

# **Open source historia**

Timo Nyysönen

Helsinki 3. huhtikuuta 2002

Tietokäsittelytieteen historia - seminaari

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

# Sisältö

<b>1 Historia</b>	<b>1</b>
1.1 Stallman ja MIT . . . . .	1
1.2 GNU ja FSF . . . . .	3
1.3 Linux . . . . .	5
<b>2 Ideologia ja lisenssit</b>	<b>7</b>
<b>3 Yhteenveto</b>	<b>10</b>
<b>Lähteet</b>	<b>11</b>
<b>Liitteet</b>	
<b>1 Käytetyt lyhenteet</b>	

# 1 Historia

Open Sourcen eli vapaan lähdekoodin voidaan katsoa olevan yhtä vanhaa kuin tietokoneetkin. Termi Open Source opettiin käyttöön vasta vuonna 1998. Karkeasti ottaen samat ideologiat ja periaatteet ovat olleet esillä vuosikymmeniä eri nimillä. Samanlainen ajattelu- ja toimintatapa on Free Software eli vapaat ohjelmistot. Myös termejä Freed/Liberated Software (Vapautettu ohjelmisto) käytetään. Termien erot selvitetään myöhemmin.

## 1.1 Stallman ja MIT

1960-luvulla nähtiin ensimmäisiä Open Source-toiminnan piirteitä laajemmassa mitakaavassa. Toki jo ennen tätäkin ilmiötä esiintyi, mutta erityisesti akateemiset piirit alkoivat harjoittamaan sitä 1960-luvulla. ARPANETin aloittaminen vuonna 1969 vaikutti toiminnan leviämiseen. Silloin Open Source tarkoitti valmiiden ohjelmien jakamista muiden käyttöön lähdekoodeineen. Tutkijat yliopistoissa luovuttivat ilomielin omia tuotoksiaan muiden käyttöön ja muunneltavaksi. Toiminta oli siis lähes samanlaista kuin tänäkin päivänä, erona että mitään määritelmiä miten ohjelmia saa käyttää ei ollut. Yleisintä ohjelmien ja lähdekoodin jakaminen oli MIT:n AI-laboratoriossa (tekoälylaboratorio).

Richard Stallman sai kesätyöpaikan vuonna 1971 MIT:n AI-laboratoriossa. Hänen tehtävänään oli lisätä ominaisuuksia Digitalin PDP-10-koneen käyttöjärjestelmään. Käyttöjärjestelmä ITS oli MIT:ssä syntyneen hakkeriliikkeen tuotos. ITS oli kirjoitettu assembler-kielellä, mutta monet lisäykset siihen kirjoitettiin LISPillä. Stallmanin työn tuloksena oli Emacs [Emacs]. Se levisi vapaana ohjelmistona yhä laajemmalle, mutta ennen kuin Stallman alkoi levittämään Emacsia, hän loi epävirallisen säännön. Sen mukaan ohjelmaan tehdyt korjaukset oli lähetettävä Stallmanille. Tästä epävirallisesta säännöstä muodostui yksi koko vapaan lähdekoodin ohjelmistojen kehityksen kulmakivistä.

Näin syntynyt ohjelmia ja lähdekoodia levittävä yhteisö koki kolauksen 1980-luvun

alkupuolella, kun Digital lopetti PDP-10-koneen valmistamisen. PDP-10 oli viimeinen kone, jolla ITS toimi. MIT:n AI laboratorion työyhteisö oli hajonnut jo aiemmin, kun monet sen työntekijät siirtyivät juuri perustetun Symbolics-nimisen yhtiön palvelukseen. Symbolics kehitti Lisp Machinea eli tietokonetta, joka on optimoitu ajamaan LISP-koodia tehokkaasti ja tarjoaa hyvän ohjelmointiympäristön LISPille. Alunperin Lisp Machine oli kehitetty MIT:n AI-laboratoriossa. Toinen yhtiö, joka kehitti Lisp Machinea, oli sen alkuperäisen suunnittelijan Richard Greenblattin perustama Lisp Machine Incorporated (LMI). Stallman ei halunnut lähteä pois MIT:sta, vaan hän alkoi lisäämään Symbolicsin Lisp Machineen tekemiä ominaisuuksia AI-laboratorion käyttämään versioon. Stallman jatkoi kehitystyötä lähes yötäpäivää melkein kaksi vuotta. Kehittäjät ajautuivat kuitenkin erilleen, eikä ohjelmat olleet enää yhteensopivia. Toinen seikka, mikä häiritsti Stallmania, oli MIT:n Lisp Machinen lisensointi. Se oli lisensoitu Symbolicsille ja LMI:lle, eikä ollut enää vapaata ohjelmistoa.

