1.1 Yleistä kurssista

- Työkalukurssi
- Olennainen pohjakurssi useille muille kielitieteen kursseille
- Suorituksesta saa yhden opintoviikon
- Suoritustapa on luennoilla käyminen ja harjoitustehtävien tekeminen. Erillistä tenttiä ei järjestetä.
- Jos osaa valmiiksi käyttää Unixia, kurssi kannattaa suorittaa tekemällä pelkästään harjoitustehtävät

Date: 2004/02/05 20:18:37

1.2 Merkinnöistä

- Merkillä [ylim] on merkitty materiaali, joka ei missään nimessä kuulu oppimäärään vaan on mukana asioiden (epä)selventämiseksi.
- Uudet termit merkitään *näin*
- Uudet komennot esitellään tällaisessa muodossa:
  komento `[ -valitsin ]` *`argumentti`* (Muistisana)
  Hakasulkeissa olevat asiat ovat *valinnaisia* eli ne voivat puuttua komennosta. Kursiivilla kirjoitettujen sanojen tilalle pannaan jotain muuta, esimerkiksi tiedoston nimi.
- Tiedostojen nimet, esimerkkikomennot, yms. kaikki, mikä saatetaan kirjoittaa komennon osana, näytetään `tällaisella kirjasimella`.

Date: 2004/02/05 20:18:37

1.3 Miksi järjestää UNIX-kurssi?

- Yleisen kielitieteen laitoksen ympäristö on Linux-pohjainen. Linux on yksi UNIXin kaltainen käyttöjärjestelmä / -ympäristö.
- Vaikkei Linuxin peruskäyttö edellytäkään tuntemusta perinteisestä Unix-ympäristöstä, tehokäyttöä ja useita kursseja varten tuntemus on välttämätöntä.
- Monet kielitieteelliset menetelmät voidaan tehdä suoraan Unixin perustyökaluilla
- Kun Unixin perusympäristöä on oppinut kerran käyttämään, taitoa pystyy hyödyntämään muissakin järjestelmissä (ainakin nähtävissä olevan tulevaisuuden ajan)

2.1 Mikä on UNIX?

- Toisaalta UNIX-*käyttöjärjestelmä*perhe, toisaalta Unix-*käyttöympäristö*, joka on kehittynyt pikku hiljaa perusasioiden pysyessä samana aina 70-luvulta asti.
- *Moniajojärjestelmä*, mikä tarkoittaa, että käynnissä on useita toisistaan riippumattomia ohjelmia yhtaikaa
- *Monikäyttäjäjärjestelmä*, mikä tarkoittaa, että samaa konetta voi käyttää yhtaikaa useita toisistaan riippumattomia käyttäjiä
- *Suojattu järjestelmä*, mikä tarkoittaa, että järjestelmä on rakennettu niin, etteivät käyttäjät pysty häiritsemään sen toimintaa eivätkä toisiaan.

  ⋆ Tämä kurssi opettaa pikemminkin Unix-käyttöympäristöä, ei niinkään UNIX-käyttöjärjestelmää.

Date: 2004/02/05 20:18:37

2.2 Miksi Unix-käyttöympäristö on kiva?

- Tietokone on käyttäjän työkalu, eikä musta laatikko, joka tekee "jotain". Unix antaa käyttäjälle avaimet ymmärtää, mitä koneessa tapahtuu, sekä käskeä sitä tekemään tarkalleen halutut asiat.
- Unix perustuu *työkaluajatteluun*, jossa jokainen ohjelma tekee oman hommansa eikä mitään muuta. Monimutkaisemmat operaatiot rakennetaan käyttäen yksittäisiä ohjelmia palasina.
- Lähes kaikkeen maan ja taivaan väliltä on olemassa jokin (yleensä vapaasti ja ilmaiseksi saatavilla oleva) ohjelma.
- Unix on voimakas ympäristö, jolla pystyy saamaan aikaan monia asioita, joihin muissa ympäristöissä ei yksinkertaisesti ole keinoja.

Date: 2004/02/05 20:18:37

2.3 Miksi Unix-käyttöympäristö on ärsyttävä?

- UNIXin historia on pitkä ja joskus omituinen. Vaikka Linuxissa tämä ei usein näykään, aina on joitain asioita, jotka ovat niin kuin ovat "historiallisista syistä".
- Aloittelijalle on usein suuri kynnys alkaa tehdä asioita eri tavalla kuin mihin on tottunut esim. Windows-ympäristössä.
- Erilaisten komentojen määrä on pyörryttävä (esim. marsissa 1900 kpl) ja niiden nimet ovat usein lyhyitä ja kryptisiä — siis vaikeita muistaa.
- UNIXeissa on pieniä eroja, ja lähes joka tehtävään on olemassa monta ohjelmaa, joista pitäisi osata valita. Ja toisaalta yhden UNIXin pystyy virittämään melkein millaiseksi vain.

