

Samuli Niemi

LINUX FROM SCRATCH

GNU/Linux-järjestelmän kokoaminen lähdekoodeista

LINUX FROM SCRATCH

GNU/Linux-järjestelmän kokoaminen lähdekoodeista

Samuli Niemi
Opinnäytetyö
Kevät 2015
Tietojenkäsittelyn koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma, Järjestelmäasiantuntemuksen sv

Tekijä: Samuli Niemi

Opinnäytetyön nimi: Linux from Scratch

Työn ohjaaja: Jouni Juntunen

Työn valmistumislukukausi- ja vuosi: Kevät 2015

Sivumäärä: 47 + 1

Tässä opinnäytetyössä käsitellään Linux from scratch -nimistä projektia, jossa rakennetaan GNU/Linux-järjestelmä täysin lähdekoodeista. GNU/Linux:in rakentaminen lähdekoodeista antaa käyttäjälle vapauden luoda sellainen GNU/Linux-järjestelmä kuin hän itse haluaa ja opettaa käyttäjälle GNU/Linux-järjestelmän perusasiat. Tällä opinnäytetyöllä ei ole tilaajaa vaan tekijä tekee sen itseään varten. Opinnäytetyön tekijä on hyvin kiinnostunut aiheesta ja odottaa sen laajentavan hänen käsitystään GNU/Linux-järjestelmistä.

Käytännön tehtävänä rakennetaan reitittimeksi ja palomuuriksi räätälöity GNU/Linux-järjestelmä. Järjestelmä kehitetään virtualisoidussa ympäristössä käytännön syistä. Opinnäytetyön raportti ei sisällä yksityiskohtaisia ohjeita miten järjestelmä tehdään, vaan viittaa asiakirjoihin ja ohjeisiin mitä käytettiin.

Teoriaosuudessa käsitellään GNU/Linuxin ja Unixin historiaa ja GNU/Linuxin pääkomponentteja kuten ydintä, tiedostojärjestelmiä, muistin käsittelyä ja X-ikkunointijärjestelmää. Opinnäytetyö selittää myös GNU:n tarkoituksen Linuxin nimessä ja mitä vapaat ohjelmistot ovat. Opinnäytetyön tietopohja on koottu parista kirjasta ja useasta eri lähteestä Internetistä. LFS-projektin pohjalta voi luoda käytännössä oman jakelun, jonka takia opinnäytetyössä esitellään yleisimmät jakelut, missä niitä käytetään ja mitä ovat sulautetut järjestelmät. Opinnäytetyössä käsitellään myös miten GNU/Linux-järjestelmän rakentamista teoriassa.

Käytännön tehtävänä luotu Linux-reititin voidaan asentaa mille tahansa x86-64-pohjaiselle tietokoneelle, jossa on vähintään kaksi verkkokorttia. Sillä voidaan tarjota Internet-yhteys, DNS- ja DHCP-palvelut ja monipuolinen palomuuuri. Linux-reititintä voidaan tulevaisuudessa kehittää lisäämällä siihen web-pohjainen käyttöliittymä, ohjelmia seuraamaan verkkoliikenteen käyttöä ja tieturvaohjelmia. Opinnäytetyössä Linux-reititin kehitettiin virtuaaliseen ympäristöön, jonka takia seuraava looginen askel olisi asentaa järjestelmä oikean laitteiston päälle, mikä onnistuisi ilman merkittäviä muutoksia.

Opinnäytetyön tekijä oppi paljon uutta oleellista tietoa ja löysi uusia hyödyllisiä ominaisuuksia GNU/Linux-järjestelmistä. LFS-projekti mahdollistaa hyvin räätälöityjen Linux-järjestelmien luonnin, mutta se vaatii merkittävästi enemmän aikaa ja vaivaa kuin valmista jakelua käyttämällä. Linux-järjestelmä tulisi kehittää lähdekoodeista asti vain jos sen tarkoitus on palvella jotain hyvin ainutlaatuisista tarkoitusta tai kohdelaitetta. Muissa tapauksissa olemassa olevaa jakelua kannattaa käyttää ensisijaisesti.

Asiasanat: Linux, jakelu, reititin, sulautettu järjestelmä, käyttöjärjestelmä

ABSTRACT

Oulu University of Applied Sciences

Degree programme in business information systems, option of computer systems expertise

Author: Samuli Niemi

Title of thesis: Linux from Scratch

Supervisor(s): Jouni Juntunen

Term and year when the thesis was submitted: Spring 2015 Number of pages: 47 + 1

The goal of this thesis is to study a project called Linux from Scratch, which provides the users with step-by-step instructions on how to build their own custom Linux system, entirely from source code. Building a GNU/Linux-system entirely from source allows the user to fully customize the system to fit the user's needs and teaches the user the basics of GNU/Linux-system. This thesis does not have a client. The author of this thesis is highly interested in the subject and expects it to broaden his understanding of GNU/Linux systems.

A fully functional customized GNU/Linux system was built as a practical assignment. The GNU/Linux system was then configured to work as a router and firewall. The system was developed in a virtualized environment for practical reasons. This thesis does not include step-by-step instructions on how the system was built. Instead, it refers to the necessary documents and books used and how they were adapted.

The theoretical background of the thesis includes general history about the GNU/Linux and UNIX, as well as information about the main GNU/Linux components such as the kernel, file systems, memory handling and the X Window System. The thesis also explains the meaning of GNU in the Linux name and what the free software means. The information in the thesis was gathered from a few literary sources and various sources on the Internet. Since the LFS-project is about creating your own Linux distribution, this thesis also introduces the most common Linux distributions: what makes them unique, what are the main components of a distribution and what are embedded systems. The thesis also discusses how the Linux system is built in theory.

The Linux router that was created as practical work in this thesis can be installed on any x86-64-based computer with at least two network interfaces. It can be used to provide Internet access with a powerful firewall. The Linux router can be further developed to include a web-based graphical interface, applications to track network usage and security related applications. In the thesis, the Linux router was developed on a virtual machine. The next logical step is to implement it on real hardware, which does not require significant changes to the system.

In the thesis the author discovered useful new properties in the GNU/Linux. The LFS-project allows the user to create a highly customized Linux system. However, it requires significantly more time and effort to build than installing an existing distribution. In conclusion, a custom Linux system should be built only when it is needed for some unique purpose or when the target hardware has very limited resources. Otherwise an existing distribution should be used.