Stallman joutui valinnan eteen ja hän näki kolme vaihtoehtoa. Yksi oli siirtyä kaupallisen ohjelmistotuotannon pariin ja tehdä rahaa. Tämä kuitenkin soti vastoin kaikkia hänen periaatteitaan, joten se oli poissa laskuista. Toinen vaihtoehto oli lopettaa työskentely tietokonealalla. Mutta siinä tapauksessa Stallman ajatteli, että hänen lahjansa menisivät hukkaan. Kolmas vaihtoehto oli aloittaa taistelu ja miettiä olisiko hänen mahdollista kirjoittaa ohjelma tai ohjelmia, jotka saisivat jo aiemmin syntyneen vapaan lähdekoodin yhteisön taas aktiiviseksi. Vastaus oli Stallmanille selvä: Ensin tarvitaan käyttöjärjestelmä. Käyttöjärjestelmällä voi tehdä monia asioita, mutta ilman sitä ei voi tehdä mitään, joten se oli luonnollinen valinta lähtökohdaksi. Lisäksi Stallmanin aiemman käyttöjärjestelmäohjelmointikokemuksen ansiosta hän tunsikin kykenevänsä siihen. Stallman valitsi lähtökohdaksi Unixin, vaikka hänellä ei ollut siitä aiempaa kokemusta. Syitä oli kaksi. Stallman halusi, että järjestelmä voidaan siirtää konetyypistä toiseen ja Unix oli ainoa tällainen järjestelmä jonka Stallman tiesi. Lisäksi kun järjestelmä olisi yhteensopiva Unixin kanssa, uusien käyttäjien olisi helppoa siirtyä siihen. Kehitettävän ohjelman nimeksi tuli GNU [GNU], joka on rekursiivinen lyhenne sanoista GNU's Not Unix.

## 1.2 GNU ja FSF

Vuonna 1984 Stallman lopetti työskentelyn MIT:ssa alkaakseen työskentelemään GNU:n parissa. Lähteminen oli käytännön sanelemana pakko, koska muuten MIT olisi voinut puuttua GNU:n perusajatukseen, vapaaseen ohjelmistoon. Ensiksi Stallman halusi korvikkeen Yacc:lle [Joh79]. Yacc on kontekstittomien kielioppien jäsentäjä. Lähtökohdaksi hän valitsi Robert Corbettin kirjoittaman Bison:in [Bison]. Stallman sai luvan Corbettilta muokata Bison:sta yhteensopiva Yacc:in kanssa siten, että kaikki Yacc:in kieliopit toimivat Bison:lla. Näin syntyi GNU:n ensimmäinen palanen. Ennen GNU projektin aloittamista Stallman oli kuullut Andrew Tanenbaumin kirjoittamasta Free University Compiler Kitistä, joka oli kääntäjä mm. C:lle ja Pascalille. Kääntäjä toimi Unixissa, joten Stallman pyysi lupaa käyttää kääntäjää GNU projektissa. Tanenbaum kuitenkin kielsi käytön perustelemalla, että vaikka yliopisto on ilmainen niin yliopiston käännösohjelma ei ole. Stallman halusi säästää omaa työmääräänsä ja jatkoi valmiin kääntäjän etsimistä, sen sijaan, että olisi kirjoittanut kääntäjän itse. Hän saikin Lawrence Livermore Lab:n kirjoittaman Pastel kääntäjän lähdekoodin käyttöönsä. Se ei kuitenkaan tukenut C-kieltä ja Stallman kirjoitti siihen laajennuksen C:lle ja alkoi siirtämään sitä Motorola 68000 tietokoneelle. Hän luopui hankkeesta huomattuaan, että kääntäjä tarvitsi monen megatavua pinomuitia, mutta käytössä olleessa järjestelmässä saattoi olla vain 64 kilotavun pino. Pastel nimittäin jäseni koko lähdetiedoston ensin syntaksipuuksi, jonka se sitten muutti käskyjen jonoksi, josta lopulta muodostettiin käännetty tiedosto ja muistia ei vapautettu ollenkaan koko prosessin aikana. Stallman lopetti valmiin kääntäjän etsimisen ja aloitti ohjelmoimaan kääntäjää alusta alkaen. Hän sai osittain käytettyä koodia, jonka oli kirjoittanut Pastelia varten ja tuloksena oli GCC.

