

Metadatan käytöstä — nimien tärkeys

Panu A. Kalliokoski

Sisältö

1 Motivaatio: miksi tämä artikkeli on tärkeä?	1
1.1 Yksi yleinen työkalu on parempi kuin leegio erityisiä	2
1.2 Liittyykö tämä mitenkään Unix-järjestelmään?	2
2 Peruskäsitteet	3
2.1 Tiedostojärjestelmä	3
2.2 <i>Metadata</i>	3
2.3 <i>Hierarkia</i>	4
2.4 Hierarkian hyvät ja huonot puolet	4
3 Järjestelyvinkit	5
3.1 Hakemistojen käyttö	5
3.2 Tiedostojen nimeäminen	7
3.3 Muu metadata	8

1 Motivaatio: miksi tämä artikkeli on tärkeä?

Uskon, että yleisesti ottaen on vain kolme asiaa, joissa tietokone on hyvä (ja niistäkin vain viimeisessä se on tosi hyvä):

1. tiedon tarkka säilöminen
2. kommunikaatio, verkottuminen
3. tarkan tiedon automatisoitu käsittely

Kurssi paneutuu muilta osiltaan eniten kolmanteen kohtaan ja jonkin verran toiseen kohtaan. Ensimmäinen kohta on kuitenkin hyvin tärkeä, sillä se on suurimmalle osalle ihmisiä tietokoneen ensisijainen käyttötarkoitus (mikä on sääli): ihmiset käyttävät tietokoneita nykyään kaikkein eniten omien ja toisten tuotosten säilömiseen ja muokkaamiseen. Tämä artikkeli paneutuu siis ensimmäiseen kohtaan.

Jos olet käynyt kurssin ensimmäisen opintomonisteen, tiedät, miten tiedostoja ja hakemistoja siirrellään, kopioidaan, luodaan, muokataan ja tuhotaan. Olet ehkä oppinut, mitä hakemistot ja tiedostot *ovat*; jos olet edennyt näin pitkälle, onneksi olkoon, sillä tiedät huomattavasti enemmän kuin tietokoneen käyttäjät keskimäärin. Mutta vaikka oletkin oppinut, *miten* tiedostojärjestelmän eri juttujen kanssa leikitään ja ehkä jopa *mitä* sen eri osat ovat, puuttuu tieto siitä *miksi*, eli mikä on tiedostojärjestelmän tarkoitus.

1.1 Yksi yleinen työkalu on parempi kuin leegio erityisiä

Tämä osio on toistoa tttarkoitus-dokumentin teemoille. Jos olet lukenut sen jo, hyppää yli rauhassa.

Tietokone (ainakin ennen kuin vahvaa tekoälyä onnistutaan kehittämään) on "vain" työkalu. Se on erittäin hyvä ja monipuolinen työkalu, ja monipuolisuutensa takia hankala käyttää. On helppoa käyttää rajoitettua ja erikoistunutta järjestelmää, joka pakottaa tekemään asiat tietyllä tavalla. Vaikeampaa on käyttää yleispätevää järjestelmää, jossa pitää itse keksiä, miten mikin saadaan aikaan. Ja kuitenkin yleispätevän järjestelmän mahdollisuudet ovat valtavat verrattuna erikoistuneeseen.

Otetaan esimerkiksi erikoistuneesta järjestelmästä kalenteriohjelma (vaikka se onkin osa yleispätevää järjestelmää, tietokonetta). Se on rajoitettu ja pakottaa tekemään asiat tietyllä tavalla. Kaiken siihen syötetyn tiedon voisin yhtä hyvin kirjoittaa `emacs`:lla johonkin tiedostoon. Joutuisin itse päättämään, missä muodossa kirjoitan päivämäärät, aikarajat, merkintöjen kuvaukset ja muut niihin liittyvät tiedot. Mutta jos tekisin työni hyvin, voisin muutamalla `grep`-, `sort`- ja `cut`-käskyllä etsiä tiedostostani lähes mitä vain, paljon sellaistakin, mitä kalenteriohjelmani ei koskaan antaisi minun edes kysyä.