Date: 2004/02/05 20:18:37

3.1 Mikä on Linux?

- Linux on UNIX-käyttöjärjestelmän ytimen ilmainen toteutus.
- Unixin kaltainen käyttöympäristö (perusohjelmat kuten `grep` yms.) on toteutettu GNU-projektin ohjelmilla.
- Linux-ympäristö (Linux-ydin, GNU-ohjelmat, muut ilmaiset ohjelmat) on vapaaehtoisten hajautetusti ylläpitämä kokonaisuus.
- Kaikkien ohjelmien lähdekoodi on vapaasti saatavilla. Tämä tarkoittaa, että ohjelmiin voi tehdä haluamiaan muutoksia ja parannuksia (jos osaa ohjelmoida).

3.2 Linuxin hyviä ja huonoja puolia

+ Valtavat määrät niin “Windowsmaisia” kuin “Unixmaisia” ohjelmia tarjolla
+ Yhteisöjen tarjoama vahva käyttötuki ja paljon dokumentaatiota
+ Viriteltävissä todella monenlaiseksi
+ Hyvä tuki erilaisille laitteille, niin vanhoille kuin uudemmillekin
– Edellyttää edelleenkin silloin tällöin asioista selvän ottamista
– Kehittyy sellaista vauhtia, että mukana on joskus vaikea pysyä
– Kotikäyttöön se pitää yleensä asentaa itse (vs. Windows)

4.1 Sisään- ja uloskirjautuminen

- Yksittäiseen UNIX- tai Linux-koneeseen on monta reittiä kirjautua sisään: ssh-etäyhteys, päätteet (vanhoissa järjestelmissä) ja tietenkin koneen oma näyttö ja näppäimistö.
- Kaikissa kirjautumistavoissa kysytään käyttäjän tunnus ja salasana. Nämä ovat tiedot, joiden perusteella käyttäjä tunnistetaan.
- Useimmat sisäänkirjautumistavat käynnistävät onnistuneen sisäänkirjautumisen jälkeen käyttäjän *oletuskomentotulkin*. CS Linuxeissa kuitenkin oletuksena käynnistyy graafinen käyttöympäristö.
- Istunto loppuu uloskirjautumiseen. Komentotulkista kirjaudutaan pois komennolla `logout` (tai `exit`) ja graafisesta ympäristöstä jostain valikosta löytyvällä logout-toiminnolla.

Date: 2004/02/05 20:18:37

4.2 Turvallisuudesta

- Istunnon aikana toiminnassa olevat ohjelmat toimivan sisään kirjautuneen käyttäjän oikeuksilla.
- Niinpä niillä ei voi vahingoittaa muita käyttäjiä tai järjestelmää: Unix on turvallinen ympäristö kokeilla kaikenlaista.
- Unix tarjoaa kuitenkin vahvat työkalut, joita voi käyttää verkkohäiriköintiin jne. Taitavampi käyttäjä saa tunnuksellasi paljon enemmän vahinkoa aikaan kuin sinä. Siksi *istuntoa ei saa jättää vartioimatta.*
- Poistuessasi koneen äärestä voit *lukittaa* istunnon. KDE:ssa tämä tapahtuu painamalla alapalkin lukonkuvaa.

4.3 Tositoimiin: istunto

1. Kirjaudu sisään antamalla käyttäjätunnuksesi ja salasanasi. Koneessa käynnistynee graafinen käyttöympäristö “KDE”.
2. KDE:n käynnistyttyä kokeile avata joitain ohjelmia alapalkista: esimerkiksi punaisesta dinosauruksen päästä käynnistyy www-selain, “mozilla”, televisioruudun näköisestä painikkeesta komentotulkki ja paperi-sulkakynä-yhdistelmästä kirjoitusohjelma.
3. Sulje ikkunoita niiden yläkulmassa olevasta ruksista.
4. Lukita istunto, ja avaa se taas antamalla salasanasi.
5. Kirjaudu ulos K-valikosta löytyvällä “logout”-toiminnolla.

Date: 2004/02/05 20:18:37

5.1 Mikä on komentotulkki?

- *Komentotulkki* (shell) on ohjelma, joka on suunniteltu helpoksi keinoksi käynnistää muita ohjelmia.
- Unixin komentotulkit osaavat itse asiassa paljon enemmänkin.
- Komentotulkki ja perusohjelmat muodostavat yhdessä Unixin "sielun", joka on kehittynyt pikku hiljaa 70-luvulta lähtien.
- Tämä ympäristö on UNIXien ja erilaisten Linuxien yhteinen punainen lanka (ohjelmointiympäristön lisäksi).
- Komentotulkin toiminta muistuttaa jonkin verran keskustelua: annetaan komento/kysymys, saadaan vastaus, jatketaan samaan tapaan.
- Tällä kurssilla käytetään `bash`-komentotulkkia.

Date: 2004/02/05 20:18:37

5.2 Komentotulkin perustoiminta

- Valmiustilassa komentotulkki tulostaa *kehotteen* (prompt) (esim. `pkalliok@mars $`) ja jää odottamaan käyttäjän *syötettä* (eli komentoa).
- Käyttäjä kirjoittaa komennon ja painaa lopuksi rivinvaihtonäppäintä (enter / return / hassu kulmanuoli).
- Komentotulkki suorittaa komennossa määritetyn *ohjelman*, ja jää odottelemaan, että se hoitaa hommansa loppuun
- Tämän jälkeen komentotulkki tulostaa uuden kehotteen ja odottaa uutta syötettä.

