

hyväksymispäivä arvosana

arvostelija

**Unixin historia:
avaruuspelistä moderniksi käyttöjärjestelmäksi**

Teemu Kemppainen

Helsinki 18.2.2007

Tietojenkäsittelytieteen historia -seminaari

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Sisältö

1 Johdanto	1
2 Syntymä	1
2.1 Edeltäjänä Multics-projekti	1
2.2 Space Travel -peli	2
2.3 Unics syntyy	3
3 Alkuvuodet	3
3.1 PDP-11: ensimmäinen kunnon Unix-kone	4
3.2 C-kieli tekee Unixista siirrettävän	5
3.3 Unixin julkistaminen: artikkeli ACM:ssä 1974	5
3.4 Unixin tekniset innovaatiot	6
4 Maailmanvalloitus	6
4.1 Innokas kehittäjäyhteisö	7
4.2 Markkinoiden pirstoutuminen	7
5 Filosofia: mikä Unixissa kiehtoo?	9
6 Tiivistelmä	10
Lähteet	10

1 Johdanto

Unixia ei missään tapauksessa saisi käyttää esimerkkitapauksena ohjelmistotuotannon kurssilla: ohjelmiston historiassa on enemmänkin edustettuna kaaos, sattuma ja kokeiluhenki kuin kurinalainen, suunnitelmallinen ohjelmistotuotantoprosessi. Tästä huolimatta Unix on kohta 40 vuoden ikäisenä eräs maailman pitkäikäisimmistä yhä käytössä olevista tietokoneohjelmista, joka on ratkaisevasti vaikuttanut paitsi tämän päivän käyttöjärjestelmiin, niin myös Internetin kehitykseen, ohjelmistotekniikan ajatustapoihin sekä avoimen lähdekoodin kulttuuriin. Tässä kirjoituksessa esitetään järjestelmän historia, joka on samalla eräs tietojenkäsittelyn kuuluisimpia tarinoita.

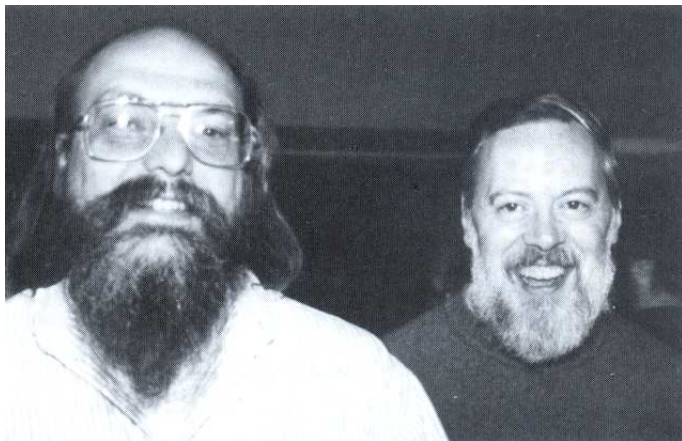
2 Syntyminen

Unix syntyi puhelinyhtiö AT&T:n Bell Labsin laboratorioissa Murray Hillsissa Yhdysvaltain New Jerseyssä vuonna 1969. Tässä luvussa käsitellään järjestelmän syntymää.

2.1 Edeltäjänä Multics-projekti

Massachusetts Institute of Technology (MIT), General Electric ja Bell Labs koettivat 1960-luvulla yhteistuumin rakentaa monen käyttäjän käyttöjärjestelmää, jossa olisi muun muassa virtuaalimuisti ja tuki pääteyhteyksille [Sal95, sivu 26–27]. Tämä projekti oli nimeltään Multiplexed Information and Computing Service eli Multics.