Keywords: Linux, distribution, router, operating system, embedded system

SISÄLLYS

1	JOHDANTO.....	7
2	TAVOITTEET JA KEHITYSTEHTÄVÄT.....	8
3	GNU/LINUX.....	10
3.1	Unix ja Linuxin historia.....	10
3.2	GNU/Linuxin ydin.....	11
3.3	Muistinhallinta.....	12
3.4	Tiedostojärjestelmät.....	13
3.5	X Window System.....	15
3.6	GNU Compiler Collection.....	17
3.7	Vapaat ohjelmistot.....	17
3.8	Linux-jakelut.....	18
3.8.1	Jakelu.....	18
3.8.2	Debian.....	19
3.8.3	Arch Linux.....	20
3.8.4	Gentoo.....	21
3.8.5	Fedora.....	21
3.9	Sulautetut Linux-järjestelmät.....	22
4	LINUX-JÄRJESTELMÄN KEHITTÄMINEN.....	25
4.1	Linux From Scratch.....	25
4.2	Edellytykset.....	25
4.3	Pakettienhallinta.....	26
4.4	Laittehallinta.....	26
4.5	LFS-järjestelmän kokoaminen.....	27
4.5.1	Kehitysympäristön valmistelu.....	27
4.5.2	Chroot.....	28
4.5.3	Pakettien kääntäminen.....	28
4.5.4	Järjestelmän konfigurointi.....	29
4.5.5	Linuxin ydin ja käynnistyslataaja.....	30
5	LINUX-JÄRJESTELMÄN KOKOAMINEN.....	31
5.1	Kehitysympäristö.....	31

5.2	LFS Järjestelmän rakentaminen	31
5.2.1	Pakettien kääntäminen	31
5.2.2	Käynnistyslataaja	32
5.2.3	Linuxin ytimen kääntäminen	33
5.3	Reitittimen kehittäminen	34
5.3.1	Tarvittavat ohjelmistot.....	34
5.3.2	Verkkoadapterien konfigurointi	34
5.3.3	Dnsmasq.....	35
5.3.4	UFW.....	37
6	POHDINTA	40
	LÄHTEET	41

1 JOHDANTO

Nykypäivän ihminen ei voi väistää GNU/Linuxin törmäämistä. Hyvin monella se voi olla jopa niin lähellä kuin omassa taskussa tai kädessä. Joillakin voi olla useita Linuxeja; olohuoneessa, autossa tai jopa keittiössäkin. Internetiä selatessa näkemämme tieto on lähes väistämättä kulkenut jonkin Linuxin kautta. Linux on kaikkialla jokapäiväisessä elämässämme, koska Linux on erittäin suosittu käyttöjärjestelmä sulautetuissa järjestelmissä ja palvelimissa. Se löytyy Android-puhelimiemme sisältä, älytelevisioista, verkkokiintolevyistä, autojen älyominaisuuksien takaa, twiittaavasta älyvessasta ja yleensäkin hyvin monesta laitteesta jossa mainitaan termi "äly". 95,2 % supertietokoneista käyttää GNU/Linuxia ja 67,6 % Internetin verkkosivuista käyttää Unixia, johon lasketaan Linuxit mukaan (Top500 2014, viitattu 16.4.2015; W3Techs 2015, viitattu 16.4.2015). Vaikka emme osaisi käyttää Linuxia tai tietäisi siitä mitään, hyödyimme siitä tiedostamattakin.

Kiinnostuin Linux-käyttöjärjestelmästä jo monta vuotta ennen kuin aloitin tietojenkäsittelyn opintoni. Kokeilin erilaisia vapaassa jaossa olleita Linux-jakeluja ja niiden ominaisuuksia ensimmäiseksi tavanomaisessa käytössä kuten Internet-selailussa, videoiden ja kuvien katselussa ja joidenkin pelien pelaamisessa. Tuohon aikaan noiden asioiden tekeminen Linuxilla ei ollut itsestään selvää, koska useimmat laitevalmistajat eivät tukeneet Linuxia virallisesti, mikä tuotti ongelmia. Ongelmatilanteissa piti hakea apua toisilta käyttäjiltä, lukea useita ohjeita ja toteuttaa epävirallisia ratkaisuja. Ongelmat järjestelmän kanssa sytyttivät kiinnostukseni sitä kohtaan. Koin ymmärtäväni sen kautta enemmän tietokoneen toiminnasta ja pääseväni lähemmäksi sen toimintaperiaatteita kuin Windowsin kanssa.

Ajan kuluessa opin hyödyntämään Linuxia myös palvelimissa ja tietokoneisiin liittyvien harrastuksien parissa. Sitä kautta opin käyttämään Linuxia, mutta en oikeastaan koskaan oppinut ymmärtämään syvällisesti Linuxia ja sitä, miten se toimii.

Linux from scratch – projektin avulla voin päästä syvemmälle Linuxin sisälle ja oppia ymmärtämään sen toimintaa ja miten sitä voi hyödyntää entistä paremmin erilaisissa laitteissa tai projekteissa. Samalla kehitän ongelmanratkontakykyjäni ja vahvistan osaamistani uusien asioiden opettelua varten.

2 TAVOITTEET JA KEHITYSTEHTÄVÄT

Linux from Scratch (LFS) on projekti, joka tarjoaa ohjeet joilla käyttäjä voi rakentaa oman Linux-järjestelmän täysin lähdekoodista. LFS:n rakentamisen avulla käyttäjä oppii miten Linux toimii, miten eri osat ovat toisistaan riippuvaisia ja toimivat keskenään. LFS myös opettaa miten sen voi räätälöidä omien tarpeiden mukaiseksi. (Linux from scratch 2014, viitattu 5.11.2014.) LFS-järjestelmän kehittämiseksi tarvitaan jokin olemassa oleva jakelu jossa on vähintään kääntäjä, linkkeri ja komentorivi (Beekmans 2014, 22). LFS-kirjassa kehitetty Linux-järjestelmä on minimaalinen, mutta se on suunniteltu vahvaksi pohjaksi minkä päälle käyttäjä voi lisätä mitä tahansa paketteja. LFS:n toisissa kirjoissa, kuten BLFS:ssä (Beyond Linux From Scratch) käydään enemmän läpi järjestelmän räätälöintiä. (Linux from scratch 2014, viitattu 5.11.2014.) Niiden avulla käyttäjä voi kehittää järjestelmän lähes mitä tahansa tarkoitusta varten. (Linux from scratch 2015, viitattu 6.3.2015.) LFS sopii myös hyvin sulautettuja järjestelmiä varten jolle on oma projektinsa; Cross Linux From Scratch (CLFS). Siinä käydään läpi miten LFS käännetään eri arkkitehtuureille kuten virtapiheille ARM-järjestelmille. (Cross Linux From Scratch 2013, viitattu 6.3.2015.)