5.3 Komentojen yleinen muoto

⋆ Esimerkkikomento: `ls -l -a /web/ling`

- Komennot muodostuvat yhdestä tai useammasta sanasta (erotettuna välilyönnein), joista ensimmäinen on suoritettavan ohjelman nimi (*mitä* tehdään, yllä `ls`) ja loput sille annettavia *argumentteja* (*miten* tai *mille* tehdään).
- Argumenttien muoto riippuu ohjelmasta, mutta on lähes aina samanlainen:
- Ensinnäkin on *valitsimia* (options), jotka muuntavat ohjelman toimintaa ("miten tehdään"). Valitsimet alkavat yleensä viivalla (-). Yllä `-l` ja `-a`.
- Yksikirjaimisia valitsimia voi yleensä yhdistää, esimerkiksi yllä olevan käskyn voisi kirjoittaa muotoon `ls -la /web/ling`

- Kaikki muut kuin valitsimet ovat “tavallisia argumentteja”, ja kertovat yleensä, mitä ohjelma käsittelee (“mille tehdään”). Yllä `/web/ling`.
- Tekstiä käsitteleville komennoille “tavalliset argumentit” yleensä kertovat, mistä tiedostoista käsiteltävä teksti (=syöte) luetaan, ja jos tavallisia argumentteja ei ole, se luetaan suoraan käyttäjältä.
- Hakemistoja käsitteleville komennoille “tavalliset argumentit” kertovat, mitä hakemistoa käsitellään, jne.

⋆ Niin komennossa, valitsimissa kuin tiedostojen nimissä (tiedostoista tuonnempana) pienet ja isot kirjaimet ovat eri kirjaimia, eli esim. `-l` ja `-L` ovat eri valitsin.

5.4 Tositoimiin: komennot

1. Kirjaudu sisään ja käynnistä komentotulkki.
2. Kokeile komentoa `ls -l -a /web/ling` siten, että jätät siitä pois eri osia.
3. Kokeile kirjoittaa komentotulkkiin jotain roskaa, esim. “Minä olen maailman valtias” ja paina enter. Mitä tapahtuu ja miksi?
4. Komento `echo` on perin yksinkertainen. Anna sille joitain argumentteja ja katso, mitä se tekee. Katso myös, mitä tekee komento `date`.
5. Kokeile, miten valitsin `-n` muuttaa `echo`-komennon toimintaa.
6. Komennossa `grep -i foo /usr/share/dict`: mitkä ovat valitsimia, mitkä tavallisia argumentteja ja mikä on ohjelman nimi?

Date: 2004/02/05 20:18:37

6.1 Mikä on tiedostojärjestelmä?

- Tiedostojärjestelmä on se osa tietokonejärjestelmästä, joka sisältää joitain (talletettuja) tietoja
- Erityisesti tiedostojärjestelmä tarkoittaa sitä, *miten* ja missä muodossa nämä tiedot on talletettu
- Unixin tiedostojärjestelmä on hyvin yksinkertainen: se koostuu *hakemistoista*, ja näissä sijaitsevista toisista hakemistoista (joita kutsutaan *alihakemistoiksi*) ja *tiedostoista*.
- Hakemisto on siis käytännössä lista tiedostoja ja alihakemistoja
- Hakemistot, alihakemistot ja ali-alihakemistot muodostavat *tiedostohierarkian*. Unix-järjestelmässä kaikki tiedostot sijaitsevat yhdessä hierarkiassa, *juurihakemiston* alla.

Date: 2004/02/05 20:18:37

⋆ Terminologiasta on huomattava sen verran, että "tiedosto" on perinteisesti Unixissa tarkoittaanut oikeastaan mitä tahansa tiedostojärjestelmässä olevaa: siis myös hakemistot ovat tämän terminologian mukaan "tiedostoja". Tästä syystä varsinaista tietoa sisältäviä tiedostoja (jotka esitellään myöhemmin) on joskus kutsuttu "tavallisiksi tiedostoiksi" (*regular files*). Tässä opintomonisteessa puhutaan yleensä yksinkertaisesti tiedostoista — asiayhteydestä pitäisi käydä selväksi, sisältääkö käsite hakemistot vai ei.

6.2 Miten tiedostoihin viitataan?

Tiedoston (tai hakemiston) voi nimetä kahdella tapaa:

1. Voi käyttää *absoluuttista* tiedostonimeä:
   - tällöin kerrotaan tiedoston paikka aina juurihakemistosta lähtien.
   - esimerkiksi `/etc/apt/sources.list` tarkoittaa juurihakemiston `etc`-alihakemiston `apt`-alihakemiston tiedostoa nimeltä `sources.list`.
   - hakemistojen nimien väliin siis pannaan aina kauttaviiva (`/`). Koska nimi tavallaan kertoo reitin juurihakemistosta tiedostoon, sitä kutsutaan myös *poluksi* (path).
   - abs. tiedostonimet alkavat aina kauttaviivalla. Siksi juurihakemiston itsensä nimi on `/`.

