

L2: linux

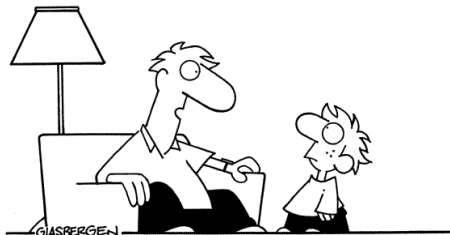
L2: linux (kuva:@www.glasbergen.com)

- ▶ **Tavoite:** Kaikki oppivat **linux**:n perusteet
- ▶ Perusteet jo tutut ⇒ Luennoille ja laskuharjoituksiin osallistuminen vapaaehtoista
- ▶ Yksi monista **linux** alkeisoppaista | www |

linux: Komentotulkki

- ▶ Suorittaa käyttäjän terminaalilta kirjoittamia, komentoja, joita voi “selata” nuolinäppäimillä  ja , sekä täydentää  näppäimellä
- ▶ Jokainen komento tekee yhden asian hyvin: tulostus, tiedon siirtäminen, verkkoyhteys,...
- ▶ “Hyvin” riippuu käyttäjästä, koska valinnanvaraa on loputtomiin
- ▶ Asioita ei tarvitse keksiä uudelleen. Vain pieni osa komentoista opeteltava ulkoa sujuvaa käyttöä ajatellen
- ▶ **Käytännössä:** Opeteltava yksi komento kerrallaan, s.o. mitä milloinkin sattuu tarvitsemaan
- ▶ **Käytännössä:** Monimutkaisemmat komennot kannattaa kirjoittaa muistiin tai ohjelmiin, koska vie aikansa ymmärtää ne ja saada ne toimimaan. **Asenne:** En muista ...? Kerran ratkaistu OK!

Copyright 2002 by Randy Glasbergen. www.glasbergen.com

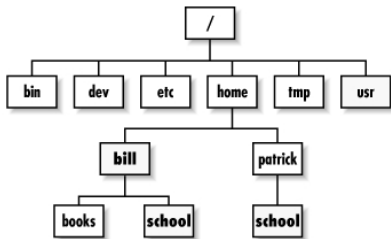


“My side of the family is full-blooded DOS and Windows, but you’re one-quarter Mac with a little Linux and Amiga on your mother’s side.”

L2: linux

Hakemistot (kuva:@www.oreilly.com)

- ▶ **linux** komennoissa isot ja pienet kirjaimet merkittäviä (case sensitive)
- ▶ Kokeile komentoja `pwd` ja `PWD` ja `PwD`
- ▶ Käyttäjän kotihakemisto on `/home/username/`
- ▶ Kotihakemiston pääsee mistä tahansa muusta hakemistosta komennolla `cd`
- ▶ Kotihakemiston voidaan lyhentää merkillä `~` eli sinne pääsee myös komennolla `cd ~`
- ▶ Kotihakemistoon luodut hakemistot ja tiedostot kuuluvat käyttäjälle
- ▶ Kotihakemisto on käyttäjän henkilökohtainen työtila koneella
- ▶ **linux**:n tiedostorakenne on kuin puun (engl. directory tree). Sen juuri (engl. root) on hakemisto `/`. Juureen pääsee komennolla `cd /` Tavallinen käyttäjä ei siellä tosin tee mitään?
- ▶ Käyttäjän oikeudet hakemistopuussa kotihakemiston ulkopuolella oleviin hakemistoihin ja tiedostoihin on tarkkaan rajattu
- ▶ Erilliset kiintolevyt ja muut vastaavat ovat osa hakemistopuuta, eivätkä ulkoisesti eroa hakemistoista



L2: linux

Hakemistot (kuva:@www.linuxforfreshers.com)

- ▶ Tästä lähtien sanaa “komento” käytetään säästeliäästi, koska komennot pyritään kirjoittamaan **violetilla**.

cd Return siirtää kotihakemistoon

- ▶ Fuksiläppäriissä komentotulkkiin ilmestyy `username@fuksi:~$`

`pwd` Return tulostaa `/username/home/`

pwd Return kertoo **työhakemiston** eli hakemiston jossa työskentelet nyt. Sen lyhenne on **./**