Tietokoneet ovat pullollaan yleispäteviä työkaluja. Usein johonkin tarpeeseen ostetaan erilaisia "ohjelmistoratkaisuja", vaikka sopivat työkalut ovat jo olemassa ja niitä on vain osattava käyttää. Niinpä ihmiset opettelevat käyttämään erityistä työkalua. Nämä eivät kuitenkaan koskaan pysty vastaamaan kaikkiin käyttäjien satunnaisiin tarpeisiin, koska niiden suunnittelijoille on mahdotonta ennakoida kaikkia tarvittuja toimintoja ja ominaisuuksia. Yleensä olisi parempi, jos tämä opettelu-aika ja -vaiva käytettäisiin yleispäteviin työkaluihin, niiden käyttöön ja yhteisiin käytäntöihin siitä, miten näitä yleispäteviä työkaluja käytetään.

Tiedostojärjestelmä on yksi tällainen yleispätevä työkalu. Se on suunniteltu yleiseksi ratkaisuksi tiedon tallettamiseen. Tämä ei ole mikään pieni tehtävä, eikä tiedostojärjestelmä aina pysty vastaamaan tietyn tietotyypin erityisvaatimuksiin. Näille yleensä on kehitetty erityisiä tiedontalletusratkaisuja. Kuitenkin suurimpaan osaan päivittäisestä tietojen hallinnasta tiedostojärjestelmä on juuri se, mitä kaivataankin. Siksi sitä on hyvä osata käyttää hyvin.

1.2 Liittykö tämä mitenkään Unix-järjestelmään?

Ei oikeastaan. Kaikki tämä on ihan yhtä tärkeää myös Windowsissa ja Macissa (ja VMS:ssa) - BeOS:sta en ole varma :) On joitain syitä, miksi tiedostojärjestelmä ja Unix liittyvät erityisesti toisiinsa:

- nykyaikainen hierarkkinen tiedostojärjestelmä vakiintui Unix:n myötä. (Unix ei toki ole ensimmäinen järjestelmä, jossa se esiintyi.)
- Unix yleensä panee ihmisen suoraan tekemisiin tiedostojärjestelmän rakenteen kanssa. Graafisissa ympäristöissä on tyypillistä, että systeemiin on yritetty rakentaa jonkinlainen "helppokäyttöisyys" päälle, joten niiden käyttäjät ovatkin tosi hukassa, kun joutuvat yhtäkkiä tiedostojärjestelmän kanssa tekemisiin.
- Unix:n tiedostojärjestelmä on yleispätevämpi eli siinä on vähemmän rajoituksia kuin yleensä on totuttu: nimet voivat olla kuinka pitkiä tahansa

ja niissä voi olla melkein mitä tahansa merkkejä, eikä sisäkkäisten hakemistojen määrälle ole rajoituksia. Ainoita erikoisnimiä ovat . ja .. sekä ainoa erikoismerkki /.

- Ohjelmat eivät `Unix`:ssa suosi talletusmuotonaan isoa klönttiä, johon on tungettu kaikki, vaan usein esim. saman dokumentin kuvat ja tekstien osat talletetaan erillisinä tiedostoina. Tämä siirtää sisällöllistä jaottelua käyttäjän näkyviin, tiedostojärjestelmän puolelle.

2 Peruskäsitteet

On hyvä aloittaa selvittämällä jonkin verran termejä ja niiden roolia ongelma-alueellamme.

2.1 Tiedostojärjestelmä

Nykyaikainen, hierarkkinen tiedostojärjestelmä koostuu tiedostoista, joita on karkeasti ottaen kahdenlaisia: hakemistoja ja “tavallisia tiedostoja”. Tavalliset tiedostot on suunniteltu toimimaan yleispätevinä tietoyksikköinä: käyttöjärjestelmä ei ota kantaa niiden sisältöön, vaan ohjelmat¹ voivat tallettaa niihin haluamiaan asioita, haluamassaan muodossa. Hakemistot on suunniteltu tavallisten tiedostojen järjestelyyn: hakemistoja selailemalla saa tietää, mitä tiedostoja on olemassa, ja tiedostoja voi jaotella panemalla niitä eri hakemistoihin.