Jo ennen GCC:n valmistumista, heti kun Stallman huomasi ettei Pastelin laajennus onnistu, hän palasi 10 vuotta vanhan projektinsa pariin. Hän alkoi tehdä Emacsista GNU Emacsia. Toiminnallisesti GNU versio muistutti vanhaa Emacsia, mutta Stallman kirjoitti koodin kokonaan uudestaan. Toinen muutos oli, että GNU Emacs tulisi käyttämään laajennuskielenä LISP:iä. Stallman kuitenkin lainasi koodia James Gos-

lingin kirjoittamasta Gosling Emacsista, jonka lähdekoodi oli aluksi vapaata. Gosling möi myöhemmin Gosling Emacsin oikeudet UniPress-yhtiölle, joka vaati Stallmania lopettamaan Emacsin levittämisen. Stallman oli myynyt GNU Emacsia 150 dollarin hintaan magneettinauhoilla. Stallman korvasi lainaamansa osat ja GNU Emacsia voitiin taas levittää. Välttääkseen jatkossa tämänkaltaisen ohjelmistojen oikeuksien myymisen, Stallman kehitti yhdessä lakimiehen Eben Moglen kanssa lisenssin, joka ei rajoita käyttäjien oikeuksia vaan suojaa niitä. Ensimmäinen tämänlainen lisenssi oli GNU Emacs General Public License. Lisenssi takasi käyttäjänoikeuden, tekijänoikeuden vastineen. Se antaa käyttäjille oikeuden kopioida ja muuttaa alkuperäistä ohjelmaa ja levittää sitä. Muokattujen ohjelmien tulee olla saman lisenssin alaisia, eli käyttäjänoikeuksien tulee pysyä samoina. Tämä lisenssi oli lähtökohtana yleisemmälle lisenssille GNU General Public License:lle (GNU GPL). Lisää lisensseistä kerrotaan luvussa 2.

Emacs on siis merkittävä ohjelma vapaiden ohjelmistojen historiassa. Ensinnäkin monet vapaat ohjelmistot ovat kirjoitettu Emacsilla ja se on merkittävä tekstieditori edelleen, mutta myös sitä kehitettäessä syntyi kaksi merkittävää vapaiden ohjelmien käsitettä. Vapauden käsite sinänsä, eli että vaikka ohjelma on vapaa (free) sen ei tarvitse olla ilmainen. Toinen syntynyt käsite on vapaiden ohjelmistojen lisenssit.

Stallman oli saanut tuloja Emacs-nauhojen myynnistä, GCC:n pyydettyjen lisäysten tekemisestä ja GCC:n käytön opetuksesta. Näiden varojen avulla hän perusti Free Software Foundation:in (FSF) vuonna 1985. Aluksi Stallmanilla ei ollut varaa palkata ketään, mutta ajan myötä säätiön toiminta kasvoi ja säätiön palveluksessa oli 1980-luvun lopussa kymmeniä työntekijöitä. FSF on omistettu edistämään tietokoneiden käyttäjien oikeutta käyttää, tutkia, kopioida, muuttaa ja uudelleenlevittää tietokoneohjelmia [FSF].

Koska GNU ohjelmat olivat Unix yhteensopivia, ne levisivät laajalle ennen kuin oli edes olemassa GNU käyttöjärjestelmää. Käyttäjät siirsivät ohjelmia omaan ympäristöönsä ja tekivät niihin lisäyksiä. Tämä edesauttoi ohjelmien kehitystä ja teki niistä parempia, mutta samalla hidasti koko GNU projektin valmistumista. GNU kehittä-

ijen aika kului siirrettyjen versioiden ylläpitoon ja ominaisuuksien lisäämiseen, sen sijaan, että he olisivat voineet ohjelmoida puuttuvia osia.