1s Return listaa työhakemiston sisällön eli saman tuloksen antaa siis myös 1s ./ Return

- ▶ Kaikkia tiedostoja ei kannata luoda kotihakemistoon. Tiedostot kannattaa luoda / jaotella / sijoittaa / siirtää ... eri hakemistoihin esimerkiksi aiheen mukaan: linux, latex, ohjelmat, ...
- ▶ Komennolla `mkdir` luodaan hakemistoja. Esimerkiksi `mkdir ohjelmat` luo `työhakemistoon` uuden hakemiston `./ohjelmat/`
- ▶ `cd kohdehakemisto` siirtää `työhakemistosta` hakemistoon `./kohdehakemisto`
- ▶ Esimerkiksi yllä annettujen komentojen jälkeen `cd ohjelmat` siirtää `työhakemistosta` äsken luotuun uuteen hakemistoon `./ohjelmat/`



L2: linux

Hakemistot (kuva: @www.rafaelespejo.com)

- ▶ **Absoluuttinen polku** ilmaisee tiedoston tai hakemiston sijainnin hakemistopuussa juuresta lähtien. Esimerkiksi seuraavasti `/home/username/kurssit/`
- ▶ Tähän hakemistoon voi siirtyä mistä tahansa **työhakemistosta** komennolla `cd /home/username/kurssit/`
- ▶ **Suhteellinen polku** ilmaisee tiedoston tai hakemiston sijainnin hakemistopuussa **työhakemistosta** lähtien.
- ▶ Esim: Jos työhakemisto on `/home/username/`, riittää komento `cd kurssit/` siirtymiseen hakemistoon `/home/username/kurssit/`
- ▶ Esim: Jos työhakemisto on `/home/username/`, riittää komento `cd ..` siirtymiseen hakemistoon `/home/`
- ▶ Esim: Jos työhakemisto on `/home/username/` riittää komento `ls ./kurssit/` listaamaan hakemiston `/home/username/kurssit/` sisällön
- ▶ **clear** tyhjentää **linux** komentotulkin näytön

Comandos Básicos I

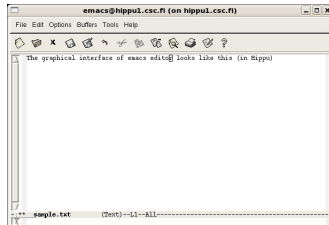
ls
cd
pwd
clear



L2: emacs

L2: emacs (kuvat:@research.csc.fi, @blogg.slu.se)

- ▶ **TAPA 2:** Tiedoston **nimi** editointi
komennolla **emacs nimi &** Return
- ▶ **linux** komentotulkki jää käyttöön
- ▶ Näyttöön ilmestyy editorin “ikkuna”
- ▶ Tiedostoon voi kirjoittaa ...
- ▶ Tallennus Ctrl + x Ctrl + s
- ▶ Sovellutuksia voi “ajaa”
latex nimi.tex Return
python nimi.py Return
octave nimi.m Return
samalla, kun editoi tiedostoa
nimi toivottuun muotoon
- ▶ Tärkeimmät näppäimet
Ctrl Alt
emacs Pikaopaaissa | www |
M = Alt ja C = Ctrl