2.2 *Metadata*

Tavallinen tiedosto koostuu karkeasti ottaen kahdesta osasta: sisällöstä ja metadatatista². Jälkimmäinen tarkoittaa suunnilleen “tietoa tiedosta”. Tiedoston sisältö on tietoa, esimerkiksi jokin kuva; tiedoston metadata on tietoa tuosta kuvasta, eli tietoa siitä, mitä ja minkälaista tietoa sisältö on.

`Unix`:n tiedostojärjestelmän metadata on perin vaatimatonta. Tiedostosta on sisällön lisäksi muistissa oikeastaan vain kolme asiaa:

1. nimi ja se, missä hakemistossa tiedosto sijaitsee
2. käyttöoikeudet (omistaja, omistajaryhmä ja oikeusasetukset)
3. ajat, jolloin tiedostoa on viimeksi muutettu, luettu tai muutettu metadattaa

Koska ainoa näistä, johon voi vaikuttaa suoraan ja lähes rajoituksitta, on tiedoston nimi ja sijainti, sen merkitys on ylivoimainen muihin nähden.

Mutta mitä sillä metadatatalla sitten tehdään? Kuvittele järjestelmä, jossa ei olisi hakemistoja eikä nimiä, vaan tiedostoihin viitattaisiin yksinkertaisesti numeroilla. Aina, kun luotaisiin uusi tiedosto, otettaisiin seuraava vapaa numero sille tunnisteeksi. Tällaisessa järjestelmässä on monta vikaa: tiettyä tiedostoa on vaikea löytää; jos on kiinnostunut tietystä tiedostojoukosta, muut ovat tiellä

¹tai käyttäjä kuten yllä olevassa `emacs`-esimerkissä

²En tunne tälle termille hyvää suomenkielistä käännöstä; joka tapauksessa se on kielellinen sekasikiö, joka tulee kreikan etuliitteestä “meta” (*tuolla puolen, takana*) ja latinan sanasta “datum” (sananmukaisesti *annettu*, mutta tarkoittaa nykyään *tietoa*).

sekoittamassa ja hämmentämässä; ja jos yrittää löytää tiedostoja tietyllä kriteerillä, esimerkiksi kaikkia väliaikaistiedostojaan³, on vaikeuksissa.

Metadata tuo tietoon järjestystä. Itse asiassa se on kaiken tehokkaan ja nopean tiedonkäsittelyn perusta. Mutta metadatasissa on yksi suuri ongelma: käyttäjän pitää kertoa se enimmäkseen itse (ainakin ennen vahvan tekoälyn kehittämistä). Tietokoneelle ei voi sanoa: “Näytäpä minulle se artikkeli, jonka kirjoitin joskus viime viikolla Birgit & Claes -lehteä varten.” Jos aikoo löytää artikkelin näillä kriteereillä, metadataan on lisättävä puuttuvat osat: esimerkiksi järjestettävä artikkelit omaan hakemistoonsa, ja mahdollisesti vielä sielläkin lehdittäin alihakemistoihin. Jopa ajan suhteen joutuu ehkä itse antamaan metadatasia, vaikka tiedostojärjestelmä viime muutosajoista kirjaa pitääkin.

2.3 Hierarkia

Nimien käyttö metadatan antamiseen näyttää olevan ihmisille luontaista, onneksi. Melko harvat tietokoneenkäyttäjät nimeävät kirjoituksensa “Dokumentti 1”, “Dokumentti 2” ja niin edelleen. Valitettavasti nimien käyttö ei riitä, sillä nimeen on vaikea saada kaikkea olennaista metadatasia ja joka tapauksessa tiedostojen kertyessä koko listan läpi selaaminen alkaa olla hyvin uuvuttavaa. Tarvitaan vahvempia lääkkeitä. Tällainen lääke on hakemistojen käyttö.

