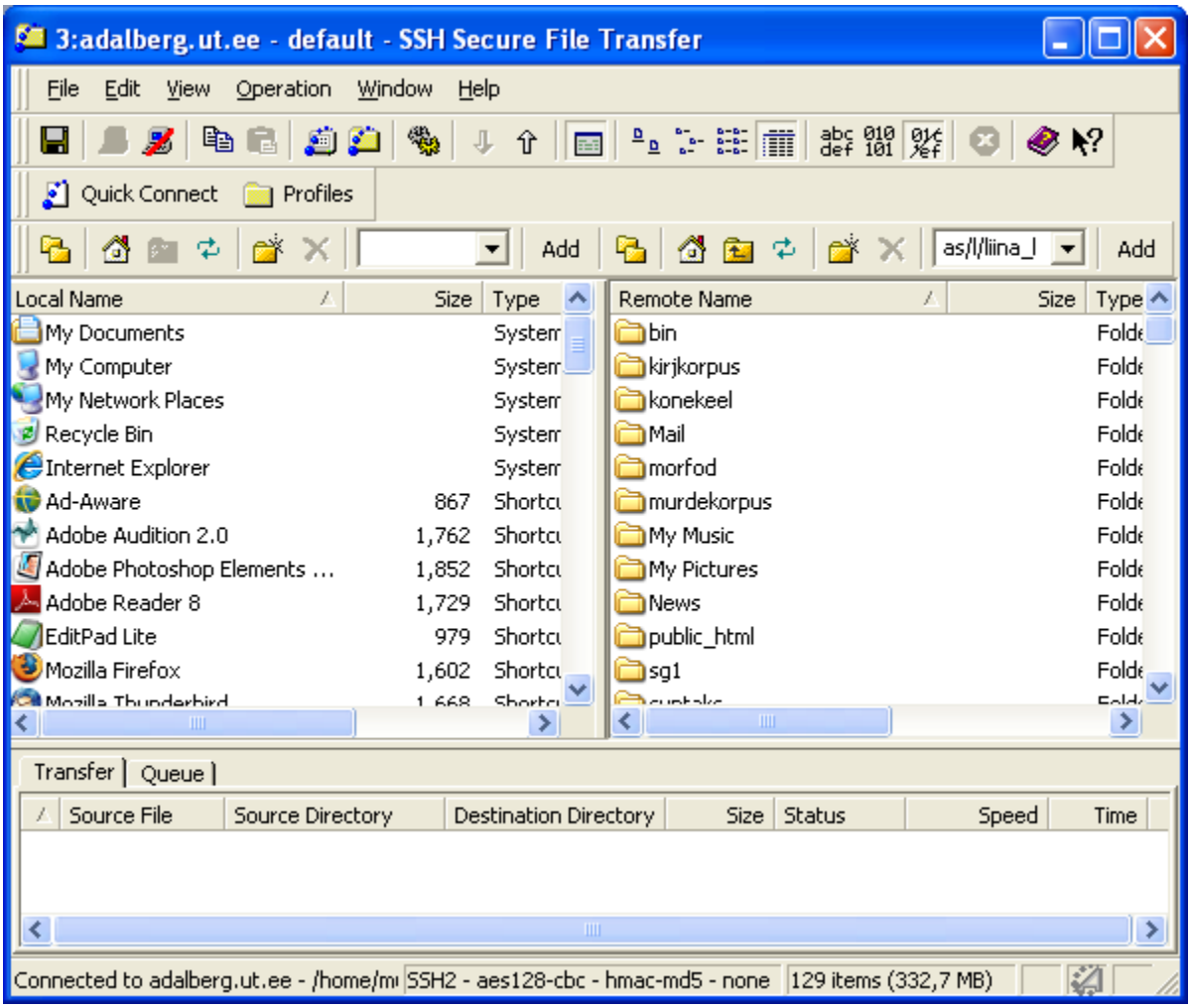
Oma materjali käsitlemine Unixis

Eelnevad näited puudutasid murdekorpuse tekste, mis paiknesid nn õppekorpuses liina_l kodukataloogis. Unixis saab töödelda aga ka muid tekste, näiteks kirjakeele korpuse tekste. Järgnevas osas on kõigepealt juhised, kuidas oma materjali viia oma kodukataloogi ülikooli arvutivõrgus. Oma materjaliks võtame õppimise jaoks kirjakeele korpuse tekstid.