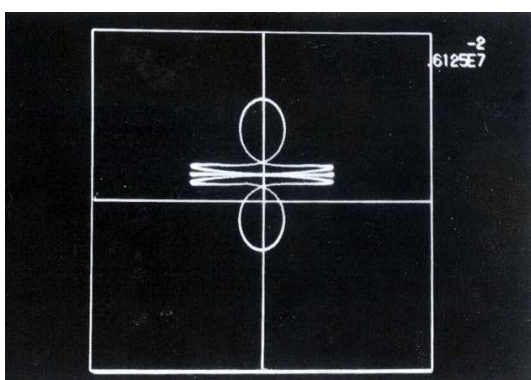
Bell Labsin riveissä tässä hankkeessa olivat mukana myös **Ken Thompson**, **Dennis Ritchie** (kuva 1), sekä **Doug McIlroy**, **Joseph ”Joe” Ossanna**, ja **Bob Morris** [Sal95, sivu 6]. Multics-hanke ajautui kuitenkin vaikeuksiin koska yhteistyötahoilla oli liikaa erimielisyyksiä päämääristä. Toukokuussa 1969 Bell Labs irtisanoutui hankkeesta. Thompson, Ritchie ja muut Bell Labsilla Multicsin parissa työskennelleet ihmiset olisivat kuitenkin halunneet jatkaa käyttöjärjestelmien tutkimista, mutta heillä ei ollut käytössään tarvittavaa laitteistoa ja hakemukset tietokoneen hankkimiseksi evättiin toistuvasti [Bel02]: tuohon aikaan tietokoneet maksoivat satojatuhansia dollareita.



Kuva 1: Ken Thompson (vas.) ja Dennis Ritchie, voimakaksikko Unixin takana. [Sal95]

2.2 Space Travel -peli

Bell Labsin käyttöjärjestelmätutkimuksen tulevaisuus näytti siis synkältä. Mutta sattumalta eräs Bell Labsin työntekijä siirtyi toiseen toimipaikkaan, minkä ansiosta talossa vapautui Digital PDP-7 -tietokone. Ken Thompsonin onnistui iskeä käteensä siihen ensimmäisenä. Tämä tapahtui alkuvuodesta 1969. Saman vuoden heinäkuussa ihminen laskeutui ensimmäistä kertaa kuun kamaralle. Ken Thompson alkoi avaruusmatkailun innoittamana rakennella PDP-7-koneeseen avaruuspelejä nimeltä Space Travel (kuva 2). Aiemmin hän oli kehittänyt tuota peliä keskuskoneella, jossa yhden pelin pelaamisen kerrotaan maksaneen 75 dollaria [Rit84]: tuohon aikaan maksettiin prosessoriajasta.



Kuva 2: Tästä se alkoi: jonkinlainen kiertorata Space Travel -pelistä, jota ohjelmoijessaan Ken Thompson oppi PDP-7:n koodaamisen. Vuosi on 1969. [Sal95]

2.3 Unics synty

Vaikkei kehittäjillä ollut käytössään omaa tietokonetta käyttöjärjestelmätutkimukseen, niin uusia ideoita palloiteltiin tässä porukassa paljon. Keväällä 1969, pahimman masennuksen Multics-hankkeen jäljiltä jo kaikottua, Bell Labsin kahvihuoneessa alettiin pyöritellä ideaa uudesta tiedostojärjestelmästä. Sitä ideoitiin liitutauluille ja sanelujärjestelmään, ja näistä keskusteluista on peräisin esimerkiksi käyttöjärjestelmäkursseilta tuttu *inode*-käsite. Tiedoston inode sisältää metatietoa tiedostoista, mm. osoitteet fyysisiin levylohkoihin joille tiedosto on tallennettu. Koodaajavelho Ken Thompson innostui kahvihuonespekseistä niin kovasti, että toteutti tiedostojärjestelmän PDP-7:llä, jonka ohjelmoinnin hän oli oppinut Space Travelia kirjoitellessaan. Tiedostojärjestelmä tarvitsi tuekseen lisäksi erilaisia tekstiskriptejä, joten projekti laajeni.

Myöhemmin kesällä Thompsonin vaimo matkusti pariskunnan vastasyntyneen pojan kanssa Kaliforniaan vanhempiansa luokse kuukaudeksi. Thompson käytti nuo neljä viikkoa järjestelmällisesti hyväkseen, ja ohjelmoi tiedostojärjestelmän päälle käyttöjärjestelmän ytimen, komentotulkin, tekstieditorin sekä assemblerin, eli ohjelman joka muuntaa symbolisen konekielen suoritettavaksi konekoodiksi.

