen paikasta toiseen:

```
mv /hakemisto1/tiedosto /hakemisto2/tiedosto
```

Tiedoston voi kopioida toiselle nimelle cp komennolla:

```
cp tiedosto1 tiedosto2
```

Tämän jälkeen on olemassa kaksi sisällöltään identtistä tiedostoa.

Jos nimellä tiedosto2 on jo ollut tiedosto, niin se tuhoutuu em. kopiointiprosessissa!

Tiedoston kopioimiseen koneesta toiseen käytetään yleensä scp-ohjelmaa:

```
scp tiedosto tunnus@kone:hakemisto
```

-kopioidaan tiedosto toisen koneen hakemistoon.

```
scp tunnus@kone:hakemisto/tiedosto .
```

-kopioidaan toisen koneen hakemistosta tiedosto siihen hakemistoon, jossa ollaan.

Kokonaisten hakemiston alihakemistoihin voi kopioida antamalla scp-komennolle argumenttin -r:

```
scp -r hakemisto tunnus@kone:.
```

Komennolla cat voi tulostaa (teksti)tiedoston sisällön:

```
cat tiedosto.
```

Koska tiedostot ovat usein pitkiä, edellinen komento ei ole aina kovin kätevä. Komento more toimii paremmin:

```
more tiedosto.
```

Tekstiä voi kelata joko ruutu (välilyönti) tai rivi (Enter) kerrallaan.

Moniin järjestelmiin on asennettu vieläkin kehittyneempi komento

```
less, jolla tiedostoa voi selata molempiin suuntiin.
```

Kahden tiedoston väliset erot voi tulostaa komennolla diff:

```
diff tiedosto1 tiedosto2.
```

Haluttua merkkijonoa voi etsiä tiedostosta komennolla grep:

```
grep hassu tiedosto
```

→ tulostaa tiedostosta ne rivit, jotka sisältävät merkkijonon hassu.

Komento `sort` asettaa tiedoston riveittäin aakkosjärjestykseen (huom! katso man-sivulta komennon muita argumentteja).

Normaalisti edellisten komentojen tulostukset ohjautuvat ruudulle.

Ne voi myös ohjata tiedostoon käyttämällä `>` -merkkiä, esim.

```
sort tiedosto1 > tiedosto2
```

tulostaa tiedoston `tiedosto1` riveittäin aakkostettuna `tiedostoon2`.

Komentoja voi myös putkittaa `|`-merkkiä käyttäen:

```
grep hassu tiedosto |more
```

aiheuttaa sen, että tiedostosta “`tiedosto`” tulostetaan rivit, joilla esiintyy merkkijono “`hassu`”, tämä tulostus ohjataan edelleen `more` komennolle, jolloin sitä voidaan tarkastella rivi/ruutu kerrallaan.

Suuria tiedostoja, joita ei tarvita kovin usein voi kompressoida tiiviimpään muotoon.

`compress tiedosto`

kompressoi tiedoston sisällön ja luo tiiviimmän tiedoston nimellä tiedosto.Z.

Komennolla

`uncompress tiedosto.Z`

tiedosto palautuu alkuperäiseen muotoon.

Useissa järjestelmässä on myös pakkaussuhteeltaan tehokkaampi gnuzip, jonka pakkaus ja purkukomennot ova

`gzip` ja `gunzip`.

Myös zip-ohjelmisto on yleinen eri koneissa.

Järjestelmän yleisen levytilan käytön näkee komennolla `df`.
Yksittäisen hakemiston taas komennolla `du`.

Hakemistossa on usein suuri määrä tiedostoja. Niiden käsittelyssä on usein näppärää käyttää ns. wildcard-merkkejä.

```
cp tiedosto?.txt hakemisto2
```

kopioi kaikki tiedostot, joiden nimessä on sanan “tiedosto” jälkeen vielä yksi merkki ennen osiota “.txt” hakemistoon hakemisto2.

```
cp tiedosto*.txt hakemisto2
```

kopioi kaikki tiedostot, joiden nimessä on sanan tiedosto jälkeen jokin merkkijono ennen osiota “.txt” hakemistoon hakemisto2.