Teema põhiosa käsitleb sagedussõnastikke. Unixis on kerge vaevaga võimalik tekitada kõikvõimalikke sagedussõnastikke, mis sisaldavad keeleuurijale vajalikku informatsiooni. Sagedussõnastike tegemine on iseenesest väga hõlpus, selleks on vaja ainult mõnda käsku kombineerida.

Failide viimine Unixisse

Failide tõstmiseks Unix-keskkonda kasutage taas programmi SSH Secure Shell Client, ent valige rippmenüüst New File Transfer. Seejäre vajutage nupule Qick Connect ning sisestage dialoogiaknasse taas masina nimi, kuhu tahete end sisse logida (adalberg.ut.ee) ja oma kasutajanimi, seejärel vajutage Connect. Seejärel sisestage uude dialoogiaknasse oma parool. Kui olete end edukalt sisse loginud, saate kahest poolest koosneva ekraanipildi. Vasakul paikneb selle arvuti sisu, milles parasjagu olete (teie lauaarvuti, läpakas vms), paremal pool aga teie kodukataloogi sisu ülikooli arvutivõrgus.



Vajalike failide tõstmiseks oma masinast adalbergi või vastupidi valige sobiv kataloog oma nasinast (vasakul pool) ja sobiv kataloog adalbergis (parem pool), vajadusel tekitage adalbergi uus kataloog. Failide tõstmiseks lohistage nad hiirega ühelt poolt teisele.

Kirjakeele korpuse materjalide tõstmine adalbergi

Et kasutada kirjakeele korpuse materjale Unixis, on üks võimalus need kõigepealt alla laadida oma arvutisse, seejärel lahti pakkida (nad on pakitud zip-failiks) ja alles seejärel tõsta tekstid Unixisse. Kirjakeele korpuse materjalid leiate korpuse sisututvustuse alt, näiteks 1930ndate tekstid paiknevad siin:
http://www.cl.ut.ee/korpused/baaskorpus/1930/

Laadime siit ilukirjandustekstid oma arvutisse ja pakime lahti, selleks:
klõpsake hiirega zip-failil (Ilukirjandustekstid), seejärel küsitakse, kas tahate faili avada või salvestada. Valige salvestamine, otsige või tehke sobiv kataloog (nt ilu1930).
Pakkige failid lahti: Windowsis klõpsake parempoolse hiirenupuga failinimel ning valige rippmenüüst Extract all.

Saate ühe suure faili, millel ei ole failinimelaiendit, seepärast ei oska windows ise valida, mis programmiga seda avada. Kui tahate faili Windowsis vaadata, vaadake seda näiteks Wordpadiga. Tõstke fail Unixisse (juhend eespool).

Nüüd on fail valmis kasutamiseks. Vahetame failivahetusakna taas terminaaliakna vastu (menüüst Window valige New terminal), olete samas masinas.

Kõigepealt võiksite faili lihtsalt sirvida, et meelde tuletada, kuidas see on organiseeritud:
cat failinimi | more

Näete, et iga rea alguses on kood, sellele järgneb neli tühikut. Täpitähed on html-kujul.

Faili saab kasutada ka samasuguste otsingute tegemiseks, nagu tegime veebipõhise kirjakeele korpuse otsimootriga. Oluline on aga kogu aeg meeles pidada sisendit ja väljundit. Järgmises näites on lihtne näide selle kohta, kuidas tekstist otsida grep-käsuga.

Harjuta: täiendlausete leidmine

Täiend- ehk relatiivlaused iseloomustavad mingit kindlat nimisõna:
Mees, kes jooksis üle tänava, oli minu kauge sugulane.