Opinnäytetyön tavoitteena on oppia kehittämään ja ymmärtämään LFS-projektin ohjeiden mukaisesti GNU/Linux-järjestelmiä. Opinnäytetyössä kerrotaan GNU/Linuxin perusasioita, kuten sen historiaa, osia, sulautettuja järjestelmiä, jakelua ja niiden kehittämistä ja lopulta käytännön tasolla LFS-järjestelmän rakentamista. Linuxin osista on tarkoitus saavuttaa yleisymmärrys siitä miten ne toimivat ja mitä ominaisuuksia niillä on. Osien yksityiskohtiin kuten ohjelmakoodiin ei perehdytä.

Kehitystehtävänä on luoda itsenäinen ja toimiva Linux-reititin LFS:n pohjalta, jolla voidaan reitittää verkkoliikenne lähiverkon ja Internetin välillä. Se toimii samalla palomuurina ja tarjoaa DHCP- ja DNS-palvelut lähiverkkoon. Reitittimen kehitysympäristön alustana toimii VMware player ja LFS-järjestelmän ydin käännetään ja optimoidaan sille. Reititin tulee myöhemmin pääosin tekijän omaan käyttöön VMware vSphere:n päälle. Kehitysympäristön käyttöjärjestelmänä toimii Arch Linux, koska se on tekijälle entuudestaan tuttu, helppokäyttöinen ja pakettivarastoiltaan hyvin kattava jakelu. Reitittimet ovat pääosin sulautettuja järjestelmiä, jonka takia opinnäytetyössä ne sisällytetään teoriaosuuteen myöhempiä kohdejärjestelmiä ajatellen.

Opinnäytetyön aiheeseen liittyen tekijä osaa jo perusasiat Linuxin käytöstä, eli komentotulkin käytön, ohjelmien kääntämisen ja asentaminen ja erilaisten jakelujen käyttö on hyvin tuttua. Tekijä on kohdannut Linuxin käytössä erilaisia ongelmia, jotka ovat liittyneet ajureihin, ohjelmistoihin ja konfigurointiin ja on osannut hakea Internetistä ohjeita ja dokumentaatiota niiden ratkaisemiseen. Tekijä osaa myös hyvän tavan mukaiset tavat kysyä neuvoa erilaisilta tukipalstoilta ja sähköpostilistoilta ongelmatilanteissa joihin ei löydy ratkaisua muuten. LFS-järjestelmän rakentamisessa noudatetaan LFS-kirjan versio 7.6:n ohjeita. Käytännön raportin osuuteen ei kopioida kirjan ohjeita vaan kirjataan kohdat joissa poikettiin tai ilmeni ongelmia.

3 GNU/LINUX

3.1 Unix ja Linuxin historia

GNU/Linuxin ydin on ohjelmoitu alusta asti uudestaan perustaen Unix-järjestelmän ytimeen ja on siten sen kanssa yhteensopiva. Linux ei kuitenkaan noudata Open Groupin Unix-määrittäjäsiä, mutta on kuitenkin käytöltään ja käyttökohteiltaan hyvin Unix-järjestelmän kaltainen ja tukee useita Unixille luotuja vapaita ohjelmia.

Unix on C-kielillä kirjoitettu moniajo- ja monen käyttäjän käyttöjärjestelmä, jonka kehityksen aloittivat Dennis Richie ja Ken Thompson. Alun perin Unix kehitettiin konekielillä, mutta vuonna 1973 kehitettiin C-kieli, jolle Unix ohjelmoitiin uudelleen siirrettävyyksensä vuoksi. Varhaisimmat ideat Unixista syntyivät kuitenkin jo vuonna 1965 aloitetussa MIT:n (Massachusetts Institute of Technology) MULTICS-käyttöjärjestelmäprojektissa (MULTIplexed Information and Computing Service). Myöhemmin Unixin jatkokehitys annettiin yliopistojen tehtäväksi. (Kuutti & Rantala 2007, 5.)

Vuonna 1977 Kalifornian yliopistossa Berkeleyssä kehitettiin BSD (Berkeley System Distribution), joka on yksi merkittävä Unix-muunnos (McGee 2014, viitattu 27.10.2014). BSD:n viimeinen virallinen versio julkaistiin vuonna 1993, mutta sen jälkeen BSD:n kehitystä ovat jatkaneet FreeBSD, NetBSD ja OpenBSD. Toinen merkittävä UNIX-muunnos on System V, jonka kehitti AT&T ja Sun vuonna 1988. (Kuutti & Rantala 2007, 5.) System V:stä kehitettiin neljä merkittävää julkaisua, nimeltään julkaisu 1, 2, 3 ja 4. Julkaisu 4, toiselta nimeltään SVR4, oli kaikista suosituin ja vaikutusvaltaisimmin julkaisu. Se oli lähde usealle yleiselle ominaisuudelle moderneissa UNIX-järjestelmissä. (The Linux Information Project 2005, viitattu 28.10.2014.) GNU projektin toteutus vapaasta Unix-järjestelmästä on GNU Hurd. Free Software Foundation aloitti sen kehityksen vuonna 1990 ja kehitys jatkuu yhä tänäkin päivänä. Viimeisin versio GNU Hurd:ista on 0.5 ja se julkaistiin 27.9.2013. (gnu.org 2013, viitattu 28.10.2014.) GNU Hurd:ia on pidetty osittain jopa Linuxia edistyneempänä, mutta sen keskeneräisyyden takia se ei sovellu tuotantokäyttöön.

Standardi-Unixia ei juurikaan ole olemassa. Valtaosa Unix-järjestelmistä on sekoitus Unix-määrittelyistä useista eri lähteistä. RISC-pohjaiset prosessorit ovat olleet Unix-kehittäjien

suosiossa, mutta nykyään Unix-järjestelmiä kehitetään kasvavassa määrin Intelin x86-prosessorisarjalle. Pääosin Unix-järjestelmiä kehitetään valmistajien omia laitteistoja varten.

Linux sai alkunsa harrasteprojektina Linux Torvaldsin toimesta. Hän aloitti kehitystyön omasta aloitteesta ja julkaisi ensimmäisen epävirallisen version 0.02 vuonna 1991. Julkaisun jälkeen Linux-projektiin osallistui yli sata henkilöä kunnes lopulta vuonna 1994 julkaistiin Linux v1.0. Kehitystyötä nopeutti erityisesti Internetin olemassaolo yliopistomaailmassa, minkä ansiosta kehitykseen osallistui ihmisiä ympäri maailmaa. (Kuutti & Rantala 2007, 5.) Linuxin kehitystahtia voidaan pitää erityisen nopeana kun verrataan esimerkiksi Microsoftin Windows 95:n kehitysaikaa, mikä on hieman paranneltu versio Windows 3.11:sta ja silti sen kehittämiseen meni lähes yhtä kauan kuin Linux v1.0:n kehittämisessä. On syytä ottaa myös huomioon että Linux on alusta alkaen uutta ohjelmakoodia. (Koski 2008, 8.)