Nykyaikaisen tiedostojärjestelmän suuri keksintö on hierarkia⁴. Sillä tarkoitetaan, että hakemistot voivat sisältää tiedostojen lisäksi toisia hakemistoja, nämä taas puolestaan uusia hakemistoja, ja niin edelleen, rajoittamattomasti. Hakemistohierarkian rakenne antaa paljon metadatasia tiedostosta. Esimerkkinä tämän tiedoston täydellinen (absoluuttinen) nimipolku kannettavalla tietokoneellani:

```
/home/atehwa/proj/mtx/unix/nimeys.stx
```

Komponentti	merkitys
home	käyttäjien omat tiedostot
atehwa	käyttäjän atehwa
proj	projektit
mtx	sekalaiset tekstit
unix	Unix-kurssimateriaali
nimeys.stx	juttu nimeämisestä + tiedostotyyppi (stx)

2.4 Hierarkian hyvät ja huonot puolet

Hierarkista tiedostojärjestelmää on sekä ylistetty että haukuttu välineenä, ja aiheesta. Yleinen ongelma on se, että ihmiset eivät osaa käyttää sitä. On vaikea ellei mahdoton tietää ennalta tai usein edes jälkikäteen, millaiseen hierarkiaan tiedostot kannattaisi sijoitella, jotta ne pystyisi kätevästi löytämään ja yhdistelemään. Tämän dokumentin loppuosa on paljolti käytännön vinkkejä ja neuvoja tähän tehtävään.

³siis tiedostoja, jotka on tallettanut vain väliaikaisesti tiedostojärjestelmään eikä varsinaisesti säilytettäväksi

⁴Sana hierarkia on kreikkaa ja tulee sanoista hiero(s) (*pappi*), ja arkhe (*perusta, valta*), siis pappisvalta. Termi on yleistynyt tarkoittamaan mitä tahansa järjestelmää, jossa yksiköillä / yksilöillä on aina toinen, jolle se on alistettu. Tiedostojärjestelmän yhteydessä tämä viittaa siihen, että kaikki hakemistot sijaitsevat jossain toisessa hakemistossa, ovat sen “alihakemistoja”.

Osa hierarkian käyttövaikeuksista johtuu yksinkertaisesti siitä, etteivät ihmiset ole tottuneet miettimään ja käyttämään metadataa, mutta osa vaikeuksista on aitoja. Hierarkia esimerkiksi on huono kuvaamaan ortogonaalisten (toisistaan riippumattomien) piirteiden vaihtelua. Oletetaan, että minulla on tekstejä ja kuvia, jotka ovat peräisin eri vuosilta. Kannattaako minun järjestää ne vuosittain hakemistoihin, joissa jokaisessa on alihakemistot **tekstit** ja **kuvat**, vai kannattaako tehdä hakemistot **tekstit** ja **kuvat**, joissa eri vuodet ovat alihakemistoina? Vai onko jompikumpi piirre niin epäolennainen, ettei sen perusteella kannata jaotella ollenkaan? Valinta tehdään tuntumalla tai mielivaltaisesti, ja sillä voi olla yllättäviä vaikutuksia myöhemmin.

Voi myös syntyä tilanteita, jossa tiedosto kuuluu loogisesti kahteen paikkaan hierarkiassa. Tämä on harmillista, koska kumpaan paikkaan tahansa tiedoston paneekin, se on jostain näkökulmasta epäintuitiivisessa paikassa. Jos esimerkiksi olen jaotellut tekstini aihepiireittäin mutta jokin teksti liittyy sekä leijanlennätykseen että ohjelmointiin, minulla on vain huonoja vaihtoehtoja: voin sijoittaa sen ohjelmointiteksteihin, jolloin leijanlennättäjät eivät löydä sitä, tai päin vastoin; tai sitten voin panna sen molempiin, jolloin joudun pitämään yllä kahta erillistä versiota ja huolehtimaan, että ne ovat aina molemmat ajan tasalla. Tai sitten voin tehdä oman aihepiirinsä, leijanlennätys+ohjelmointi, mikä monimutkaistaa hierarkiaa.