Relatiivlaused algavad eesti keeles tüüpiliselt küsiv-siduvate sõnadega *kes/mis*, mis võivad olla käänatud (üldjuhul ainult ainsuses, *mehed, kelledega me läksime....* on väga ebatavaline.) Piiramegi otsingu praegu nendele sõnadele ja vormidele. (Teoreetiliselt on ka muid võimalusi, nt *kus*.)
Seega lähtume otsingul järgmisest vormistikust:

kes	mis
kelle	mille
keda	mida
kellesse	millesse jne

Seega otsime märgijadasid ke[sdl] ja [mi[sdl], nende ühisosa oleks [km][ei][sdl]. Arvestame veel, et relatiivlause järgneb kirjakeele reeglite järgi komale. Seega:

1) avame faili: cat 30_ilu_ttxt
2) otsime järjestust , ke[sdl] või , mi[sdl]: grep ', [km][ei][sdl]'
3) sirvime faili more-iga või suuname uude faili

Kogu käsk: cat 30_ilu_ttxt | grep ', [km][ei][sdl]' | more

Sagedussõnastikud

Kursuse viimane teema puudutab lihtsate sagedussõnastike tegemist Unixis/Linuxis. Sagedussõnastikke kasutatakse keeleteaduses küllalt palju ja need sisaldavad ohtralt lingvistile vajalikku informatsiooni. Sagedussõnastikke on ka välja antud, näiteks kirjakeele korpuse põhjal on Heiki-Jaan Kaalepi ja Kadri Muischneki koostatud sagedussõnastik, selle veebiversiooni näete siit:
http://www.cl.ut.ee/ressursid/sagedused/
Lihtsat sagedussõnastikku on kasutatud ka mitmetes muudes uurimustes. Kui nipp on käes, võib sagedussõnastiku teha igast tekstist, mis ette juhtub!
Sagedussõnastiku materjali ettevalmistamine

Et tekstidest hakata sagedussõnastikke tegema, oleks kõigepealt vaja vabaneda koodist rea algul – vastasel juhul arvestame sagedussõnastikes ka lausekoode kui sõnu. Koodist on kõige hõlpsam vabaneda cut-käsu abil.

Sagedussõnastiku tegemiseks oleks vaja veel asendada kõik suurtähed väikestega (sest sõnavormid *Kõik* ja *kõik* on ju sama sõna vormid, neid on vaja koos arvestada); samuti on tülikad täpitähed. Et vabaneda html-kujul täpitähtedest ning asendada need tavalistega, võiks (vähemalt sagedasemad) sed-käsuga asendada.

cut
lõikab reast välja etteantud välja. Väli tuleb ise defineerida lipukesega -d
cut -d " " jutumärkide vahele tuleb sümbol, mis välju piiritleb, antud juhul tühik
-f1 number märgib välja numbrit, anutd juhul 1. väli (seega enne tühikut, kui tühik piiritleb välju; sisuliselt on otsitakse selles näites välja rea esimene sõna)
cut –d" " –f1-3 lõikab välja esimesed kolm välja, väljad on defineeritud tühikutega (=sõnad)
cut –c1-2 lõikab reast välja ja suunab standardväljundisse esimesed kaks sümbolit

Näide: faili ettevalmistamine sagedussõnastiku tegemiseks
Koodist vabanemiseks avame faili, seejärel defineerime tühiku väljade eristajana cut-käsus ning lõikame välja 5. väljast alates kõik:

cat 30_ilu_ttxt | cut -d" " -f5- | more

Suurtähtede asendamine väiketähtedega: tr '[A-Z]' '[a-z]'
cat 30_ilu_ttxt | cut -d" " -f5- | tr '[A-Z]' '[a-z]' | more

Täpitähtedest vabanemine (NB! kuna suured tähed on juba asendatud väiketähtedega, pole ka suurte Ä-dega jne vaja arvestada):
cat 30_ilu_ttxt | cut -d" " -f5- | tr '[A-Z]' '[a-z]' | sed 's/ä/ä/g' | sed 's/ö/ö/g' | sed 's/ü/ü/g' | sed 's/õ/õ/g' | more