Vuonna 1990 GNU järjestelmä oli lähes valmis, mutta yksi tärkeä osa puuttui, käyttöjärjestelmän ydin. Vuonna 1986 Stallman suunnitteli GNU:n ytimeksi MIT:ssa kehitettyä TRIX:a. FSF löysi kuitenkin paremmin ratkaisun heinäkuussa 1987. Sen sijaan, että he olisivat parannelleet TRIX:a, kehittäjät päättivät rakentaa ytimen Carnage Mellon yliopistossa kehitetyn Mach [Mach] mikroytimen päälle. Ydin nimettiin lopulta GNU Hurd:ksi. Tällöin ydin koostuisi joukosta 'palvelimia', joita ajetaan Mach:n päällä. Yksi syy tämänkaltaiseen ratkaisuun oli välttää ytimen debuggaus ilman lähdekoodidebuggeria. Kirjoitettava osa ytimestä olisi joukko ohjelmia, joita voitaisiin helpommin debugata jo olemassa olevalla GDB:llä.

### 1.3 Linux

1990-luvun alkupuolella tietojenkäsittelytieteen opiskelija Linus Torvalds Helsingin yliopistossa tutustui ja ihastui Unixiin. Intelin 80386-prosessori tuli markkinoille ja Torvalds huomasi, että se on oli paljon parempi kuin yksikään aikaisemmista kotitietokoneiden prosesseista. Torvalds tiesi, että Unixin voi asentaa myös PC:lle, joten hän hankki PC:n vuonna 1991, vaikka ei aiemmin ollut sellaista halunnutkaan. Torvalds asensi pian koneeseensa Tanenbaumin kirjoittaman Minix [Minix] järjestelmän.

Minix syntyi kun Tanenbaum ei voinut enää opetuksessaan käyttää AT&T:n Unixin lähdekoodia, koska vuonna 1979 ilmestyneen seitsemännen version lisenssi kielsi sen. Vuonna 1984 Tanenbaum alkoi kirjoittamaan käyttöjärjestelmää, joka olisi yhteensopiva Unixin kanssa. Kahden vuoden kuluttua Tanenbaum julkaisi Minixin yhdessä kirjan kanssa, jossa selvitettiin sen toimintaa.

Torvalds tutustui Minixiin koneellaan ja kirjoitti ohjelmia C:llä ja assemblerilla. Hän opiskeli myös Minixin toteutusta lähdekooditasolla ja alkoi tekemään siihen lisäyksiä ja korjauksia. Torvalds oli kirjoittanut myös pääte-emulaattorin Usenet-utisryhmien lukua varten kotoaan modeemin välityksellä. Hän huomasi, että hän

halusi myös ladata ohjelmia verkosta, joten hänen täytyi kirjoittaa levyajuri. Jotta Torvalds olisi voinut käyttää levyä, täytyi hänen täytyä kirjoittaa tiedostojärjestelmä. Tämän hän teki Minixin tiedostojärjestelmän pohjalta. Vähitellen Minix tuli tarpeettomaksi ja Linuxin esi-aste oli syntynyt. Pian tämän jälkeen Torvalds ilmoitti maailmalle tekemästään käyttöjärjestelmästä Usenetin comp.os.minix-uutisryhmään seuraavalla viestillä, jota monet pitävät Linuxin syntyhetkenä.

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)

Newsgroups: comp.os.minix

Subject: What would you like to see most in minix?

Summary: small poll for my new operating system

Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>

Date: 25 Aug 91 20:57:08 GMT

Organization: University of Helsinki

Hello everybody out there using minix - I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things). I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-). Linus (torvalds@kruuna.helsinki.fi)  
PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT portable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-).

Muutamaa viikkoa myöhemmin Linus julkaisi version 0.01 Linuxista. Tässä versiossa toimivat jo GCC, Bash ja useimmat GNU ohjelmat. Tämä versio oli kaik-



kien saatavilla Funetin FTP-palvelimelta. Vaikka useimmat Linuxin työkalut olivat GNU ohjelmia ja siis GPL:n lisenssin alaisia, niin Linuxin ydin ei ollut GPL:n alainen vaan Torvaldsin itse laatiman tekijänoikeuden alaisia. Lähdekoodi oli saatavilla ja muokattavissa, mutta tekijänoikeuksissa oli ehto, joka kielsi ottamasta maksua Linux-jakelusta. Yksi syy tähän oli se, että Torvalds oli pettynyt Minixin maksullisuuteen. Maksunottamisen kieltö esti Linuxin leviämistä, koska edes kopiointikustannuksista ei voinut ottaa maksua. Useiden pyyntöjen jälkeen Torvalds kuitenkin suostui muuttamaan lisenssiä ja valitsi lopulta Linuxin ytimen version 0.12 lisenssiksi GPL:n. Linux levisi ympäri maailmaa ja kehittyi koko ajan. Vaikka Torvalds oli yksin vastuussa Linuxista, sadat ohjelmoivat lähettivät hänelle korjauksia