3.2 GNU/Linuxin ydin

Linux-järjestelmä voidaan jakaa neljään kerrokseen: ytimeen, komentotulkkiin, X-ikkunointijärjestelmään ja sovellusohjelmiin. Graafiset sovellukset toimivat ikkunointijärjestelmän päällä ja merkkipohjaiset toimivat komentotulkin päällä. Ytimen päätehtävä on toimia rajapintana sovellusten ja laitteiston välillä. Linux noudattaa Portable Operating System Interface for uniX (POSIX) standardia, joka määrittelee Unix-käyttöjärjestelmältä vaaditut systeemikutsutut. Näiden avulla ydin tarjoaa palveluja mm. prosessien, muistin, tiedostojärjestelmän ja I/O-palvelujen hallintaan. (Kuutti & Rantala 2007, 18.)

Linuxin ohjelmarajapinta noudattaa useita Unix-versioille kehitettyjä standardeja ja yksi merkittävä piirre on se, että Linux on monen käyttäjän moniajokäyttöjärjestelmä. Käytännössä tämä tarkoittaa sitä, että yhtä prosessoria ja muita tietokoneen resursseja voi käyttää useampi käyttäjä yhtä aikaa. Moniajossa voidaan ajaa useita ohjelmia samaan aikaan. Moniajo jaetaan keskeyttävään ja sopimuksenvaraiseen moniajoon, joista keskeyttävää käytetään Linuxissa. Keskeyttävässä moniajossa laitteisto pysyy aina järjestelmän hallinnassa. Sitä pidetään parempana ja vakaana, koska sopimuksenvaraisessa moniajossa väärin ohjelmoidut ohjelmat ja laitteet eivät välttämättä palauta järjestelmän hallintaa käyttöjärjestelmälle, mikä voi johtaa tietokoneen jumiutumiseen. Monen käyttäjän moniajo on ollut aikoinaan merkittävä hyöty tietokoneessa niiden kalleuden takia suhteessa päätteisiin ja kirjoittimiin. (Koski 2008, 5–6.)

Linuxin modulaarinen ydin on ohjelmoitu suurilta osin C-kielellä ja pieni osa symbolisella konekielellä. Modulaarisuudella tarkoitetaan sitä, että muistissa pidetään vain tarvittavat moduulit. Moduulit ovat laiteajureita ja niitä voidaan liittää järjestelmään suorituksen aikana. Linux-ydin on alun perin suunniteltu 32-bittiseen Intelin 80386-järjestelmiin, mutta nykyään Linux-ydin voidaan myös kääntää mm. AMD 64-bit, IA-64, ARM ja jopa Playstation 3-järjestelmille. (Kuutti & Rantala 2007, 18–19.) Linuxin ytimen versiosta 2.x lähtien se on tukenut useampaa prosessoria, mikä on mahdollistanut ohjelmien ajon rinnakkain ja sitä on pidetty merkittävän hyötynä erityisesti palvelinkoneissa. Moniprosessoritukea kutsutaan nimellä SMP (symmetric multiprocessing). Linux 2.x tarjosi myös RAID:in (Redundant Array of Independent Disks) eli tuen levymatriiseille. (Koski 2008, 6.) RAID mahdollistaa useamman fyysisen levyn yhdistämisen yhdeksi loogiseksi levyksi, mikä parantaa suorituskykyä, vikasietoisuutta ja mahdollistaa suuremmat levyosiot (Jacobi, L. 2012, viitattu 28.10.2014). Linuxin ytimen versiossa 2.2.x SMP- ja RAID-ominaisuuksia on kehitetty lisää ja niiden on todettu olevan jo sopivia tuotantokäyttöön (Koski 2008, 7). Linuxin ydin vaatii vähintään Intelin 80386-prosessorin, 2 megatavua muistia ja yhden levykeaseman. Jotta järjestelmällä voisi tehdä jotain hyödyllistä, tarvitaan enemmän muistia ja levytilaa. (Merrill 2004, viitattu 28.10.2014.)

Linuxin ytimen versionumerointi on jaettu kolmeen osaan, jonka ensimmäisessä osassa numero kasvaa kun muutokset ytimeen ovat olleet merkittäviä. Toisessa osassa numeron parillisuus tarkoittaa vakaata tuotantoversiota ja pariton luku kehitysversiota. Viimeinen luku kasvaa kun tulee pieniä muutoksia, korjauksia tai lisäyksiä. (Kuutti & Rantala 2007, 19.) Päivämäärällä 11.4.2015 viimeisin vakaa Linux-versio on 3.18.11 (The Linux Kernel Archives 2015, viitattu 11.4.2015).

3.3 Muistinhallinta

Linux käsittelee muistia neljän kilotavun kokoisina siivuina, mikä mahdollistaa ohjelmien tarvelatauksen ja virtuaalimuistin tehokkaan käytön. Tarvelatauksella tarkoitetaan suoritettavan ohjelman lataamista muistiin vain siltä osin, mikä on sillä hetkellä suorituksessa. Käyttäjä voi halutessaan ottaa käyttöön virtuaalimuistin. Virtuaalimuisti on RAM-muistin jatke, joka sijaitsee kiintolevyllä. Virtuaalimuisti voi olla joko oma swap-osio tai erillinen tiedosto. Tarpeen vaatiessa Linux voi käyttää siivuittain levymuistia tarpeen mukaan. (Kuutti & Rantala 2007, 20.) Swap-osion