Täpitähtede sisestamisel võib tekkida probleeme.
Kuna seda rida on tülikas pidevalt korrata, on mõistlik tekitada endale fail, kus need eeltööd on tehtud, ning hiljem teha otsinguid /töödelda edasi seda faili:
cat 30_ilu_ttxt | cut -d" " -f5- | tr '[A-Z]' '[a-z]' | sed 's/ä/ä/g' | sed 's/ö/ö/g' | sed 's/ü/ü/g' | sed 's/õ/õ/g' >1930ilu

Edaspidi võtta sisendiks see fail, näiteks cat 1930ilu | more

Sagedussõnastiku koostamine

Sagedussõnastike tegemiseks on vaja, et teksti oleks eeltöödeldud, st eemaldatud oleks kood rea algusest, suurtähed oleks asendatud väiketähtedega, samuti oleks mõistlik kustutada kirjavahemärke. Kuna eelmises näites koodi kustutamist ja suurtähtede asendamist näitasime, ei hakka seda siin kordama. Kui olete faili juba eeltöödelnud ja selle ka salvestanud, kasutage sisendina seda eeltöödeldud faili.

Sagedussõnastiku jaoks oleks hea kustutada ka kirjavahemärgid. Selle vajalikkus sõltub peamiselt konkreetsest korpusest: kui kirjavahemärgid on juba muust tekstist tühikuga eraldatud, ei hakka need sagedussõnastiku tulemusi mõjutama, kirjavahemärk loetakse siis nagu omaette sõna ja reeglina paiknevad sagedussõnastiku sagedasemas otsas. Sealt võib nad lihtsalt pärast välja visata.

Kui kirjavahemärgid ei ole muust tekstist tühikutega eraldatud, on kasulik nad eelnevalt kustutada; vastasel juhul loetakse näiteks sõnavormid *mees, mees. mees! mees? mees* kõik erinevateks sõnadeks sagedussõnastikus.

Kirjavahemärkide kustutamiseks kasutame käsku tr, lipukese–d taha ülakomade vahele lisame kõik märgid, mis oleks vaja kustutada (kontrolli kindlasti, milliseid kirjavahemärke selles tekstis on kasutatud):
tr –d '.,!?<>"-'

Seejärel oleks vaja iga sõna asetada eraldi reale. Selleks asendasime tühikud reavahetusega:
tr ' ' '\012'

Seejärel 1) sorteerime read; 2) kustutame korduvad read (nii, et jääb alles arv, mitu korda rida=sõna esines); 3) võime sorteerida saadu tagurpidises järjekorras (kuna sagedusinfo paigutatakse rea ette, reastatakse esinemissagedus normaaljuhul vähimast kordade arvust suurimani; kui aga reastame tagurpidi järjekorras, siis suurimast vähimani).

sort
sorteerib read tähestiku järjekorras
sort -f ei tee vahet väike- ja suurtähtedel
sort -r sorteerib vastupidises järjekorras

uniq
kustutab korduvad read
uniq -c lisab rea ette arvu, mis näitab, mitu korda seda rida esines

Näide: sagedussõnastiku tegemine

1) Puhastame teksti
cat 30_ilu_ttxt | cut -d" " -f5- | tr '[A-Z]' '[a-z]' | tr -d ',.?!<>:'
Võime kasutada ka juba eelpuhastatud faili sisendina.
Kui täpitähed html-kujul häirivad, võite ka need asendada (vt eespoolt).

2) asetame iga sõna eraldi reale: tr ' ' '\012'
kogu käsk:
cat 30_ilu_ttxt | cut -d" " -f5- | tr '[A-Z]' '[a-z]' | tr -d ',.?!<>:' | tr ' ' '\012'

3) sorteerime, kustutame korduvad read: sort | uniq –c
kogu käsk:
cat 30_ilu_ttxt | cut -d" " -f5- | tr '[A-Z]' '[a-z]' | tr -d ',.?!<>:' | tr ' ' '\012' | sort | uniq –c

4) sorteerime veelkord tagurpidi järjestuses: sort -r

kogu käsk:
cat 30_ilu_ttxt | cut -d" " -f5- | tr '[A-Z]' '[a-z]' | tr -d ',.?!<>:' | tr ' ' '\012' | sort | uniq –c | sort -r

Näide: millega algab küsilause kõige sagedamini?