2. Voi käyttää *suhteellista* tiedostonimeä:
   - Unixissa ohjelmilla (myös komentotulkilla) on aina *oletushakemisto*. Siinä missä abs. nimet ovat polkuja suhteessa juurihakemistoon, suhteelliset nimet ovat polkuja suhteessa oletushakemistoon.
   - esimerkiksi `projektit/aatos.tex` tarkoittaa oletushakemiston `projektit`-alihakemiston `aatos.tex`-nimistä tiedostoa.
   - sisäänkirjauduttaessa oletushakemisto on käyttäjän *kotihakemisto*. Tämä tarkoittaa käyttäjän omaa paikkaa, johon käyttäjä voi tallettaa omat tiedostonsa, asetuksensa jne.
   - kaikissa hakemistoissa on pseudohakemisto `..`, joka viittaa hakemiston ylähakemistoon.
   - esimerkiksi, jos oletushakemisto on `/web/ling/kit`, nimi `../monako/index.html` viittaa samaan tiedostoon kuin abs. polku `/web/ling/monako/index.html`.

7.1 Hakemistoihin liittyviä komentoja

ls `[ -l ] [ -a ] [ hakemisto ]` (LiSt)
Näyttää hakemiston sisällön, eli mitä tiedostoja ja hakemistoja se sisältää. Jos hakemiston nimeä ei anneta, näyttää oletushakemiston sisällön. Valitsin `-l` antaa (paljon) enemmän yksityiskohtia tiedostoista, ja `-a` näyttää myös ns. *piilotiedostot* (näistä lisää tuonnempana).

cd `[ hakemisto ]` (Change Directory)
Vaihtaa oletushakemistoa. Vaihdon jälkeen kaikki suhteelliset tiedostonimet ovat suhteessa uuteen oletushakemistoon. Jos hakemiston nimeä ei anneta, vaihtaa oletushakemistoksi kotihakemiston.

pwd (Print Working Directory)
Kertoo tämänhetkisen oletushakemiston.

mkdir *hakemisto [hakemisto2 ...]* (MaKe DIRectory)
Luo argumenteiksi annetut hakemistot.

rmdir *hakemisto* (ReMove DIRectory)
Poistaa (tyhjän) hakemiston.

mv *hakemisto uusinimipolku* (MoVe)
Uudelleennimeää tai siirtää hakemiston (sisältöineen) uuteen paikkaan / uudelle nimelle. (Tiedoston nimipolku on oikeastaan sen sijainti, joten uudelleennimeäminen ja siirtäminen ovat sama asia.)
Esim. `mv projektit proj` vaihtaa projektit-hakemiston nimeksi proj ja `mv build /tmp/mybuild` siirtää oletushakemiston `build`-alihakemiston `/tmp`-hakemiston alihakemistoksi `mybuild`.

7.2 Tositoimiin: hakemistot

1. Käynnistä komentotulkki ja katso, mikä on oletushakemistona.
2. Katso, mitä tiedostoja se sisältää.
3. Katso, mitä tiedostoja juurihakemisto sisältää.
4. Tee uusi alihakemisto `kokeilu`, katso, mitä tiedostoja se sisältää, tee sille jokin ali-alihakemisto ja katso, mitä tiedostoja `kokeilu` nyt sisältää. Poista hakemistot.
5. Mene (vaihda oletushakemistoksi) `/web/ling`, liikuskele siellä ympäriinsä ja katsele, mitä tiedostoja siellä on.
6. Palaa kotihakemistoosi.
7. Katso `pwd`:lla, mikä on kotihakemistosi absoluuttinen polku.

Date: 2004/02/05 20:18:37

7.3 Nippelitietoa poluista [ylim]

- Jokaisessa hakemistossa on pseudohakemistot `.` ja `..`, joista ensimmäinen viittaa hakemistoon itseensä ja jälkemmäinen siihen hakemistoon, joka sisältää kyseisen hakemiston.
- Hakemiston nimessä voi olla kauttaviiva (/) lopussa. Tämä kauttaviiva jätetään huomiotta. Samoin kaksi peräkkäistä kauttaviivaa tiedostonimessä tulkitaan yhdeksi.

Siis esimerkiksi jos oletushakemisto on `/home/p/k/pkalliok`, seuraavat polut osoittavat samaan hakemistoon:

- `tmp`
- `./tmp/`
- `../pkalliok//tmp/.`
- `/usr/../home/p/./../k/pkalliok//../pkalliok//.//tmp`

Date: 2004/02/05 20:18:37

8.1 Mikä on tiedosto? (puoliteoriassa)

- Tiedosto on paikka, jossa on (tallessa) tietoa.
- Unixin käsitys (tavallisesta) tiedostosta on hyvin yksinkertainen: tiedoston sisältö on läjä merkkejä (esim. luonnollisen kielen tekstiä), tai oikeastaan merkkijono (tarkoittaa, että merkit ovat tietyssä järjestyksessä).
- Unix sinänsä ei aseta tämän kummempia rajoituksia tiedoston muodolle.
- Usein teksti on järjestelty riveiksi. Rivinvaihtokin on yksi merkki muiden joukossa.
- Joskus rivit on järjestelty kentiksi (esim. jos tekstissä on taulukko). Kentänvaihto (“tab”) on myös yksi merkki.