Korjauksia ja parannusehdotuksia tuli kuitenkin niin paljon, että Torvaldsin oli mahdollonta selviytyä niistä yksin. Hänellä oli apunaan ”luutnantteja” jotka olivat vastuussa tietystä ytimen osasta ja suodattivat korjausehdotukset ja lähettivät ne edelleen Linuxille. Merkittävimpiä luutnantteja tässä vaiheessa olivat Alan Cox ja Ted Ts'o. Cox oli kirjoittanut Linuxin TCP/IP-pinon ja vastasi Linuxin verkko-osasta. Ts'o vastasi useammasta alueesta ja oli ensimmäisen Linuxin FTP-peilin perustaja Yhdysvalloissa. Linuxin versio 1.0 ilmestyi maaliskuussa 1994.

## 2 Ideologia ja lisenssit

Stallmanin perustaman FSF perusaate on ohjelmistojen vapaus. Koska englannin sana 'free' tarkoittaa myös ilmaista, termi 'free software' voi aiheuttaa sekaannusta. Ymmärtääkseni sanan merkityksen Stallman kehoittaa vertaamaan käsitteitä 'free speech' ja 'free beer'. Stallman mukaan vapaus käsittää neljä eri vapautta:

- Vapaus käyttää ohjelmaa, mihin tarkoitukseen hyvänsä (vapaus 0).
- Vapaus tutkia kuinka ohjelma toimii ja sovittaa se omiin tarpeisiin (vapaus 1). Esiehto tälle on lähdekoodin saatavuus.
- Vapaus levittää ohjelmaa auttaakseni lähimmäisiä (vapaus 2).

- Vapaus parantaa ohjelmaa ja julkaista parannukset koko yhteisön hyväksi (vapaus 3). Esiehto tälle on lähdekoodin saatavuus.

Näitä vapauksia suojatakseen vapailla ohjelmilla on lisensoijia, jotka takaavat ettei ohjelmaa uudelleen levitettäessä (joko alkuperäisessä muodossa tai muutettuna) sille voida määrätä käyttäjänoikeuksia rajoittavia ehtoja.

Vapaat ohjelmistot eivät siis välttämättä ole ilmaisia. Mutta ne voivat olla ilmaisia ja lisensoijien mukaisesti niitä voi muuttaa ja tuotosta levittää, esimerkiksi myymällä. Vapaa ohjelmisto ei myöskään tarkoita ei-kaupallista ohjelmistoa. Vapaat ohjelmat voivat olla kaupallisia ohjelmia, siis kaupallisessa käytössä, kehitystyössä ja levityksessä.

GPL lisensoijista on olemassa toinenkin versio LGPL (Lesser GPL). Alunperin lisensoijien nimi oli Library GPL, koska se kehitettiin GNU:n kirjastoja varten. Lesser GPL:n käyttö sallii sen lisensoijien alaisen koodin käytön ohjelmissa, jotka ovat suljettuja ohjelmistoja. Tämä voi tuntua koko vapaan lähdekoodin ideologian kanssa ristiriitaiselta, mutta Stallman perustelee LGPL:n käyttöä joissain kirjastoissa sillä, että jos kaikki kirjastot olisivat GPL:n alaisia se rajoittaisi koko GNU järjestelmän käyttöä.