L2: emacs

L2: emacs

- ▶ **emacs** on **L^AT_EX**, **python** ja **octave** "ystävällinen" tekstieditori
- ▶ Kopioidaan **wget** komennolla tiedosto **H1bkesken.tex** kurssin kotisivulta komennolla
 - **wget http://www.helsinki.fi/jetsu/tila/H1bkesken.tex**
 - Opetellaan samalla **wget** komennon ja "Copy Link Location" sujuva yhdistäminen
 - Aloitetaan editointi **emacs H1bkesken.tex &** Return
 - Tätä tiedostoa voi editoida kuten tahtoo. Alkuperäisen voi hakea aina uudestaan kotisivulta
 - **emacs**: Näyttää **kursorin** rivinumeron. Kursori hyppii välillä. **L^AT_EX** komennot eri värisiä, ...
 - Editoidaan **virhe** mielivaltaisella rivillä ⇒ **emacs** ilmoittaa virheestä (huom: ei aina) ⇒ Tallennetaan **virhe** siitä huolimatta ⇒ Ajetaan **latex H1bkesken.tex** Return ⇒ **L^AT_EX** ilmoittaa millä **H1bkesken.tex** rivillä **virhe** on ⇒ Voidaan löytää ja korjata **virhe**
- ▶ **Pmalli1.py** kotisivulta ⇒ **emacs Pmalli1.py &** Return ⇒ Kursori pomppii, värejä, tabulaattori tuntee komennot, ... ⇒ ... **virhe** ... ⇒ **python Pmalli1.py** Return ⇒ ... **python** ilmoittaa ... ⇒ Voidaan löytää ja korjata **virhe**
- ▶ **Omalli1.m** kotisivulta ⇒ **emacs Omalli1.m &** Return ⇒ Kursori pomppii, ⇒ ... **virhe** ... ⇒ **octave Omalli1.m** Return ⇒ ... **octave** ilmoittaa **virheen** rivin

L2: L^AT_EXL^AT_EX dokumentin tuottaminen

- Kopioidaan **H2bkesken.tex** kotisivulta
- Käytetään **linux** komentotulkissa yhtä aikaa **emacs** ja **evince** ikkunaa ja ajatetaan **L^AT_EX** komentoja

emacs **H2bkesken.tex** & Return

latex **H2bkesken** Return

dvips **H2bkesken -o** Return

evince **H2bkesken.ps** & Return

- **latex** kääntää **.tex**-tiedoston. **dvips** luo **.ps**-tiedoston. **evince** | www | näyttää lopputuloksena syntyvän **.ps**-tiedoston, jota voi lukea, tulostaa, ...
- Yksi monista vaihtoehtoisista tavoista:

emacs **H2bkesken.tex** & Return

pdflatex **H2bkesken** Return

evince **H2bkesken.pdf** & Return

- **pdflatex** kääntää **.tex**-tiedoston, **evince** näyttää **.pdf**-tiedoston, jota voi lukea, ...



L^AT_EX syötetiedoston rakenne

L^AT_EX syötetiedostojen rakenne

- ▶ L^AT_EX syötetiedosto alkaa määrittelemällä toivotut optiot (loputtomasti vaihtoehtoja)

```
\documentclass[optiot]{luokka}
\usepackage[optiot]{paketti}
\usepackage[optiot]{paketti} ...
\begin{document}
...tekstiä...
\end{document}
```

- ▶ Esimerkiksi kotisivun L^AT_EX syötetiedosto **H2bkesken.tex** alkaa riveillä

```
% Latex ohittaa kaiken, mikä on merkin "%" oikealla puolella
\documentclass[A4paper,10pt]{article} % Dokumentin muoto .....
\usepackage[english]{babel}           % Paketti: skandelleille ä,ö,
\usepackage[utf8]{inputenc}            % -"-      : -"- .....
\begin{document}                       % Dokumentti alkaa .....
```

```
Tässä dokumentissa toimivat skandit: \\
''Älyköt öisin ihmettelevät, miksi \LaTeX ~tökkii.''
```

- ▶ Komentit alkavat L^AT_EX ohjausmerkillä **%**. Sen voi kirjoittaa tekstissä komennolla **\%**
- ▶ Muita erikoismerkkejä ovat **\ # \$ & _ ^ ~ { } |**
- ▶ Nämä erikoismerkit ohjaavat L^AT_EX ladontaohjelmaa. Koettakaa ottakaa itse selvää, **kuinka** ne voidaan kirjoittaa L^AT_EX dokumentin sisään, s.e. ne tulostuvat "pelkkänä tekstinä" kuten yllä