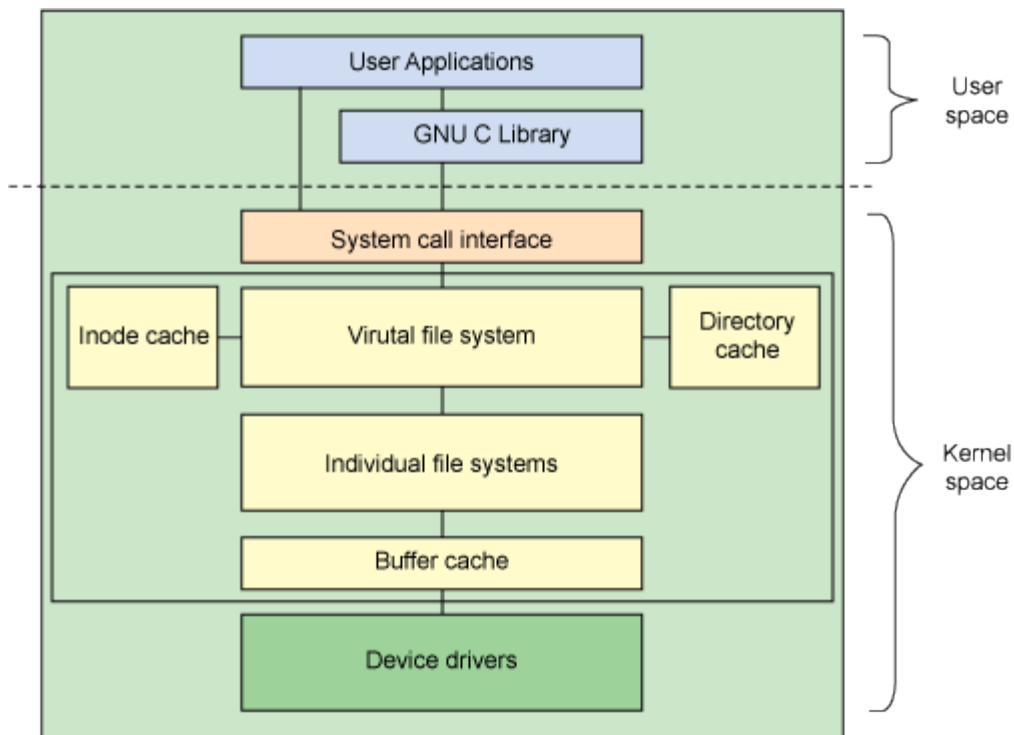
tai -tiedoston suositeltu koko riippuu tietokoneen keskusmuistin määrästä ja käyttökohteesta. Tietokoneissa joissa on esimerkiksi 256 megatavua muistia, virtuaalimuistia suositellaan varaamaan 1,5–2 kertainen määrä suhteessa fyysiseen muistiin. Tietokoneissa joissa on useita gigatavuja fyysistä muistia, ei virtuaalimuistia tarvitse välttämättä ottaa käyttöön ollenkaan. (Linux.fi 2008, viitattu 5.11.2014.)

Muistinhallinnan muita ratkaisuja ovat jaettu koodi ja jaetut kirjastot. Jakamalla joitakin järjestelmän vakiopalveluja omina kirjastoinaan, voidaan säästää muistia. Ohjelma on silloin riippuvainen siitä, että järjestelmässä on käytössä kyseinen kirjasto, mutta se säästää muistia ja levytilaa kun useasti käytetystä toimenpiteestä on vain yksi kopio muistissa ja levyllä. Jaetulla koodilla tarkoitetaan ohjelmakoodin jakamista usean käyttäjän kesken. Sovellus ladataan vain kerran muistiin vaikka sitä käyttäisikin useampi käyttäjä. (Kuutti & Rantala 2007, 20.)

Jos muistia jää ohjelmilta ja järjestelmältä paljon vapaaksi, sitä hyödynnetään dynaamisesti levyjen välimuistina. Dynaamisuudella tarkoitetaan sitä, että muistialueet merkitään tarpeen vaatiessa välimuistin osiksi ja muistin käydessä vähiin sitä voidaan palauttaa normaalin muistin osaksi. Muistia säästyy, koska muistin sisältöä ei tarvitse kopioida paikasta toiseen turhaan ja samaa tietoa ei pidetä kahtena kopiona muistissa. (Koski 2008, 9.)

3.4 Tiedostojärjestelmät

Tiedostojärjestelmä yleisesti on järjestelmä tietoa ja metatietoa tallennuslaitteella. Linuxissa tiedostojärjestelmä on rajapinta, joka toteutetaan kerroksittaisena arkkitehtuurina. Siinä erotellaan tiedostojärjestelmän käyttäjärajapinta ajureista, jotka ohjaavat itse tiedontallennusvälinettä. (IBM 2007, viitattu 7.11.2014.) Linux tukee useita erilaisia tiedostojärjestelmiä erilaisiin tarkoituksiin. Ne tarjoavat erilaisia ratkaisuja tietoturvan, koon, nopeuden ja vikasietoisuuden välillä ja huolehtivat tiedostojen ja hakemistojen hierarkiasta. Virtual File System (VFS) toimii rajapintana Linuxin ytimen ja käytettävän tiedostojärjestelmän välillä mahdollistaen sen, että samalla tiedontallennusvälineellä voi olla useampi eri osio eri tiedostojärjestelmillä.



KUVIO 1. Arkkitehtuurikuva Linuxin tiedostojärjestelmän osista (IBM 2007, viitattu 19.12.2014).

Linuxin kannalta tärkeimpiä tiedostojärjestelmiä ovat ext-tiedostojärjestelmät (extended filesystem). (Kuutti & Rantala 2007, 23.) Viimeisin ext-järjestelmä tänä päivänä on ext4, joka tukee jopa eksatavun kokoisia tiedostojärjestelmiä ja 16 teratavun kokoisia tiedostoja olettaen, että käytetään 4 kilotavun kokoisia lohkoja. Ext4-tiedostojärjestelmää edelsi ext3, josta lähtien ext on ollut journaloiva tiedostojärjestelmä. (IBM 2009, viitattu 7.11.2014.)

Journaloivassa tiedostojärjestelmässä pidetään aina kirjaa siitä mihin kohtaan levyä tullaan seuraavaksi kirjoittamaan. Kaikki kirjaustapaukset tallennetaan omalle varatulle levyalueelle. Kun järjestelmän käynnistetään, keskeneräiset kirjaukset pyyhitään pois jolloin tiedostojärjestelmä pysyy paremmin kunnossa erilaisissa häiriötilanteissa. Onnistuneet kirjaukset kuitataan yksinkertaisesti pois. Journalointi tuottaa järjestelmälle hieman lisätyötä, mutta vastapainona järjestelmässä ei tarvitse käyttää mitään virheentarkistustyökaluja häiriötilanteissa, kuten virtakatkoksissa. Käyttäjän täytyy kuitenkin pitää itse huolta tiedostojen palautuksesta, mikäli niitä on jäänyt auki häiriötilanteissa

Linuxissa käytetään hakemistorakennestandardina FHS:aa (Filesystem Hierarchy Standard). (Kuutti & Rantala 2007, 23–24.) FHS on suositus, jossa kuvaillaan käytäntöjä UNIX-järjestelmän sommittelulle. FHS on suosittu Linux-jakeluissa ja Linux Standard Base (LSB) referoi sitä

standardina. (Linux Foundation 2011, viitattu 7.11.2014.) FHS:n noudattaminen mahdollistaa keskenään yhteensopivien Unix-järjestelmien kehittämisen, koska samat asiat löytyvät samasta paikasta. Standardista huolimatta joidenkin Linux-jakelujen hakemistorakenteet poikkeavat hieman toisistaan. (Kuutti & Rantala 2007, 24.)