Date: 2004/02/05 20:18:37

- Tekstiä käsittelevät Unix-ohjelmat käsittelevät tiedostoa sellaisenaan: esimerkiksi se, mitä tekstinmuokkausohjelma `emacs` näyttää, vastaa merkki merkiltä sitä, mitä tiedostossa on.
- Vastakohtana monet Windows-ohjelmat, joilla kaikilla on omat *tiedostomuotonsa*, jolloin ruudulla näkyvän tekstin ja tiedostoon talletetun merkkijonon välillä ei välttämättä ole mitään yhteyttä.
- Joillain asioilla, kuten kuvilla ja äänellä, ei voi olla yksi yhteen -vastaavuutta merkkijonojen kanssa (toisin kuin tekstillä). Niinpä näitä käsittelevät ohjelmat tallentavat kuvan tiedostoon jossain tietyssä tiedostomuodossa.
- Unix-ohjelmat käyttävät yleensä standardeilla määriteltyjä tiedostomuotoja (kuten JPEG, MP3 jne.)

Date: 2004/02/05 20:18:37

⋆ Tiedostot, joiden nimi alkaa pisteellä (`.`), ovat ns. piilotiedostoja / piilohakemistoja. `ls`-komento ei näytä niitä ilman `-a`-valitsinta, eikä komentotulkin nimilavennus (esitellään tuonnempana) kiinnitä oletuksena huomiota niihin. Piilotiedostot sisältävät usein ohjelmien asetuksia tai muuta sellaista, jonka ei oleteta kiinnostavan käyttäjää kovin paljon, ja yleensä kotihakemisto on pullollaan erilaisten ohjelmien piilotiedostoja. Myös erikoishakemistot `.` ja `..` ovat piilotiedostoja.

Date: 2004/02/05 20:18:37

8.2 Tiedostoihin liittyviä komentoja

cat [*tiedosto tiedosto ...*] (CATenate)
Yhdistää argumentteina annetut tiedostot ja tulostaa ne näytölle.
`cat tiedosto` on idiomaattinen tapa tulostaa yksi tiedosto näytölle.

less *tiedosto* (Ei muistisanaa)
Jos tiedosto on pitkä, `cat` on huono keino sen tarkastelemiseksi. `less` näyttää tiedoston yksi näytöllinen kerrallaan. `less`-komennon lyhyt käyttöohje on seuraavassa diassa.

file *tiedosto* (what kind of FILE)
Kertoo (so. arvaa) tiedoston sisällön perusteella, minkä tyyppistä tietoa tiedosto sisältää: onko se tekstiä, ohjelma, kuva, ...

grep *sana* [*tiedosto tiedosto ...*] (Global RegExp Print)
Näyttää tiedosto(i)sta kaikki rivit, joilla esiintyy annettu sana. Hyvä tiedon etsintään.

cp `[ -r ] [ -i ]` *`lähde [ lähde ... ] kohde`* (CoPy)

mv `[ -i ]` *`lähde [ lähde ... ] kohde`* (MoVe)

rm `[ -r ] [ -i ]` *`tiedosto`* (ReMove)

Kopioi (`cp`) tai siirtää (`mv`) tiedoston tai tiedostoja uuteen paikkaan. `cp` jättää lähdetiedoston jäljelle, kun taas `mv` poistaa sen. `rm` vain poistaa tiedoston (käytä varoen!).

`cp`:lla ja `mv`:lla on kaksi toimintatapaa. Jos *kohde* on olemassa oleva hakemisto, tiedostot tulevat sen sisään samalla nimellä kuin olivat aiemmassakin paikassaan. Muussa tapauksessa lähdetiedostoja saa olla vain yksi ja *kohde* on sen uusi sijainti/nimi.

Valitsin `-r` aiheuttaa, että `cp` ja `rm` voivat käsitellä myös hakemistoja. `cp` tekee tällöin kopion hakemistosta ja kaikesta sen alla olevasta (*rekursiivinen* kopio), ja `rm` poistaa koko hakemiston ja kaiken sen alla olevan. `mv`-komennolle voi aina antaa lähteiksi myös hakemistoja.

8.3 `less`-komennon lyhyt käyttöohje

`less` on hyvin monipuolinen ohjelma, joka on tarkoitettu yksinomaan (teksti)tiedostojen tarkasteluun. Koska monet muut Unix-ohjelmat käyttävät `less`:a näyttääkseen tiedostoja, sen jonkinlainen hallitseminen on tärkeää. `less`:n komennot ovat hyvin lyhyitä, enintään muutaman näppäimenpainalluksen mittaisia.