Tietokannat

Tietokannat

- ▶ **Ongelma:** Joku Cubarsi on kirjoittanut jotain tähteen Sirius liittyvää vuonna 2014? Pitäisi löytää, lukea ja referoida sitä tulevassa artikkelissa
- ▶ Etsitään **SIMBAD** | [www](#) | tietokannasta tietoa tähdestä **Sirius**
- ▶ Klikataan ja löydetään **CUBARSI R. 2014A&A..561A.141C**
- ▶ Klikataan ja etsitään Cubarsi, ... klikataan **2014A&A..561A.141C**
- ▶ Etsitään ja klikataan **Bibtex entry for this abstract** ja löydetään

```
@ARTICLE{2014A&A...561A.141C,
  author = {{Cubarsi}, R.},
  title = "{Conditions of consistency for multicomponent axisymmetric stellar systems . Is an axisymmetric}",
  journal = {\aap},
  keywords = {galaxies: kinematics and dynamics, solar neighborhood, galaxies: statistics},
  year = 2014,
  month = jan,
  volume = 561,
  eid = {A141},
  pages = {A141},
  doi = {10.1051/0004-6361/201322436},
  adsurl = {http://adsabs.harvard.edu/abs/2014A&A...561A.141C},
  adsnote = {Provided by the SAO/NASA Astrophysics Data System}
}
```

- ▶ **emacs lahteet.bib** & Kopioidaan hiirellä kaikki tuo tiedostoon **lahteet.bib** "Kaikki" tarkoittaa tässä **jokaista** merkkiä väliltä **@ARTICLE{** ja "yksinäinen" } ...
- ▶ Tähän artikkeliin viittaamista jossain **L^AT_EX** dokumentissa harjoitellaan myöhemmin

Ohjelmointi - perusteet

Ohjelmoinnissa tarvitaan

- ▶ Tekstieditori **ohjelmien** lähdekoodin laadintaan: Tällä kurssilla **emacs**
 - ▶ Kääntäjä: Kääntämään lähdekoodin koneen ymmärtämään muotoon
 - ▶ Tällä kurssilla kahden ohjelmointi-kielen alkeet: **python** ja **octave**
- Eräs monista **python** oppaista | [www](#) |
- Eräs monista **octave** oppaista | [www](#) |
- ▶ Ohjelmien lähdekoodi koostuu
- Muuttujista:** | [www](#) | Muuttuvia suureita, kuten lukuja, tekstinpätkiä. ... (Esimerkiksi **a=1**)

Funktioista, aliohjelmista: | [www](#) |

Ottavat vastaan muuttujia. Tekevät niiden avulla jotain toivottua. Luovat ja palauttavat uusia muuttujia, ... (Esimerkiksi: $b = \cos(a)$)

Ohjausrakenteista: | www | Ohjaavat ohjelman toimintaa. Käskevät esimerkiksi toistamaan tiettyä ohjelman kohtaa useita kertoja (Esimerkiksi: **for ...**) tai valitsemaan useista mahdollisista suorituspoluista vain yhden (Esimerkiksi: **if ... elseif**)



python ja octave

python

- ▶ **python**  käynnistää **python** komentotulkin `>>>`
 - ▶ Poistuminen: `>>>`  + 
 - ▶ Kolme tapaa ajaa ohjelmia
- 1: **linux** komentotulkissa
`python file.py` 
 - 2: **python** komentotulkissa
`>>> import os` 
`>>> os.system('python file.py')`

 - 3: `file.py`:n ensimmäisenä rivinä
`#!/usr/bin/env python`
linux komentotulkissa
`./file.py`  Jos ei toimi?

`chmod u+x file.py` 
- ▶ Tällä kurssilla: 1. vaihtoehto

octave

- ▶ **octave**  käynnistää **octave** komentotulkin `octave:*>`
 - ▶ Poistuminen: `octave:*> exit`
 - ▶ Kolme tapaa ajaa ohjelmia
- 1: **linux** komentotulkissa
`octave file.m` 
 - 2: **octave** komentotulkissa (ilman `.m`)
`octave:*>file` 
 - 3: `file.py`:n ensimmäisenä rivinä
`#!/usr/bin/env octave`
linux komentotulkissa
`./file.m`  Jos ei toimi?