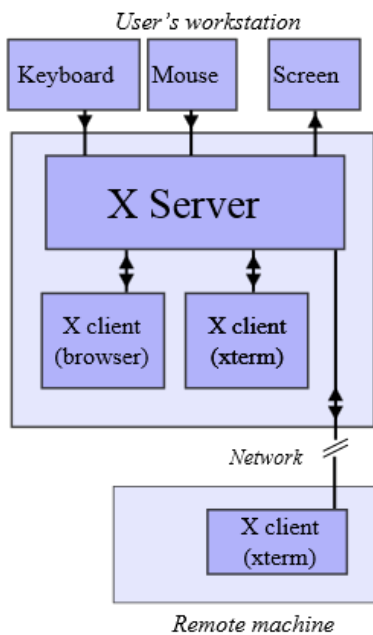
Linux ei käytä Windowsin tapaan levyasematunnuksia vaan Linuxissa levyosiot liitetään suoraan hakemistopuuhun juurihakemiston alle (Koski 2008, 10). Juurihakemisto merkitään kauttaviivalla ja hakemistot järjestellään sen alle puumaiseksi rakenteeksi. Levyasemat voidaan esimerkiksi liittää hakemistoon `"/media/cdrom"` jolloin cd-levyn sisältö löydetään tuosta hakemistosta. Joissakin tapauksissa tiedosto tai hakemisto voidaan asettaa eri paikkaan kuin virallisessa hakemistorakenteessa. Näissä tilanteissa voidaan luoda symbolinen linkki osoittamaan sitä paikkaa, missä tiedosto oikeasti on. (Linux.fi 2011, viitattu 7.11.2014.) Symbolinen linkki on viittaus tiedostojärjestelmässä toiseen sijaintiin. Se voi viitata yksittäiseen tiedostoon, hakemistoon, itseensä tai kokonaan eri tiedostojärjestelmään. (Linux.fi 2015, viitattu 16.4.2015.)

3.5 X Window System

Linux-järjestelmää ohjataan yleensä joko komentotulkin tai graafisen käyttöliittymän kautta. Oletuksena Linuxissa ei ole graafista käyttöliittymää vaan se tarjotaan erillisenä pakettina tai tulee Linux-jakelun mukana. Graafisen käyttöliittymän takana toimii X Window System, joka koostuu X-palvelimesta, X-asiakasohjelmasta ja tyypillisesti myös ikkunointimanagerista. (Kuutti & Rantala 2007, 22.)

X on suunniteltu alusta asti verkkokeskeiseksi ja siksi se käyttää asiakas-palvelin-tekniikkaa. X-palvelimen vastuulla on näppäimistön, hiiren, näytön ja/tai muiden oheislaitteiden I/O-toimintoja. X-asiakasohjelma on graafinen ohjelma, kuten Internet-selain. Se lähettää viestejä X-palvelimelle kuten mihin mikäkin ikkuna piirretään ja X-palvelin vastaa viesteihin käyttäjän tekemillä toiminnoilla kuten mitä nappia käyttäjä on painanut.

Asiakas-palvelin-mallilla on mahdollista suorittaa X-palvelin vähätehoisella työasemalla ja X-asiakasohjelmisto tehokkaalla tietokoneella. Tällöin kommunikaatio ohjelmien välillä tapahtuu verkon yli. Tyypillisesti kuitenkin X-palvelin ja X-asiakasohjelmisto ajetaan samalla tietokoneella.



KUVIO 2. X:n asiakas-palvelinmalli (Wikipedia 2006, viitattu 19.12.2014).

X ei määrää miltä mikäkin ruudulla näyttää vaan sen tekee ikkunointimanageri. Ikkunointimanagereita on useita erilaisia ja jokainen tarjoaa erilaisen ulkonäön ja joukon erilaisia ominaisuuksia. (The FreeBSD Project 2014, viitattu 8.11.2014.) Ikkunointimanageri vastaa ikkunoiden piirtämisen lisäksi niiden sulkemisesta, siirtämisestä, koon säätämisestä ja kaikesta muusta ikkunoiden manipuloinnista.

Ikkunointimanageri toimitetaan usein työpöytäympäristön mukana. Työpöytäympäristö on kokoelma ohjelmia ja aliohjelmakirjastoja, joita yhdistää ulkonäkö ja ovat usein suunniteltuja kokonaisuuksia. Ne tarjoavat keinoja käynnistää ohjelmia, kuten valikkoja ja pikakuvakkeita. Työpöytäympäristö voi tarjota myös yhdistelmän erilaisia sovelluksia erilaisia käyttötarkoituksia varten, esimerkiksi valmiin ohjelmointiympäristön. Työpöytäympäristö ei kuitenkaan vaadi, että ohjelma on suunniteltu juuri sille, vaan eri työpöytäympäristöjen sovellukset usein toimivat myös toisissa työpöytäympäristöissä. Yleisimpiä työpöytäympäristöjä ovat Gnome ja KDE. Ne tulevat mukana useimmissa jakeluissa. Hitaampia koneita varten yksi yleinen työpöytäympäristö on XFce ja sitäkin tarjotaan useissa jakeluissa vaihtoehtona. (Sfnet.atk.linux 2005, viitattu 8.11.2014.)

3.6 GNU Compiler Collection

GCC eli GNU Compiler Collection on osa GNU-projektia, joka on Free Software Foundationin sponsoroima (gnu.org 2014, viitattu 27.10.2014). GCC sisältää käyttöliittymän käyttäjän ja back endin välille ja tarjoaa binaarit C, C++, Objective-C, Fortran, Java, Ada ja Go ohjelmointikielille (gcc.gnu.org 2014, viitattu 27.10.2014). GCC kehitettiin alun perin GNU-käyttöjärjestelmälle, joka myöhemmin yhdistettiin Linuxin kanssa luoden nykyisen GNU/Linux käyttöjärjestelmän (Stallman, R. 2014, viitattu 27.10.2014).

GCC kehitystyön tavoite on houkutella suuria kehittäjäryhmiä tarjoamalla maailmanluokan optimoivan kääntäjän, joka käyttää avointa kehitysympäristöä ja tukee useita alustoja. GCC:n kehitystyö pyrkii sillä varmistamaan, että GCC ja GNU-järjestelmä toimii useissa erilaisissa ympäristöissä ja testaamaan ja laajentamaan GCC:n ominaisuuksia. (gcc.gnu.org 2014, viitattu 27.10.2014.)