Toisaalta hierarkialla on myös joitain hienoja ominaisuuksia, jotka löytyvät harvasta muusta järjestelmästä. Hierarkian jokainen osa on myös hierarkia, joten jokaisen hakemiston alla⁵ voi vallita omansalainen järjestys häiritsemättä muiden hakemistojen järjestystä vähimmässäkään määrin (esimerkiksi proj-hakemiston sisältö on arvatenkin järjestetty projekteittain, mutta muissa hakemistoissa on toisenlainen järjestys). Tämän takia myös sotkut voi siirtää omaan hakemistoonsa häiritsemättä muiden hakemistojen käyttöä.

Kahdesta asiasta voi kuitenkin olla varma:

1. vain kokemus ja harjoittelu tuo intuitiota sen suhteen, miten hierarkia kannattaa rakentaa. (Mutta yrittänyttä ei laiteta.)
2. hierarkiat elävät: niiden tarkoituksenmukainen rakenne vaihtuu sitä mukaa, kun niiden sisältö kehittyy. Niinpä uudelleenjärjestelyä silloin tällöin ei voi välttää.

3 Järjestelyvinkit

3.1 Hakemistojen käyttö

Hakemistojen tarkan käytön määrää siis järjesteltävän tiedon luonne. On kuitenkin monta nyrkkisääntöä, joita noudattamalla saattaa päästä alkuun tai pitkälle (ja silti, yksikään näistä ei ole kumoamaton):

Hyvän hakemiston listaus on vastaus johonkin kysymykseen Kunnon hakemisto sisältää jonkin mielekkään kokonaisuuden. Hakemiston alla sijaitsevilla tiedostoilla pitäisi kaikilla olla jotain yhteistä, esimerkiksi kaikki **bin**-hakemiston alla olevat asiat voivat olla ohjelmia.

⁵eli siis hakemistossa, sen alihakemistoissa, tai näiden alihakemistoissa, jne.

Lisäksi hakemistossa välittömästi sijaitsevat tiedostojen tai hakemistojen pitäisi “puhua samasta asiasta”, olla toisensa poissulkevia. Ei kannata esimerkiksi panna samaan hakemistoon alakategorioita tiedostoille sen mukaan, keneltä ne on saatu, ja sen mukaan, mihin niitä aiotaan käyttää, vaan kannattaa alistaa jompikumpi näistä jaotteluista toiselle.

Myös, jos esimerkiksi kuvat jaotellaan aihepiireittäin, ei välttämättä sekalaisia kuvia kannata jättää suoraan kuvat-hakemistoon. Sen sijaan kannattaa tehdä aihepiiri muuta ja panna sekalaiset sinne.

Tiedostot, joita ei ole mitään tapaa / syytä jaotella, tulee sijoittaa omaan hakemistoonsa erilleen muusta hierarkisoinnista. Tämän hakemiston listaus on vastaus kysymykseen: “Mitä kaikkea järjestelmätöntä taukkaa minulla on?”

Hyvä hierarkia vastaa mahdollisimman moneen kysymykseen Sen lisäksi, että etsit usein tiedostoja yksitellen, haluat niistä joskus yhteenvedoja tai haluat kiinnittää huomiosi johonkin tiedostojen ryhmään. Hyödyllisten ja runsaiden vastausten löytyminen hakemistolistauksista helpottaa kaikkia näitä päämääriä.

Kunnollisten vastausten löytyminen on synergeettistä: joskus vastauksia yhdistelemällä saa esiin tietoa, jonka tarvetta ei osannut ennakoida. Esimerkiksi jos väliaikaistiedostoni on jaoteltu projekteittain ja projektini vuosittain, pystyn etsimään loppuneiden projektieni väliaikaistiedostot. (Todellisuudessa mieluummin panen projektieni väliaikaistiedostot projektikansioihin eikä erilliseen hierarkiaansa.)

Hyvä hierarkia on käytännöllinen Tämä on edellisen vastavoima. Hyvä hierarkia ei aiheuta ylimääräistä vaivaa. Jos joudut kirjoittamaan kohtuuttoman pitkiä tiedostonimiä tai ylimääräisiä kommentoja selvittääksesi hierarkiasi kanssa, on jotain vialla. Samoin jos hierarkian yksi taso sisältää vain 1–3 alakohtaa, taso saattaa olla turha. Itse asiassa usein hyvän jaottelun tunnistaa siitä, että alakohtia on 5–10.