Näin syntyneeseen uuteen käyttöjärjestelmään omaksuttiin monia piirteitä Multicista, kuten hierarkkinen hakemistorakenne sekä komentorivitulkki ("shell"). Thompsonin koodaussuoritusta voidaan pitää poikkeuksellisen lahjakkuuden osoituksena. Ennen assemblerin valmistumista nimittäin kaikki koodi oli kirjoitettava toisella järjestelmällä, josta se siirrettiin kömpelöillä paperinauhoilla PDP-7-koneeseen. Tähän aikaan koneita käytettiin mekaanisten kirjoituskoneiden kaltaisilla terminaaleilla, joissa hädin tuskin oli rivieditorin toiminnallisuutta. Pienenkin näppäilyvirheen korjaaminen koodin keskeltä vaati huomattavia ponnisteluja.

Multicsin pohjalta syntyneelle järjestelmälle väännettiin nimeksi Unics, joka myöhemmin asettui muotoon Unix. Tavaramerkiksi rekisteröityä muotoa UNIX käytetään myös paljon.

3 Alkuvuodet

Unixin alkuaikoina Bell Labsin rahakirstun vartijat eivät olleet alkuunkaan kiinnostuneita käyttöjärjestelmätutkimuksesta, eikä Unix-kehittäjien tarvitseman tietokoneen hankkimiseen myönnetty varoja.

3.1 PDP-11: ensimmäinen kunnan Unix-kone

Kun Unixin kehityksessä nyt oli päästy vauhtiin, alkoivat kuitenkin jo vanhan PDP-7:n rajoitukset tulla vastaan: tarvittiin ehdottomasti uutta laitteistoa toteuttamaan kaikki kehittäjien innovaatiot. Koska aiemmat hakemukset tietokoneen hankkimisesta käyttöjärjestelmätutkimukseen olivat epäonnistuneet, turvautuivat Bell Labsilaiset nyt vippaskonsteihin.

Markkinoille oli juuri ilmestynyt kohtuuhintainen (\$65 000) PDP-11-tietokone, jollaista he himoitsivat. Hankintakulut perusteltiin johtoportaalle tällä kertaa sillä verukkeella, että koneella kehitettäisiin tekstinkäsittelyjärjestelmä. Käyttöjärjestelmistä tai Unixista oltiin nyt viisaasti täysin vai. Tämä perustelu meni kuin menikin johtoportaassa läpi, ja pian Bell Labsiin saatiin moderni PDP-11. Kuvassa 3 Ritchie ja Thompson työskentelevät uuden koneensa kimpussa. Tekstinkäsittelyjärjestelmän ensimmäinen asiakas oli Bell Labsin patenttiosasto.



Kuva 3: Dennis Ritchie vilkuilee olan yli, kun Ken Thompson koodaa PDP-11-myllyllä. Tällä koneella kehiteltiin 1970-luvun alussa mm. C-kieli sekä ensimmäinen ”oikea” Unix. [Www02]

3.2 C-kieli tekee Unixista siirrettävän

Ensimmäinen Unix oli kuten melkein kaikki ohjelmat tuohon aikaan kirjoitettu suoraan konekoodilla. Tässä yhteydessä Ken Thompson oli taas erittäin tarkkanäköinen ja aikaansa edellä, ja hän halusi siirtyä korkeamman tason kieleen. Tarjonta tähän aikaan oli kuitenkin vähäistä. Fortran ei kelvannut Thompsonille, joten edessä oli – taas kerran – oman systeemin kehittäminen.

Thompson laati oman kielen nimeltä B. Se oli – nykyisen Javan tapaan – tulkkauva kieli, jonka suurin ongelma oli hitaus. Kun sitten Dennis Ritchie paranteli B:tä lisäten siihen muuttujatyypit sekä kääntäjän, syntyi C-kieli.