- Välilyönti vie tekstissä eteenpäin, `b` (Back) taaksepäin.
- Komento `q` (Quit) vie ulos `less`:sta.
- Komento `g` (Go) vie tiedoston alkuun, `G` loppuun.
- Tiedostosta voi etsiä jotain merkkijonoa komennolla `/` (kauttaviiva). Tämän jälkeen kirjoitetaan etsittävä merkkijono ja painetaan enter. Samaa voi etsiä uudelleen komennolla `n` (Next).

Date: 2004/02/05 20:18:37

8.4 Tulosteen ohjaus tiedostoon

- Tähän mennessä käsittelemämme komennot, silloin kun ne tulostavat jotain, tulostavat sen näytölle.
- Tulosteen voi ohjata myös tiedostoon. Tämä tarkoittaa, että teksti, joka olisi tullut näytölle, tuleekin tiedoston sisällöksi.
- Tulosteen ohjaus on komentotulkin ominaisuus, ja sen takia se tepsii kaikkiin ohjelmiin.
- Lisäämällä komennon loppuun `> tiedosto` voidaan ylikirjoittaa (tai luoda uusi) tiedosto. Esimerkiksi `echo Hepskukkuu >tervehdys` luo tiedoston `tervehdys`, jossa on sana "Hepskukkuu" ja rivinvaihto.
- Lisäämällä komennon loppuun `>> tiedosto` voidaan liittää komennon tuloste tiedoston perään tuhoamatta vanhaa sisältöä.

8.5 Tositoimiin: tiedostot

1. Kirjaudu sisään, käynnistä komentotulkki ja tee kokeiluille uusi alihakemisto `testausta`. Vaihda se oletushakemistoksi.
2. Luo tiedosto `mun_jutut`, johon panet `echo`:lla jonkin elämänviisautesi. Lisää elämänviisauksia tiedoston perään `echo`:lla.
3. Luo tiedosto `tämä_hetki` komennolla `date >tämä_hetki`.
4. Tarkastele näiden tiedostojen sisältöä `cat`:lla ja `less`:lla.
5. Tee `mun_jutut`-tiedostosta kopio `cp`:lla. Lisää siihen vielä joitain viisauksia.
6. Tee alihakemisto `kopiot`, johon teet kopiot kaikista tekemistäsi tiedostoista.

Date: 2004/02/05 20:18:37

7. Katso `file`- ja `less`-komennoilla, mitä sisältävät/ovat esim. nämä: `/usr/include/stdio.h`, `/bin/cat`, `/etc/passwd` ja `/tmp`.
8. Etsi `/etc/passwd`-tiedostosta tietoja itsestäsi (ja kavereistasi) `grep`-komennolla ja `less`-komennolla. Kumpi tuntuu kätevämmältä?
9. Tee `echo`:lla itsellesi puhelinmuistio: kasa nimi-puhelinnumeropareja, yksi kullakin rivillä. (Onko tällainen vanhanaikaista nykyään?) Etsi sieltä tietoja `grep`:lla.
10. Siirrä puhelinmuistio (sekä ehkä elämänviisaustiedosto) turvaan kotihakemistoosi, tuhoa muut kokeilutiedostot `rm -r`:lla.

9.1 Komentotulkin apuja

Komentotulkissa on monta asiaa, jotka hidastavat tai vaikeuttavat sen käyttöä. Yksi on komentojen ja varsinkin valitsimien muistaminen. Toinen on se, että pitkien komentojen ja pitkien tiedostonimien kirjoittaminen on hidasta ja virheille altista.

Tässä osiossa paneudutaan komentotulkissa oleviin erilaisiin apuominaisuuksiin. Niiden tehokas käyttö tekee komentotulkista lähes tilanteessa kuin tilanteessa nopeamman ja vaivattomamman käyttää kuin vastaavat hiiren klikkailuun perustuvat graafiset ohjelmat.

Date: 2004/02/05 20:18:37

9.2 Apua: `man`-sivut

UNIXin perinteinen *online*-apu.

man *aihe* (MANual)
tietoa tietystä ohjelmasta (tms.), esimerkiksi tietoa `date`-ohjelmasta saa komennolla `man date`. Käyttää `less`-ohjelmaa tiedostojen näyttämiseen (katso "`less`-komennon lyhyt käyttöohje" aiempana).

apropos *aihe* (APROPOS)
etsii `man`-sivuja, jotka liittyvät tiettyyn aiheeseen, etsimällä aihesanaa sivujen tiivistelmistä. Esimerkiksi `apropos editor`.

⋆ Tätä jaksoa ennen tulleet komennot on syytä muistaa ulkoa, mutta valitsimet ja tämän jakson jälkeen tulevat komennot voi yhtä hyvin tarkistaa `man`-sivuilta aina tarvittaessa.