Vuonna 1997 Eric Raymond julkaisi artikkelin ”The Cathedral and the Bazaar” [Ray97]. Artikkelin oli lähtölaulus koko vapaiden ohjelmistojen yhteisön jakautumiselle. Raymond jakaa artikkelissa ohjelmistokehityksen kahteen eri tapaan, katedraali- ja basaari-tapaan. Katedraalitalalla hän tarkoittaa perinteisempää ohjelmistotuotantometodia, jossa pieni joukko ohjelmoijia on vastuussa muutoksista. Ohjelmistoa rakennetaan ikäänkuin eristettynä muusta maailmasta ja versioita julkaistaan harvoin. Tästä seuraa Raymondin mukaan ilmiö, että virheet ja ongelmat ovat hankalia. Virheiden korjaaminen ja ongelmien ratkominen voi viedä kuukausia ja kun lopulta versio julkaistaan, se ei ole silti virheetön. Tämä aiheuttaa pettymyksiä käyttäjille. Basaaritapa on täysin päinvastainen. Siinä versioita julkaistaan usein, jolloin Raymondin mukaan virheet muuttuvat ”pinnalliksi”. Joku löytää virheen ja joku toinen saattaa korjata sen. Korjaus saattaa olla mukana hyvinkin nopeasti, koska versioi-

ta julkaistaan usein. Tämä puolestaan motivoi kehittäjiä, koska he näkevät oman työpanoksensa. Linuxin kehitys on ollut basaaritavan mukaista. Raymond huomasi tavan onnistuneeksi ja halusi kokeilla voiko tapaa soveltaa muihin projekteihin. Hän aloitti fetchmailin kehittämisen basaaritavan mukaisesti ja huomasi tavan toimivaksi. Fetchmail on ohjelma, joka hakee käyttäjän sähköpostit sähköpostipalvelimelta ja välittää ne paikalliselle sähköpostijärjestelmälle.

Raymondin artikkeli aiheutti suurta mielenkiintoa ja kohua vapaiden ohjelmien kehittäjien keskuudessa. Ensimmäisen kerran artikkeli oli julkisuudessa Linux-kongressissa 1997. Paikalla oli myös kustantaja Tim O'Reilly, (<http://www.oreilly.com/>) joka piti artikkelia loistavana ja halusi Raymondin pääpuhujaksi 1998 kesällä pidettävään Perl-konferenssiin. Tämä oli ensimmäinen kerta, kun monet vapaiden ohjelmien kehittäjistä tapasivat, vaikka olivat tehneet yhteisiä projekteja vuosia. O'Reilly innostui tapaamisesta ja halusi järjestää uuden tapaamisen, johon kutsuttaisiin kaikkien tunnettujen projektien johtajat kuten Linus Torvalds, Larry Wall (Perlin kehittäjä), Eric Allman (Sendmailin kehittäjä) ja Eric Raymond. Richard Stallman jätettiin kuitenkin kutsumatta, koska O'Reilly ja muut järjestäjät ajattelivat että hänen kanssaan ei päästäisi yhteisymmärykseen, jota ilman ei taas voitaisi jatkaa eteenpäin [Moo01].

Palo Altossa 1998 pidetyssä kokouksessa syntyi termi Open Source eli avoin lähdekoodi. Yhtenä syynä oli Netscapen ilmoitus julkistaa Communicator-selaimen lähdekoodi. Free software termi ei ollut saanut kannatusta kaupallisesti ja oli aika päästä eroon sen tuomasta imagosta uudella käsitteellä, joka olisi yksiselitteisempi termi samalle ideologialla. FSF:n ajatusmaailmalle avoin lähdekoodi on edellytys, mutta uuden termin kehittäjät halusivat painottaa juuri tätä, joten syntyi käsite Open Source. Myös muille lisensseille GPL:n lisäksi oli tarvetta ja ne eivät välttämättä olleet FSF:n ideologian mukaisia. Open Source ohjelmistojen määritelmä [OSD] saatiin Debianin Social Contractista [DSC]. Se sisältää samankaltaisia ehtoja kuin GPL ja käsittää muutakin kuin lähdekoodin avoimuuden kuten ohjelmien vapaan levittämisen ja muutosten teko-oikeuden. Open Source ohjelmistoja edistämään on perustettu Open Source Initiative [OSI]. FSF ei hyväksynyt Open Source termiä.

Heidän mielestään se ei ilmaise oleellisinta osaa eli vapautta ja saattaa aiheuttaa sekaannuksia.