Usein on kätevää arkistoida useita tiedostoja tai jopa kokonaisia hakemistoja yksittäisiksi arkistotiedostoiksi.

```
tar cvf arkisto.tar *.jpg
```

arkistoi tiedostoon arkisto.tar kaikki tiedostot, joiden nimi päättyy merkkijonoon .jpg. (Huomaa järjestys!)

```
tar xvf arkisto.tar
```

puolestaan purkaa arkiston.

Huom! ennen purkamista kannattaa arkiston sisältö tarkistaa:

```
tar tvf arkisto.tar.
```

Myös zip-ohjelmistossa on mukana hakemistorakenteen pakkaaminen yhteen .zip-tiedostoon.

.zip- tai .gz-tiedoston sisällön voi tarkistaa myös graafisessa käyttöliittymässä purkamatta tiedostoa kiintolevyllä!

Yksittäisiä prosesseja voidaan käynnistää myös taustalle käyttämällä & -merkkiä, jolloin niiden käynnistämiseen käytetty terminaali pystyy edelleen vastaanottamaan komentoja, esim.

```
emacs tiedosto &.
```

Jos haluaa muuttaa jo pyörivän prosessin taustaprosessiksi, niin se tapahtuu painamalla ensiksi prosessin käynnistäneessä xterminaalissa Ctrl-z ja kirjoittamalla sitten `bg`.

Käynnissä olevat prosessit näkee komennolla `ps` (huom! tarkista optiot `man-komennolla`).

Jos pyörivä prosessi, esim. jumiin jäänyt terminaali halutaan keskeyttää resurssien vapauttamiseksi, niin se tapahtuu komennolla `kill nro` tappaa prosessin, jonka numero `nro` on saatu komennolla `ps`.

Muista: www.linux.fi/wiki

Lyhyt Emacs-opas:

Olemassaolevan tiedoston tiedosto.txt avaaminen:

emacs tiedosto.txt &

Tiedosto avataan taustalle, omaan ikkunaan; ikkunassa josta komento annettiin voidaan jatkaa uusien komentojen antamista

Huom: jos yhteys on *hidas tai ikkunointia ei voi käyttää*, voit avata tiedoston samaan ikkunaan komennolla **emacs -nw tiedosto.txt**

- Komentoja voidaan antaa Emacs-valikoista tai nopeammin **näppäinyhdistelmillä**, joissa käytetään **control** ja **esc** -näppäimiä

(cntr-näppäintä pidetään pohjassa, esc-näppäintä painetaan ja irroitetaan ennen jatkopainalluksia; eri näppäimet erotettu seuraavassa tavuviivalla)

- Tiedoston talletus: Ctrl-x-Ctrl-s (CTRL-näppäintä voi pitää koko komennon ajan pohjassa!)
- Poistuminen Emacsista: Ctrl-x-Ctrl-c (Emacs kysyy talletetaanko tiedosto muutoksineen).
- Merkkijonon etsiminen: Ctrl-s (pääsee tilaan, jossa ohjelma kysyy merkkijonon)
- Siirtyminen dokumentissa paikasta toiseen tapahtuu nuolinäppäimillä.
- Lisäksi on olemassa mm. seuraavat siirtokomennot
 - Ctrl-a -siirtyminen rivin alkuun
 - Ctrl-e -siirtyminen rivin loppuun

- Laajoissa dokumenteissa joutuu usein siirtymään niin pitkiä matkoja, että nuolinäppäinten käyttö ei ole kätevää. Tällöin voi käyttää ruutu kerrallaan siirtymistä:
- Ctrl-v -ruutu eteenpäin
- Esc-v -ruutu taaksepäin.
-
- Lisäksi voidaan siirtyä:
Esc-< -siirtyminen puskurin alkuun\\
Esc-> -siirtyminen puskurin loppuun.

- Komennolla Esc-x pääsee tilaan, jossa voi kirjoittaa useita erilaisia komentoja. Tärkeimpiä:

replace-string -ohjelma kysyy korvattavan ja korvaavan merkkijonon. Muutokset kohdistuvat koko tekstiin.

goto-line -ohjelma kysyy mille riville siirrytään

-

Tekstin hävittäminen/siirtäminen (kokeile varovasti!)

cntrl-k rivi tai rivin loppu kursorin osoittamasta paikasta poistetaan ('kill')
voidaan poistaa useita rivejä

cntrl-y lisää poistettu teksti kursorin kohdalle ('yank')
teksti voidaan lisätä useaan paikkaan