9.3 Komento- ja tiedostotäydennys

- `tab`-näppäin (sarkainnäppäin, näppäin `q`:n vasemmalla puolella) on maaginen komentotulkissa. Sen painaminen täydentää sillä hetkellä kirjoitettavan komennon / tiedostonimen niin pitkälle kuin voi.
- Molempien täydennys toimii periaatteessa samalla tavalla: komentotulkki etsii komennon(/t) / tiedoston(/t), joka alkaa jo kirjoitetuilla kirjaimilla, ja lisää loput kirjaimet puolestasi.
- Jos annetulla alulla löytyy useampia vaihtoehtoja, komentotulkki täydentää vain ensimmäiseen eroavaan kirjaimeen asti.
- Toinen `tab`-näppäimen painallus heti perään näyttää kaikki vaihtoehdot, jotka komentotulkki näkee.

Date: 2004/02/05 20:18:37

- Täydennystä oppii käyttämään paremmin kokeilemalla kuin lukemalla.
- Käytännössä tee näin: kirjoita haluamastasi komennosta / tiedostosta pari ensimmäistä merkkiä, paina `tab`.
- Jos täydentyi loppuun asti, sano "jee" ja jatka kirjoittamista.
- Jos ei täydentynyt, anna seuraava kirjain ja paina uudestaan `tab`. Toista kunnes saat koko sanan kirjoitetuksi.
- Jos et muista / tiedä seuraavaa kirjainta, paina toisen kerran `tab` heti perään ja katso vaihtoehtoja.
- Lyhyihin sanoihin ei ole yleensä järkeä käyttää täydennystä.

9.4 Komentohistoria

- Komentotulkki tallettaa kaikki annetut käskyt (nekin, joista on tuloksena vain virheitä). Näitä talletettuja komentoja sanotaan *komentohistoriaksi*.
- Yksinkertaisin tapa liikkua komentohistoriassa on käyttää ylös- ja alas-nuolia: ylös-nuoli antaa aiempia komentoja, alas-nuoli myöhempiä.
- Näppäinkomento `C-r` (eli pidä `ctrl`-näppäin pohjassa, paina `r`) aloittaa tekstihaun historiasta. Kirjoita jokin osa vanhasta komennosta ja paina enter.

history (HISTORY)
Tulostaa komentohistorian. Komentohistoriasta voi suorittaa jonkin komennon uudestaan kirjoittamalla **!*n***, jossa *n* on komennon historianumero.

9.5 Komentorivimuokkaus

- Olet varmaan huomannut, että komentorivillä voi kulkea edestakaisin nuolinäppäimillä, poistaa kirjaimia `backspace`-näppäimellä (enterin yläpuolella) ja lisätä kirjaimia haluamaansa kohtaan.
- Itse asiassa komentotulkki tarjoaa hyvin monipuolisen valikoiman näppäinkomentoja komentorivin muokkaukseen. Komennot muistuttavat `emacs`-tekstieditorin (esitellään tuonnempana) komentoja.
- `backspace`:n jälkeen hyödyllisin komento on edeltävän sanan poisto, joka tehdään `C-w`:lla (muistathan: ctrl pohjassa ja ...)
- Muita hyödyllisiä ovat `C-a` ja `C-e`, jotka hyppäävät rivin päihin.
- `C-d` poistaa kohdistinta seuraavan kirjaimen.

Date: 2004/02/05 20:18:37

9.6 Tositoimiin: komentotulkin tehokas käyttö

1. Kirjaudu sisään ja käynnistä uusi komentotulkki. Anna `history`-komento ja tarkastele tulosta.
2. Katso, mitä `man history` sanoopi. Entä `apropos shell`?
3. Hae vanhoja käskyjäsi eri tavoin (nuolinäppäimet, `C-r`) ja muuttele niitä. Esim. muuta jostain `cat` `less`:ksi tai päin vastoin.
4. Tarkastele jotain kotihakemistosi alihakemistossa olevaa tiedostoa `less`:lla. Käytä tiedoston nimeämiseen `tab`-täydennystä. Tarkkaile, kuinka paljon täydentyy aina yhdellä painalluksella.
5. Kokeile, mitä tapahtuu, jos painaa `tab`-näppäintä kahdesti, ennen kuin on kirjoittanut komennosta mitään. Entä, kun itse komento (esim. `cat`) ja välilyönti on kirjoitettu, muttei vielä mitään muuta?

Date: 2004/02/05 20:18:37

9.7 Tiedostolavennus

- Usein halutaan antaa komennolle argumenteiksi iso määrä (mahdollisesti tietynlaisia) tiedostoja. Tähän tarkoitukseen komentotulkissa on *tiedostolavennus*.
- Lavennus toimii siten, että kun käyttäjä on antanut komentonsa ja painaa enter, komentotulkki tutkii kaikki komennon sanat, joissa on tiettyjä maagisia merkkejä, ja korvaa nämä sanat olemassa olevien tiedostojen nimillä.
- Tämäkin on helpompi selittää esimerkeillä. Olkoon meillä seuraavansisältöinen hakemisto:

```
[atehwa@maim ~/proj/sed/Stx/common]$ ls
begin.m4    common.m4     end.m4      quote.sed
block.sed   CVS           inline.sed  quote_us.sed
cleanup.m4  emphasis.sed  markup.m4
```