Pidä päätaso siistinä *Päätaso* tarkoittaa sellaista kohtaa hierarkiassa, jolla on erityisasema jostain sosiaalisesta tai ohjelmien toimintaan liittyvästä syystä. Esimerkkejä päätasoista ovat kotihakemisto (koska kaikki omat tiedostosi ovat sen alla ja koska se on oletushakemistosi sisään kirjautuesssa), www-sivujen ylin hakemisto (koska kaikki www-sivusi ovat sen alla), sähköpostikansioittesi ylin hakemisto, sekä erilaiset jaetut projektihakemistot (koska hakemiston sijainti on jaettu tieto, jonka muuttumisesta pitää tiedottaa).

Päätason siistinä pitäminen on tärkeää useasta syystä: päätason kanssa joutuu säännöllisesti tekemisiin, joten jos sen selailuun menee aikaa, aiheutuu paljon vaivaa; päätason sisällölle ei yleensä ole olemassa mitään itsestäänselvää jaottelutapaa; ja päätaso on usein helpoin paikka tunkea vaikeasti jaoteltavia, epämääräisiä tiedostoja, joten sen siivoamisen pitää myös olla helppoa.

Suosittelen, että päätasoa ei koskaan käytetä muiden kuin sellaisten tiedostojen säilömiseen, joiden on pakko olla siellä tai jotka voi tuhota melkein heti. Näiden lisäksi päätasoille voi luoda alihakemistoja, jotka ovat jokin

sillä hetkellä hyvältä vaikuttava jaottelu kaikelle päätason alla olevalle. Mutta varsinaista sisältöä ei siis aseteta suoraan päätasolle.

Siivoa tarpeeksi aikaisin Kun huomaat ärtyväsi hakemiston selaamisesta, on aika siivota. Siivoaminen tarkoittaa turhien tiedostojen poistamista ja mahdollisesti sisällön järjestelemistä uudelleen. Mitä myöhemmin siivoat, sitä suurempi vaiva kunkin yksittäisen tiedoston lajittelussa on.

Kun siivoat, yritä luoda jaottelu, johon saa lajitelluksi *kaikki* hakemiston tiedostot, ja joka tekee mahdolliseksi löytää tietystä alakohdasta tiedoston selaamatta kaikkia alakohtia läpi. Jotain hyötyä on myös siitä, jos tiedostot jakautuvat tasaisesti alakohtien välille.

Luo hierarkia “tiputtamalla” Sen sijaan, että yrittäisit tehdä mahdollisimman täydellisen hierarkian alusta alkaen, voit tehdä ylimalkaisen ja katsoa, miten se toimii. Aina, kun jokin hakemisto näyttää sotkuiselta, jaottele se uudestaan: “tiputa” tiedostot alemmas. Älä pelkää poistaa vanhoja, huonoiksi osoittautuneita jaotteluita: mitä nopeammin pääset niistä eroon, sitä nopeammin saat tiedostot hyvään hierarkiaan.

Väliaikaisuuden huomioon ottaminen Mitä pysyvämpiä tiedostot ovat, sitä enemmän niiden lajittelusta on hyötyä. Niinpä vähemmän pysyvät tiedostot kannattaa usein tuoda pois pysyvämpien tarpeettoman perusteellisesta jaottelusta, omaan hakemistoonsa (olettaen, etteivät ne liity suoraanaisesti johonkin jo hierarkiassa olevaan). Tässä voi olla monta tasoa: pysyvät tuotokset, kausittain vanhenevat tiedot, tiedostot, jotka voi poistaa tehtyään jonkin homman, ja vain läpikulkumatalla olevat tiedostot.