Kun C:hen kesällä 1973 vielä lisättiin taulukot sekä muut perustietorakenteet ja parneltiin kääntäjää, kieli oli jo niin voimakas että koko Unix-järjestelmä kirjoitettiin uudestaan kokonaan C:llä. Oli tähän aikaan täysin ennenkuulumatonta, että käyttöjärjestelmä oli kirjoitettu korkean tason kielellä [Sta05, sivu 91]. Tämä Thompsonin ja Ritchien intuitioon perustunut ratkaisu oli erittäin merkittävä Unixille ja itse asiassa koko tietojenkäsittelyn alalle. Vielä tänäkin päivänä, melkein 40 vuotta myöhemmin, C on mitä ahkerimmassa käytössä kaikessa laiteläheisessä ohjelmoinnissa, ja sen piirteitä on paljon sekä C++:ssa että Javassa. Esimerkiksi se ohjelmointikielten piirre, että oheislaitteita kohdellaan kuten levyllä olevia tiedostoja, on mitä ilmeisimmin Ritchien ja C-kielen peruja.

Kun Unix nyt oli tehty C:llä, eikä enää suoraan PDP-11:n konekielellä, järjestelmä olikin yhtäkkiä siirrettävä. Sen pystyi asentamaan periaatteessa mihin vain koneeseen, jossa vain oli C-kääntäjä. Tämä piirre teki Unixista ainutlaatuisen, sillä tähän asti tietokonevalmistajat olivat myyneet laitteiston sekä käyttöjärjestelmän aina yhdessä.

3.3 Unixin julkistaminen: artikkeli ACM:ssä 1974

Viimeistään tässä vaiheessa kaikille Unixin parissa työskennelleille alkoi olla selvää, että heillä oli käsissään jotakin merkittävää. Vuoden 1973 lokakuussa, kun ACM järjesti käyttöjärjestelmäseminaarin jossa Ken Thompson piti esitelmän Unixista, järjestelmä oli käytössä 16 paikassa [Sal95, sivu 54]. Puheen jälkeen asennusten määrä nopeasti tuplaantui, mutta varsinainen mylläkkä alkoi kun Communications of the ACM -julkaisussa seuraavana vuonna ilmestyi artikkeli ”The Unix Time Sharing System”. Sen myötä Unix-kärpänen alkoi levitä ympäri maailman.

3.4 Unixin tekniset innovaatiot

Unixista teki niin kiinnostavan sen sisältämät uudet tekniikat. Nyt julkituotu Unix toimi PDP-11/45-tietokoneessa jossa oli kaikkiaan 144 kilotavua muistia, josta Unix vei 42 kilotavua [RiT74]. Kuitenkin jo näin pieneen tilaan mahtunut järjestelmä sisälsi monia tuohon aikaan mullistavia piirteitä:

- Prosessit ja lapsiprosessit
- Putket ja suodattimet (`"ls | sort | more"`)
- Tiedostojärjestelmä ja hierarkkinen hakemistorakenne
- Liitettävien levy-yksiköiden tuki (`"mount"`-komento)
- Moniajo
- Monen käyttäjän tuki
- Tiedoston suojaukset (`"chmod"`-komento)
- Komentotulkki shell
- C-kielinen lähdekoodi

Nämä piirteet olivat siis vakiona Unixissa jo vuonna 1974. Listan ominaisuudet sisältyvät – muodossa tai toisessa – jokaiseen nykyaikaiseen käyttöjärjestelmään. Sillä erotuksella, että suljettujen järjestelmien (kuten Windowsit) mukana ei tule lähdekoodia, ja nykyisissä järjestelmissä (Linux, MacOS, Windows) komentotulkkiä on täydennetty graafisella käyttöliittymällä, sekä lisätty järjestelmään verkko-ominaisuudet, eivät käyttöjärjestelmien perusratkaisut ole viimeisten vuosikymmenien aikana juurikaan muuttuneet. Näin voidaan katsoa, että Unix loi koko nykyaikaisen käyttöjärjestelmän käsitteen.

4 Maailmanvalloitus

"No advertising, no support, no bug fixes, payment in advance" – tällä tavoin kuvattiin Bell Labsin alkuperäistä Unix-politiikkaa [Sal95, sivu 59]. AT&T, jonka osa Bell Labs oli, oli nimittäin USA:n antitrust-viranomaisten toimesta vuonna 1949

kielletty toimimasta muussa kuin puhelinbisneksessä. Tästä syystä Bell Labs ei viranomaistoimien pelossa uskaltanut tehdä rahaa Unixilla. Näin ensimmäiset Unix-lisenssit myytiin nimelliseen hintaan joillakin tuhansilla dollareilla ei-kaupallisille tahoille kuten yliopistoille, ja koko lähdekoodi seurasi aina mukana. Näkyvää mainontaa ei harrastettu eikä Unixille annettu käyttäjätukea. Koko Unix-toiminta oli jokseenkin vaiettua.