- Tärkein maaginen merkki (*jokerimerkki*) on asteriski (*). Sanat, joissa on tähti, sovittuvat kaikkiin tiedostoihin, joissa tähden kohdalla on mikä tahansa merkkijono:

```
[atehwa@maim ~/proj/sed/Stx/common]$ echo *.m4
begin.m4 cleanup.m4 common.m4 end.m4 markup.m4
[atehwa@maim ~/proj/sed/Stx/common]$ echo b*
begin.m4 block.sed
[atehwa@maim ~/proj/sed/Stx/common]$ echo quote*ed
quote.sed quote_us.sed
```

- Tähti toimii myös hakemistojen kanssa:

```
[atehwa@maim ~/proj/sed/Stx]$ echo */make.m4
html/make.m4 latex/make.m4 man/make.m4
```

- `echo`-komento on köyhä muuhun kuin testaamaan, mitä se komentotulkki sinne oikein laventaa (kuten yllä). Mutta esimerkiksi komennolla `cat *.txt` voin tulostaa yhtenä pötkönä kaikki `.txt`-loppuiset tiedostoni.
- Pelkkä tähti laventuu kaikiksi hakemistossa oleviksi tiedostoiksi.
- Tähti *ei* sovitu tiedostonimen alussa olevaan pisteeseen (`.`), koska se on piilotiedoston merkki eivätkä ihmiset yleensä halua piilotiedostojen tulevan mukaan lavennukseen.
- Piilotiedostot saa lavennetuksi kirjoittamalla `.*` — tämän lavennuksen kanssa on kuitenkin syytä olla varovainen, sillä siihen sisältyvät mm. pseudohakemistot `.` ja `..`.

  ⋆ Erityisesti ei koskaan, koskaan pidä kirjoittaa komentoa `rm -r .*` (Miksi? Tämä on harjoitustehtävä.)

- Jokerimerkki `?` sovittuu tasan yhteen mihin tahansa merkkiin. Tätä ei tarvitse kovin usein; lähinnä tietynmittaisten tiedostonimien listaamiseen:

  ```
  [atehwa@maim ~/proj/sed/Stx/common]$ echo ?????????
  block.sed common.m4 markup.m4 quote.sed
  ```

- Seuraavaksi tärkein jokerirakenne on vaihtoehtoisuus, jota merkitään aaltosulkeiden (`{` ja `}`) sisässä olevalla listalla. Se laventuu vuorollaan kuksikin listassa olevaksi kohdaksi:

  ```
  [atehwa@maim ~/proj/sed/Stx/common]$ echo ?????.{m4,sed}
  begin.m4 block.sed quote.sed
  ```

- Aaltosulkeet ovat siitä erikoinen rakenne, etteivät ne piittaa, onko tiedostoa oikeasti olemassa:

  ```
  [atehwa@maim ~]$ echo {cat,dog}{,s,like}
  cat cats catlike dog dogs doglike
  ```

9.8 Tositoimiin: tiedostolavennus

1. Yhdistä kaikki jonkin hakemistosi tiedostot yhdeksi suureksi tiedostoksi (`cat` ja `>`-ohjaus).
2. Muuta kaikki tekstiä sisältävät tiedostosi `.txt`-päätteisiksi ja siirrä ne sitten yhdellä komennolla johonkin toiseen hakemistoon. (Molempiin `mv` on oikea komento.)
3. Komennolla `wc` (katso `man wc`) voi laskea tiedostosta rivit, sanat ja kirjaimet. Tulosta tiedostoon, mitä viisikirjaimisia tiedostoja `/etc`-hakemistossa on, ja katso `wc`:lla, kuinka monta niitä on.
4. Laske `wc`:lla 2-kohdan tiedostoista sanat (`*`).
5. Muodosta aaltosulkeilla jokin vähintään 50-sanainen sanajoukko. (Liitä vaikkapa kaikki omistusliitteet erilaisiin yhdyssanoihin.)

9.9 Vielä pari hengissäselviytymisasiaa

- Parhaillaan toiminnassa olevan käskyn saa keskeytetyksi näppäinkomennolla `C-c` ja pysäytetyksi `C-z`:lla.
- Jos kaikki näyttää pysähtyneen eikä ruudulle tule mitään, kannattaa painaa `C-q`. Kyseessä saattaa nimittäin olla komentotulkin typerä tulosteenkeskeytysominaisuus.
- Jos ruutusi on jotenkin omituisessa tilassa (esim. pelkkää merkkimössöä tai kirjoittamasi komennot eivät näy), kannattaa antaa komento `reset`. Tämä palauttaa normaaliasetukset.
- Ruudun saa tyhjennetyksi komennolla `C-l`.
- Jos jokin komento odottaa sinulta syötettä (esim. `cat`, jos sille ei anna yhtään tiedostoa argumentiksi), voit kertoa ohjelmalle syötteen loppumisesta `C-d`:lla. Ei tarvitse ymmärtää (vielä).