### 3 Yhteenveto

Vapaat ohjelmistot ovat saavuttaneet vahvan jalansijan muiden ohjelmistojen joukossa. Tätä on edesauttanut Internetin nopea kasvu. Näin onkin luonnollista että Internetin ”ytimen” muodostavat vapaat ohjelmistot, kuten Apache [Apa99], Bind [Bind], Perl [Perl] ja sendmail [Sendm]. Apache on ollut suosituin webpalvelin huhtikuusta 1996 lähtien. Joulukuussa 2001 kaikista webpalvelimistä 56.5% oli Apache-palvelimia [NET]. Myös useat perinteisten kaupallisten suljettujen lähdekoodien ohjelmistojen valmistajat ovat reagoineet tilanteeseen ja vapauttaneet lähdekoodejaan.

## Viitteet

- Apa99 *The Apache Software Foundation*, <http://www.apache.org>, checked 6.2.2002.
- Bind Internet Software Consortium, *Berkeley Internet Name Domain*, <http://www.isc.org/products/BIND/>, checked 6.2.2002.
- Bison *Bison*, <http://www.gnu.org/software/bison/bison.html>, Last update 27.8.2001, checked 6.2.2002.
- Ray97 Raymond E., *The Cathedral and the Bazaar*, <http://www.tuxedo.org/~esr/writings/cathedral-bazaar/>, Last update 27.11.2001, checked 10.1.2002.
- DSC *Debian Social Contract*, [http://www.debian.org/social\\_contract.html](http://www.debian.org/social_contract.html), Last update 6.11.2001, checked 10.1.2002.
- Emacs *GNU Emacs*, <http://www.gnu.org/software/emacs/emacs.html>, Last update 1.1.2001, checked 6.2.2002.
- FSF Stallman R., *Free Software Foundation*, <http://www.gnu.org/fsf/fsf.html>, Last update 28.12.2001, checked 9.1.2002.
- GNU Stallman R., *The GNU Project*, <http://www.gnu.org/gnu/thegnuproject.html>, Last update 31.7.2001, checked 9.1.2002.
- NET *Netcraft Web Server Survey*, <http://www.netcraft.com/survey/>, checked 10.1.2002.
- Mach *The Mach Project Home Page*, <http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/mach/public/www/mach.html>, Last update 27.2.1997, checked 9.1.2002.
- Minix Tanenbaum A., *MINIX INFORMATION SHEET*, <http://www.cs.vu.nl/~ast/minix.html>, Last update 15.11.1996, checked 6.2.2002.

- Moo01 Moody G., *Rebel Code. The Inside Story of Linux and the Open Source Revolution*, Perseus Publishing, 2002.
- OSD *Open Source Definition*, <http://www.opensource.org/docs/definition.html>, checked 10.1.2002.
- OSI *Open Source Initiative*, <http://www.opensource.org/>, Last update 28.11.2001, checked 10.1.2002.
- Perl Wall L., *Larry Wall's Very Own Perl Page*, <http://www.wall.org/~larry/perl.html>, checked 6.2.2002.
- Sendm *Sendmail home page*, <http://www.sendmail.org/>, checked 6.2.2002.
- Joh79 Stephen C. Johnson. *YACC: Yet another compiler-compiler.*, Unix Programmer's Manual Vol 2b, 1979.

Muita linkkejä aihepiiristä:

<http://www.gnu.org/licenses/> (GNU lisenssit)

<http://history.perl.org/> (CPAST, Perlin historia)

<http://www.li.org/linuxhistory.php> (Linux History, Torvaldsin kirjoittama 1992)

<http://www.stallman.org/> (Richard Stallmanin sivut)

<http://www.wall.org/~larry/> (Larry Wallin sivut)

<http://www.ai.mit.edu/> (MIT Artificial Intelligence Laboratory)

<http://www.cs.vu.nl/~ast/> (Andrew S. Tanenbaumin sivut)

## Liite 1. Käytetyt lyhenteet

**AI** Artificial Intelligence

**ARPANET** Advanced Research Project Administration NETwork

**Bind** Berkeley Internet Name Domain

**Emacs** Editing MACroS

**GCC** GNU C Compiler

**GDB** The GNU Debugger

**GNU** GNU's Not Unix

**GPL** General Public License

**Hird** Hurd of Interfaces Representing Depth

**Hurd** Hird of Unix-Replacing Daemons

**ITS** Incombatible Time-Sharing System

**LISP** List Processing

**LGPL** Lesser General Public License

**MIT** Massachussets Institute of Technology

**Perl** Practical Extraction and Report Language