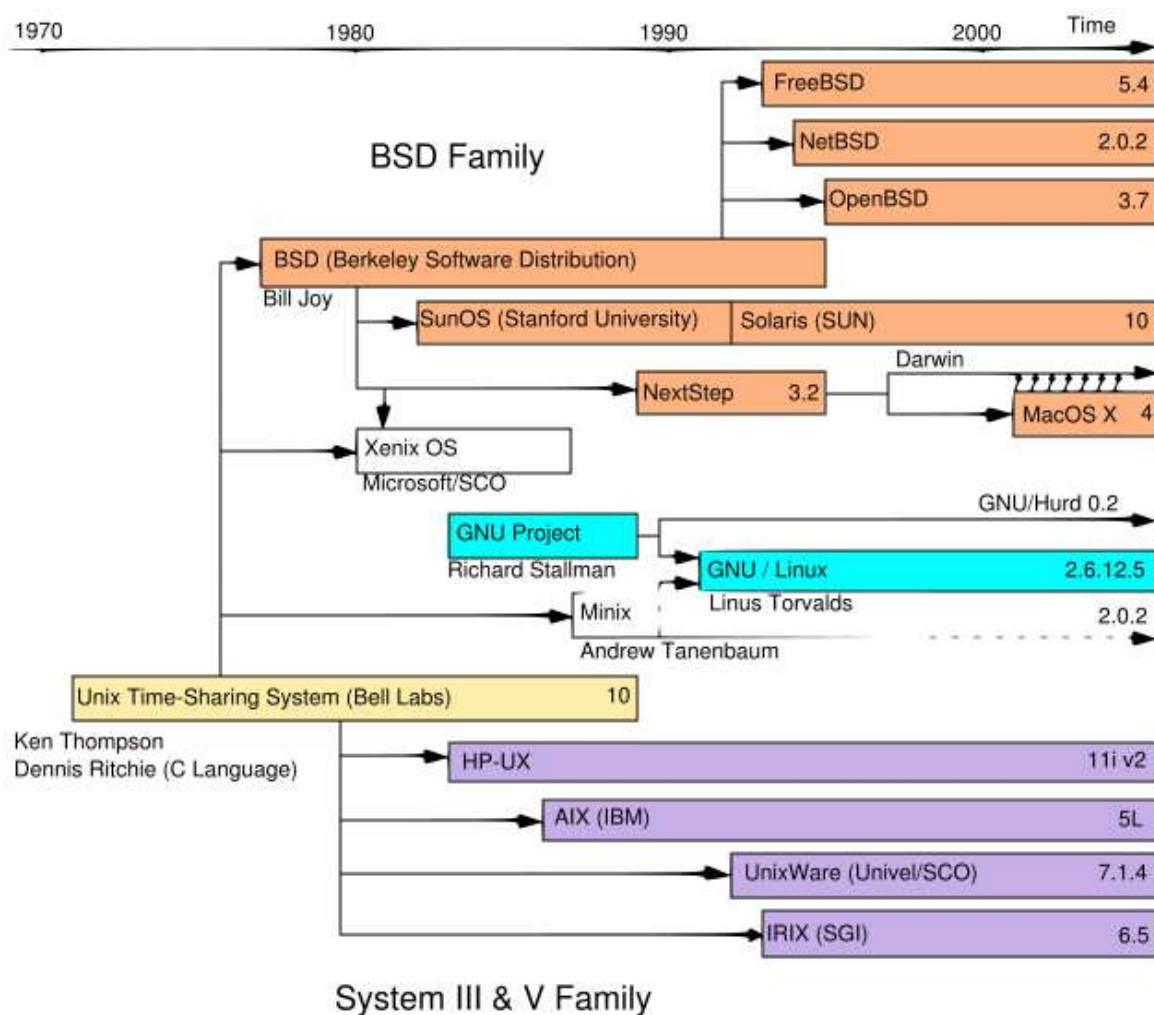
4.1 Innokas kehittäjäyhteisö

Epäsuorasti saamme kiittää USA:n kilpailuviranomaisia siitä, että Unix lähti valloittamaan maailmaa. Nimittäin nyt jokainen taho, joka osti Unix-lisenssin, saattoi parannella ja kehittää lähdekoodia parhaan kykynsä mukaan. Virallisen käyttäjätuen ja bugikorjausten puuttuessa Unixin ympärille kehittyi voimakas alakulttuuri ja ympäri maailman – niin Japanissa, Australiassa, Isossa-Britanniassa kuin Hollannissa, myöhemmin myös Suomessa (1984) – syntyi UUG-ryhmiä (Unix User Group) jotka kehittivät Unixia, antoivat käyttäjätukea ja ennen kaikkea keräsivät Unixista kiinnostuneita ihmisiä yhteen. Tuohon aikaan ei vielä puhuttu nöртеistä, vaan tietokoneharrastajia kutsuttiin hakkereiksi.

4.2 Markkinoiden pirstoutuminen

Seurauksena oli, että Unix jakaantui useaksi eri murteeksi (kuva 4). Berkeley Unix syntyi Ken Thompsonin luennoissa 1976-1977 Berkeleyyn yliopistossa, minne hän luonnollisesti vei myös Unixin. Hänen lähdettyään Unixin kehittäminen jatkui **Bill Joy**-nimisen miehen johdolla BSD Unixina (Berkeley Software Distribution). Pian Berkeleyyn asiakkaaksi tuli Internetin perustan luonut DARPA (Defence Advanced Research Projects Agency), USA:n maanpuolustusteknologiaa kehittävä organisaatio. Internetin alkuaikoina käytännössä kaikki siihen liitetyt tietokoneet olivat Unix-järjestelmiä.

Tällä välin juristit olivat tehneet töitään ja vuonna 1984 AT&T sai alkaa tehdä bisnestä myös Unixilla. Tässä vaiheessa erilaisia merkittäviä Unix-murteita oli markkinoilla jo puolenkymmentä, ja kaikki kilpailivat keskenään.