Kui tahame teada, millega algab küsilause kõige sagedamini, tuleb kõigepealt mõelda välja, kuidas leida küsilaused, seejärel lõigata välja nende lausete esimesed sõnad ning teha neist sagedussõnastik.

1) Valime faili: cat 30_ilu_ttxt
2) otsime välja küsilaused: grep '?'
3) lõikame välja lause esimese sõna. Selleks defineerime tühiku kui väljade eristaja, ja nagu varem rehkendasime, on sel juhul lause esimene sõna 5. väli. Kuna lause lõpuni pole vaja, siis näeb see käsk välja nii: cut –d' ' –f5
   Kuna lõikasime välja vaid ühe sõna lausest, pole vaja enam reavahetusi tühikutega asendada.
4) Seekord pole vaja tingimata ka suuri tähti väikestega asendada, sest lause algul peaksid kõik sõnad algama suure tähega. Seega võime jätkata sorteerimise ja sageduse järgi reastamisega: sort | uniq –c | sort -r

Kogu käsk: cat 30_ilu_ttxt | grep '?' | cut –d' ' –f5 | sort | uniq –c | sort –r

Näeme, et 1930ndate tekstist otsides häirib otsingut « - see peaks olema jutumärk html-is. Enne sagedussõnastiku tegemist võiks selle osa sed-käsuga kustutada, st asendada mittemillegagi. Samuti võiks eelnevalt kustutada jutumärgid.

Kogu käsk: cat 30_ilu_ttxt | grep '?' | sed 's/«//g' | tr –d "" | cut –d' ' –f5 | sort | uniq –c | sort –r
Tulemusi sirvides näeme, et palju on hulgas sõnu, mis pole küsisõnad. Need on tingitud lausetest, kus näiteks teine või kolmas osalause on küsilause.

Näide: millise kaashäälikuühendiga algavad sõnad kõige sagedamini?

Sagedussõnastikke ei pea tegema ainult sõnadest, vaid võib teha ka sõnaosadest vms.
See näide puudutab sõnaalgulisi kaashäälikuühendeid. Nagu teame, on kaashäälikuühend sõna alguseses eesti keeles seotud üldjuhul laensõnadega, mitte omasõnadega, ning nende hulk on piiratud. Missugused on aga kõige tavalisemad?

Selleks, et kaashäälikuühenditest sagedusnimestikku teha, tuleb arvestada, et 1) igasuguste asjade väljaotsimine grep-käsuga toimub reakaupa. Selleks on vaja kõik sõnad tõsta eraldi reale, samuti suurtähed asendada väiketähtedega; 2) jätta välja kõik kahetähelised lühendid: *st, jt* jne. Selleks võiks mingil kompel defineerida sõna pikkuse, ilmselt peaks sõna olema pikem kui kolm tähte. 3) tuleb välja otsida iga sõna esimesed kaks tähte ning nende hulgast välja valida need, milles mõlemad on konsonandid. Neist saab siis teha sagedussõnastiku. Töö etappide kaupa:

5) Valime faili: cat 30_ilu_ttxt
6) Asendame iga sõna eraldi reale, asendame suurtähed väiketähtedega: tr ' ' '\012' | tr '[A-Z]' '[a-z]' Sellega 1 sõna=1 rida. Kui on veel segavaid märke, kustutame need: tr –d',.?!:"-'
7) otsime välja read, mis sisaldavad vähemalt nelja tähte: grep '....'
8) lõikame välja iga rea esimesed kaks tähte: cut –c1-2
9) otsime nende hulgast välja need, milles nii eismene kui teine oleks kaashäälikud: grep '[bdghjklmnprstv][bdghjklmnprstv] '
10) teeme neist sagedussõnastiku: sort | uniq –c | sort-r

Kogu käsk: cat 30_ilu_ttxt | tr ' ' '\012' | tr '[A-Z]' '[a-z]' | tr –d',.?!:"-' | grep '....' | cut –c1-2 | grep '[bdghjklmnprstv][bdghjklmnprstv]' | sort | uniq –c | sort -r