3.7 Vapaat ohjelmistot

Vapaissa ohjelmistoissa vapaus on oikeus, jolla ohjelmistoa voidaan käyttää, jakaa, tutkia, kehittää, soveltaa ja myydä ilman mitään rajoitteita. Rajoitteita voi olla mm. erilaiset kokeilujaksot, joihinkin käyttökohteisiin rajoittaminen, maantieteelliset rajoitukset, kopiointikielto tai erilaiset lisensoinnit. (fsfe 2015, viitattu 6.1.2015.) Vapaan ohjelmiston kehittäjä voi kuitenkin vaatia muokkaajia uudelleennimeämään, poistamaan logon tai muuten erottamaan muokatun version alkuperäisestä kunhan vaatimukset eivät estä tai haittaa kohtuuttomasti muokatun ohjelmistoversion julkaisua tai toimintaa. Vapaata ohjelmistoa ei pidä sekoittaa ilmaisiin ohjelmistoihin, koska ilmaisissa voi olla erilaisia rajoituksia ja vapaa ohjelmisto voi olla kaupallinen. Vapaa ohjelmisto täytyy olla saatavilla kaupalliseen käyttöön, kehitykseen ja levitykseen. Alkuperäinen kehittäjä voi myös vaatia että muokatuista versioista lähetetään hänelle kopio ja julkaisujen lähdekoodien jakoa julkisuuteen. (GNU 2014, viitattu 6.1.2015.)

GNU/Linux noudattaa GNU GPL-lisenssiä, mikä on avoimen lähdekoodin ohjelmistoissa käytetty lisenssi. GPL:n alaista ohjelmaa saa vapaasti käyttää ja tutkia kuten vapaita ohjelmistoja, mutta GPL:ään liittyy erityisehtojakin. Kun ohjelmaa jaetaan edelleen, siitä on annettava mukaan lähdekoodi ja lisensointia ei voi muuttaa. Kun ohjelmaan tehdään muutoksia ja ohjelmaa

levitetään edelleen, muutokset on dokumentoitava tai muuten ilmoitettava erikseen levityksen yhteydessä. Ohjelmasta saa ottaa osia ja käyttää muissa ohjelmissa kunhan ne käyttävät GPL:n samaa tai erikseen mainittua versiota ja tekijät mainitaan. (Linux.fi 2014, viitattu 9.1.2015.) GPL-lisenssiä muutaman muun lisenssin ohella pidetään joissain paikoissa vapaiden ohjelmistojen lisensseinä. Debian-jakelu on yksi tunnetuimmista GNU/Linux-jakeluista ja se perustuu vahvasti vapaiden ohjelmistojen periaatteeseen. (Debian 2014, viitattu 9.1.2015.)

3.8 Linux-jakelut

3.8.1 Jakelu

Linux-jakelu (engl. distribution) on kokoelma ohjelmia, jotka on rakennettu Linux-ytimen päälle. Kokoelma koostuu Linux-ytimeistä, pakettienhallintajärjestelmästä, asennusohjelmasta ja useista erilaisista sovelluksista. Linux-jakelun voi kehittää kuka tahansa, koska suurin osa ohjelmista Linuxille on vapaata ohjelmistoa, joiden käyttäminen ja levittäminen on ilmaista. Tämän ansiosta Linux-jakeluita on kehitetty hyvin moneen erilaiseen tarkoitukseen. (Kuutti & Rantala 2007, 9.)

Yksi tunnusomainen piirre Linux-jakelussa on sen pakettienhallintajärjestelmä. Sen löytää kaikista suosituimmista Linux-jakeluista. (distrowatch.com 2014c, viitattu 5.2.2015.) Pakettienhallintajärjestelmällä voidaan asentaa, poistaa ja päivittää Linux-järjestelmään ohjelmistoja. Pakettienhallinnan tehtävä on selvittää riippuvuudet ja automatisoida ohjelmistojen käsittely siten, että käyttäjän ei tarvitse puuttua siihen. Käyttäjän tarvitsee vain tietää asennettavan ohjelmiston nimi ja pakettienhallintajärjestelmä hakee sen riippuvuuksineen jakelun pakettivarastoista. (Linux.fi 2009, viitattu 9.11.2014.) Pakettivarasto on yleensä Linux-jakelun oma varasto paketteja, joka sijaitsee Internetissä HTTP- tai FTP-palvelimella tai joissain tapauksissa erillisellä medially kuten CD- tai DVD-levyllä. Useimpien Linux-jakelujen pakettivarastot ovat niin laajoja, että käyttäjän harvoin tarvitsee etsiä pakettia käsin Internetistä. Kaupallisia ohjelmia usein ei ole pakettivarastoissa johtuen lisenssirajoitteista, mutta epävirallisia pakettivarastoja käyttämällä ohjelman voi löytää. Epävirallisen pakettivaraston luotettavuuden arviointi jää käyttäjälle. (Linux.fi 2013, viitattu 9.11.2014.)

Tunnetuimpia pakettienhallintajärjestelmiä ovat RPM, apt, Portage, tgz ja Pacman. Jokaisella järjestelmällä on omat toimintaperiaatteet ja ovat usein aluksi kehitetty osana Linux-jakelua.

Esimerkiksi RPM-pakettienhallintajärjestelmää käytetään Red Hatissa ja siihen pohjautuvissa jakeluissa. Apt löytyy debianista ja siihen perustuvasta Ubuntusta. Portage on osa Gentootta, tgz osa Slackwarea ja Arch Linux käyttää Pacmania. (Linux.fi 2009, viitattu 9.11.2014.)

3.8.2 Debian

Debian on yksi suosituimmista ja vanhimmista toiminnassa olevista Linux-jakeluista, sen ensimmäinen versio julkaistiin vuonna 1993 Ian Murdockin toimesta. Debian painottaa erityisesti vapaiden ohjelmistojen periaatteeseen ja tarjoaa käyttöjärjestelmän ja kaikki ohjelmistot ilmaiseksi. Debianin mukana tulee yli 37500 esikäännettä ohjelmistoa, jotka on todella helppo asentaa APT-pakettienhallintajärjestelmällä. (Debian 2014a, viitattu 12.12.2014.) Debian on julkaistu virallisesti seuraaville arkkitehtuurille: amd64, armel, armhf, i386, ia64, mips, mipsel, powerpc, ppc64el, s390, s390x ja sparc. Debian on myös julkaistu FreeBSD-ytimelle. (Debian 2014b, viitattu 12.12.2014.) Debianilla ei ole varsinaisesti mitään tiettyä käyttökohdetta, vaan se löytyy hyvin useasta eri paikasta ja käyttötarkoituksesta (Debian 2014a, viitattu 12.12.2014). Debian on hyvin suosittu jakelu, josta on kehitetty omia erillisiä jakeluita. 12.12.2014 päiväyksellä Distrowatchin mukaan Debianista on kehitetty 343 Linux-jakelua, joista 132 on aktiivisia. (Distrowatch 2014a, viitattu 12.12.2014.) Debian pyrkii painottamaan äärimmäiseen vakauteen vakaisissa julkaisuissaan. Vakailta julkaisuilla käyttäjät saavat järjestelmän, jossa on täysi tietoturvatuki ja se ei hajoa, koska kaikki paketit ovat pitkään ja huolellisesti testattuja. Huonona puolena se ei välttämättä tue viimeisintä laitteistoa automaattisesti ja useimmat paketit ovat kohtalaisen vanhoja. Äärimmäinen vakaus on kuitenkin hyvin tärkeää, koska järjestelmää suoritetaan joissain paikoissa 24 tuntina vuorokaudessa seitsemänä päivänä viikossa. (Debian 2014c, viitattu 12.12.2014.)