`chmod u+x file.m` 
- ▶ Tällä kurssilla: 1. vaihtoehto
⇒ Rivi 1: `#!/usr/bin/env ...` vain malliohjelmissä `Pmalli1.py` ja `Omalli1.m`

python ja octave

python

- ▶ **python** komentotulkissa

```
>>> help('import') Return
```

kertoo mitä komento

import tekee

- ▶ **python** komentotulkissa

```
>>> help() Return
```

vie interaktiiviseen

python manuaaliin

- ▶ Jotkin ongelmat voivat johtua versiosta, jonka saa selville komennolla

```
python --version Return
```

- ▶ Esimerkiksi **python 3.4.0** litania

```
>>> import os
```

```
>>> os.system('python file.py')
```

korvasi **python 2.7.6** komennon

```
>>> execfile('file.py')
```

octave

- ▶ **octave** komentotulkissa

```
octave:*)>help disp Return
```

kertoo mitä komento

disp tekee

- ▶ **octave** komentotulkissa

```
octave:*)>doc Return
```

pitäisi viedä interaktiiviseen

octave manuaaliin

- ▶ Myös tämä ongelma voi johtua versiosta, jonka saa selville komennolla

```
octave --version Return
```

- ▶ Käytännössä **python** & **octave** komentojen sopivia muotoja mihin tahansa tarkoitukseen kannattaa etsiä useista manuaaleista ja webistä

python ja octave

python

- ▶ **python** komentotulkissa

```
>>> aa=2
```

Return

```
>>> Aa=1
```

Return

```
>>> print(aa-Aa)
```

Return

- ▶ **python**: “Case sensitive”

```
>>> ä=9
```

Return

Onnistuu **python 3.4.0** :ssa

Ei onnistunut **python 2.7.6** :ssa

- ▶ Monet ongelmat vältetään,
kun ei käytetä skandeja,
edes # merkin jälkeen

```
>>> print(aa-Aa)
```

Return

- ▶ **Kaatu!** Eli **python**:ssa myös
tyhjiillä merkkeillä merkitystä

octave

- ▶ **octave** komentotulkissa

```
octave:*>aa=2;
```

Return

```
octave:*>Aa=1;
```

Return

```
disp(aa-Aa);
```

Return

- ▶ **octave**: “Case sensitive”

- ▶ Kokeile mitä tapahtuu
ilman ; merkkiä

```
>>> ä=9;
```

Return

Ei onnistu **octave 3.8.1** :ssa

- ▶ Monet ongelmat vältetään,
kun ei käytetä skandeja

```
>>> octave:*> disp(aa-Aa);
```

Return

- ▶ **Ei kaadu!** Eli **octave**:ssa
tyhjiillä merkkeillä ei merkitystä

Ensimmäiset ohjelmat

Ensimmäiset ohjelmat

- ▶ **python**: 1. ohjelma

```
#!/usr/bin/env python
# Kommenttirivi: Tama on ohjelmani Pmalli1.py
print("Hello_world!")
```
- ▶ **python**: Sama tulos `print("Hello")` ja `print('Hello')`, s.o. `"` tai `'`
- ▶ Kaikissa ohjelmatulostuksissa `_` selventää, että kyseisessä kohdassa on yksi tyhjä merkki
- ▶ **octave**: 1. ohjelma

```
#!/usr/bin/env octave
# Kommenttirivi: Tama on octave ohjelmani Omalli1.m
clear # Poistetaan mahdollinen aiempi input ..
disp('Hello_now_world')
```
- ▶ **octave**: Sama tulos `disp("Hello")` ja `disp('Hello')`, s.o. `"` tai `'`
- ▶ `clear` hävittää **octave**:ssa aiemmat muuttujat yms.
- ▶ **Kertausta**: `clear`  tyhjentää näytön **linux** komentotulkissa
- ▶ `>> import os ; os.system('clear')` tyhjentää näytön **python** komentotulkissa
- ▶ `octave:> clc` tyhjentää näytön **octave** komentotulkissa
- ▶ **python** & **octave**: kommenttirivit alkavat merkillä `#`. Käytä niitä ohjelman selkeyttämiseen   