Kuva 4: Unixin voima ja heikkous on sen jakaantuminen moneksi eri haaraksi. [Wik06]

Yritykset Unixin standardoinnista, kuten POSIX-hanke (Portable Operating System Interface for uniX), eivät onnistuneet luomaan yhtenäistä ja vahvaa Unixia. Niinpä muut käyttöjärjestelmät, etenkin Microsoftin DOS sekä eri Windowsit, rohusivat itselleen nopeasti kasvaneet mikrotietokonemarkkinat 1980 ja 1990 -luvuilla. Sovellusohjelmien ja laitteiden valmistajien oli kannattavampaa suuntautua isoon ja yhtenäiseen Microsoft-maailmaan kuin hajanaiseen Unix-ympäristöön.

”Alkuperäisen” AT&T Bell Labsin Unixin viimeinen versio 10 ilmestyi vuonna 1989 [Bel02]. Vuonna 1991, maailmantalouden laman kynnyksellä, AT&T möi Unixliiketoimintansa, mihin käytännössä loppui alkuperäisen Unixin tarina. Tänä päivänä Unix elää näkyvimmin Linux-maailmassa, mutta harva tietää, että helppokäyttöisenä pidetyn Applen MacOS X -käyttöjärjestelmän pohjalla on BSD Unix. Yritys-

järjestelmien puolella suosittuja Unix-versioita ovat esimerkiksi IBM:n AIX, HP:n HP-UX ja Sunin Solaris. Sun-yhtiön tarina käynnistyi aikoinaan Stanfordin yliopistosta.

5 Filosofia: mikä Unixissa kiehtoo?

Unix- ja open source -kulttuurissa on edelleen voimassa paljon siitä hengestä, joka vallitsi Bell Labsilla 1960-luvun lopussa. Tutkijoilla oli tuolloin vapaus tehdä mitä haluavat, eikä heiltä varsinaisesti vaadittu yhtään mitään. He loivat järjestelmän, jonka käytöstä he pitivät. Joskus tällainen asenne tuottaa hedelmää, ja Unix on siitä hyvä esimerkki. ”You should be writing programs that you want to use”, tiivistää Unix-aktiivi **Eric Allman** [Sal95, sivu 230].