3.2 Tiedostojen nimeäminen

Jos tiedostojen järjestely hakemistoihin on hoidettu kunnolla, hyvä nimeäminen ei ole suuri ongelma. Tässä on silti muutamia ohjeita:

Hyvä nimi kertoo, mikä on kyseessä Tiedoston nimeä ei sanota turhaan nimeksi. Tiedoston nimi on jotain yleis- ja erisnimen väliltä, koska sen pitäisi toisaalta kuvata, mikä tiedosto on, toisaalta erottaa se muista tiedostoista. Esimerkiksi Kallea potalla esittävälle kuvalle hyvä nimi on vaikkapa `kalle-potalla.jpg`.

Hyvästä nimestä käy ilmi tiedoston tyyppi ja asema Usein joutuu etsimään tiedostoja tai -joukkoja riippumatta siitä, missä kohtaa jaottelua ne sijaitsevat - esimerkiksi laskeakseen kaikkien videoiden tilankulutuksen. Tyypillisesti tässä auttavia asioita ovat:

1. tiedoston tyyppin (tiedostomuodon) kertominen nimessä. Tiedostomuoto kerrotaan yleensä lyhyellä päätteellä, joka erotetaan muusta nimestä pisteellä, esimerkiksi `.txt` tekstille.
2. tiedoston merkistön, kielen, tai tiedostolle tapahtuneen prosessoinnin kertominen nimessä. Esimerkiksi päätteestä `.html.fi.gz` arvaa, että kyseessä on suomenkielinen www-sivu, joka on pakattu gzip-ohjelmalla.

3. tiedoston aseman kertominen nimessä. Esimerkkejä asemaan vaikuttavista tekijöistä ovat tärkeys, kiireellisyys, autoritatiivisuus eli onko tiedosto pääversio vai siitä tehty kopio, varmuuskopioisuus⁶, ja niin edelleen. Jotkut nimeävät kaikki roskatiedostonsa `,-etuliitteellä` ja kiireelliset tiedostonsa esim. `XXXX--etuliitteellä`.
4. erityistiedostojen vakiintuneet nimet. Esimerkkejä vakiintuneista nimistä ovat `TODO` tehtäville asioille, `CHANGES` tai `ChangeLog` muutostokille, `INSTALL` asennusohjeille.

3.3 Muu metadata

Joskus osa tiedostojen metadatasta on kirjoitettu itse tiedoston sisältöön. Esimerkiksi `HTML`-sivut sisältävät tiedon otsikostaan sekä mahdollisesti kirjoittajastaan, aiheanoistaan, muista asiaan liittyvistä sivuista ja niin edelleen. Samoin tiedostojen muutamalla ensimmäisellä tai viimeisellä rivillä on joskus editoriohjelmille ohjeita, jotta nämä tietäisivät, millä asetuksilla tekstiä pitää muokata. Ohjelmatiedostot sisältävät usein kommentteissaan lisenssiehtoja ja huomautuksia tiedoston asemasta. Jotkin versiohallintajärjestelmät käyttävät tiedostoihin merkittyjä tunnisteita, joiden perusteella tiedetään siirtämisenkin jälkeen, mistä tiedostosta on kyse. Tällaisen metadatan käyttämisen mahdollisuus ja hyödyllisyys riippuu tietenkin tiedostomuodosta.

Hakemistoihin ei voi samaan tapaan kirjoitella satunnaista metadataa. Siksi pä onkin tullut käytännöksi kirjoittaa hakemistojen huomautukset tietynnimiisiin tiedostoihin, jotka sijaitsevat kyseisessä hakemistossa. On esimerkiksi hyvin vahva perinne, että hakemistossa oleva `README`-tiedosto sisältää huomautuksia hakemiston sisältöön liittyen ja `LICENSE`-tiedosto hakemiston sisällön lisenssiehdot. Yleensäkin on tapana käyttää isoja kirjaimia sellaisten tiedostojen nimissä, jotka ovat hakemiston metadataa tai infrastruktuuria, ja pelkkiä pieniä kirjaimia varsinaisten sisältötiedostojen nimissä.

Usean ihmisen projekteissa on muuten hyvin olennaista, että hakemistoissa *on* kunnolliset `README`-tiedostot.

⁶jonka `emacs` merkitsee `~`-pääätteellä