Unix-järjestelmistä tekee insinööriluonteelle kiehtovia niiden haasteellisuus. Monimutkaisen, tavalliselle tietokoneen käyttäjälle käsittämättömän Unix-järjestelmän opettelu ja ymmärtäminen on palkitsevaa. Lähdekoodin ja dokumentaation ollessa avointa Unixin voi todella *ymmärtää*, toisin kuin suljetun järjestelmän, jonka toiminnasta voi muodostaa mielikuvan vain sen käyttöliittymän perusteella. Avoimen lähdekoodin kulttuuriin tottunut haluaa avata auton konepellin ja tutkia moottoria; suljettu järjestelmä on kuin auto, jonka konepelti on hitsattu kiinni. Samalla puolet auton tarjoamasta hauskuudesta on poissa.

Koko Unix-filosofia on kiteytetty seuraavasti [Sal95, sivu 53]:

- Write programs that do one thing and do it well
- Write programs to work together
- Write programs that handle text streams, because that is a universal interface

Unix-systeemit koostuvat tuhansista pienistä apuohjelmista (”utilities”) jotka toteuttavat ylläolevaa. Mikä muu piirre Unixissa sitten vetosi hakkereihin ja loi niin massiivisen kehittäjäyhteisön? Eräs syy saattaa olla Unixiin liittyvä älyllinen rehellisyys ja avoimuus. Esimerkiksi Unix-järjestelmien **man**-sivut (ohjelmien opastetekstit) ovat aikojen alusta sisältäneet Bugs-osion, jossa kerrotaan avoimesti ohjelmien sisältämät korjaamattomat bugit. Tällainen toiminta myös houkuttelee osaavia koodaajia mukaan yhteisöön: korjaanpa tuon, ja laitan ratkaisuni jakeluun.

6 Tiivistelmä

Tämän kirjoituksen johdannossa sanotaan, että Unixin historiassa on edustettuna kaaos, sattuma ja kokeiluhenki. Kaoottisuus ilmenee siinä, miten epäonnistuneen Multics-projektin raunioille rakentui ilman minkäänlaista pitkän tähtäimen suunnitelmaa näin tärkeä käyttöjärjestelmä. Sattuman vaikutus Unixin syntyyn korostuu esimerkiksi siinä, miten Ken Thompson sai käsiinsä ylimääräiseksi jääneen PDP-7-tietokoneen, jolla rakennettiin ensimmäinen Unix-versio. Ja sattuman ansiota on sekin, että Bell Labs, emoyhtiön antitrust-sotkujen vuoksi, lisensoi Unixia tavalla joka salli järjestelmän jatkokehittelyn. Kokeiluhenkeä edustaa taas Unixin ympärille syntynyt voimakas käyttäjäyhteisö. Alun perin tutkijoiden toisille tutkijoille tekemän käyttöjärjestelmän innovaatiot näkyvät käytännössä jokaisessa tämän päivän käyttöjärjestelmässä, eikä näköpiirissä ole mitään mikä syrjäyttäisi Unixin ajattelumallit.

Lähteet

- Bel02 Bell Labs, The creation of the UNIX operating system, 2002. <http://www.bell-labs.com/history/unix/>. [27.1.2007]
- Rit84 Ritchie, D. M., The evolution of the Unix time-sharing system. *AT&T Bell Laboratories Technical Journal*, 63,6 part 2(1984), sivut 1577–1593.
- RiT74 Ritchie, D. M. ja Thompson, K., The UNIX time-sharing system. *Commun. ACM*, 17,7(1974), sivut 365–375.
- Sal95 Salus, P. H., *A Quarter Century of Unix*. Addison-Wesley Publishing Company, Massachusetts, USA, 1995.
- Sta05 Stallings, W., *Operating Systems: Internals and Design Principles*. Pearson Prentice Hall, Upper Saddle River, NJ, USA, 2005.
- Wik06 Wikipedian Unix-artikkeli, 2006. <http://en.wikipedia.org/wiki/Unix>. [11.2.2007]
- Www02 <http://www.psych.usyd.edu.au/pdp-11/Images/ken-den.jpeg> [11.2.2007]