Ubuntu on Debian GNU/Linuxin perustuva Linux-jakelu. Ubuntun tavoite on tarjota ajantasainen ja yhtenäinen jakelu työpöytä- ja palvelinympäristöön. Ubuntu tarjoaa oletuksena toimisto-, Internet-, kuvankäsittely- ja multimediaohjelmia, mikä tekee siitä sopivan yleiskäyttöön. Ubuntu-projekti on valinnut huolellisesti Debian-jakelun pakettivarastoista suuren määrän paketteja ja hyödyntää sen deb-pakettienhallintajärjestelmää ohjelmien asentamisessa, poistamisessa ja päivittämisessä. (Ubuntu suomi 2014, viitattu 24.11.2014.) Ubuntusta on saatavilla useita erilaisia virallisia julkaisuja: Ubuntu, Kubuntu, Edubuntu, Lubuntu, Xubuntu, Ubuntu studio, Mythbuntu, Ubuntu server, Edubuntu server ja mobiililaitteille on myös omat Ubuntu-julkaisut. Ubuntu-

julkaisut poikkeavat toisistaan esiasennetun työpöytäympäristön ja ohjelmistojen suhteen. (Ubuntu Suomi 2014, Viitattu 24.11.2014.)

Ubuntusta ja Debianista on kehitetty Linux Mint. Se tarjoaa kaksi erilaista versiota: Linux Mint ja Linux Mint Debian Edition (LMDE). LMDE eroaa tavallisesta Linux Mint:istä käyttämällä pelkästään Debiania pohjana ja olemalla "semi-rolling" -jakelu eli järjestelmää ei tarvitse asentaa uudelleen kun tulee laajoja päivityksiä, vaan sitä päivitetään jatkuvasti testaus-jakeluvarannoja käyttäen. Päivityksiä tulee tiheämpään tahtiin, mutta LMDE vaatii käyttäjältä vähän enemmän teknisiä taitoja. Pelkkä Linux Mint pohjautuu enemmän Ubuntuun ja tarjoaa siten käyttäjäystävällisemmän järjestelmän. Molempia Linux Mint-versioita tarjotaan Cinnamon- ja MATE-työpöytäympäristöillä. (Linux Mint 2014, viitattu 26.11.2014.)

3.8.3 Arch Linux

Arch Linux on kanadalaisen Judd Vinetin vuonna 2002 perustama ja myöhemmin amerikkalaisen Aaron Griffinin kehittämä i686 ja x86-64 arkkitehtuureille julkaistu itsenäinen jakelu, joka on riittävän monipuolinen suorittaakseen mitä tahansa roolia. (archlinux 2014a, viitattu 18.12.2014.) Jakelu noudattaa omaa "The Arch Way" filosofiaa, jossa pyritään pitämään kaikki mahdollisimman yksinkertaisena. Arch määrittelee yksinkertaisuuden sillä, että ei tehdä mitään ylimääräisiä lisäyksiä, muokkauksia tai monimutkaisuuksia ja tarjotaan kevyt Unixin kaltainen rakenne, jonka käyttäjä voi muokata omien tarpeidensa mukaiseksi. Arch pyrkii olemaan elegantti ja minimalistinen. (archlinux 2014b, viitattu 18.12.2014.) Arch Linux noudattaa "rolling release" mallia, eli järjestelmä asennetaan kerran ja sen jälkeen sitä päivitetään jatkuvilla päivityksillä ilman tarvetta koskaan uudelleenasettaa järjestelmää tai suorittaa vaikeita jakelupäivityksiä. Arch pyrkii ylläpitämään ja tarjoamaan viimeisimmät pakettiversiot niin kauan kuin vain pakettirikot voidaan välttää. Archin C-ohjelmointikielillä kehitetyllä kevyellä ja yksinkertaisella Pacman-pakettienhallinnalla käyttäjä voi päivittää järjestelmänsä kaikki paketit yhdellä komennolla viimeisimpiin versioihin. Archin virallisista sekä epävirallisista pakettivarastoista käyttäjä voi muokata järjestelmän täysin mieleisekseen. Archin mukana ei tule mitään graafista käyttöliittymää, vaan se jää käyttäjän asennettavaksi muiden ohjelmien ohella. Archin tarjoamia ohjelmia ei ole paikattu muulloin kuin äärimmäisissä tapauksissa. Muuten paketit ovat täysin alkuperäisessä muodossaan sellaisenaan kuin kehittäjä on ne jakanut. (archlinux 2014a, viitattu 18.12.2014.)

3.8.4 Gentoo

Gentoo on Linuxiin tai FreeBSD:hen pohjautuva jakelu, joka voidaan automaattisesti optimoida ja kustomoida lähes minkä tahansa ohjelman tai tarpeen mukaiseksi. Gentoo tunnetaan äärimmäisestä konfiguroitavuudesta, suorituskyvystä ja huippuluokan käyttäjä- ja kehittäjäyhteisöstä. (Gentoo 2007, viitattu 18.12.2014.) Gentoon filosofia on suunnitella työvälineet ja järjestelmät joilla käyttäjä voi tehdä työnsä niin miellyttävästi ja tehokkaasti kuin vaan on mahdollista. Gentoo pyrkii tarjoamaan työvälineet, joita on mukava käyttää ja ne auttavat käyttäjää arvostamaan Linuxin rikkautta ja yhteisön tarjoamia ilmaisia ja joustavia ohjelmistoja. Gentoon mukaan työkaluja tulisi käyttää siten, miten käyttäjä haluaa eikä siten, miten työkalua pitää käyttää. Jos työkalua on pakko käyttää tietyllä tavalla, se toimii käyttäjää sekä Gentoon periaatetta vastaan. (Gentoo 2006, viitattu 18.12.2014.) Gentoo käyttää pakettienhallintajärjestelmänä Portagea. Portage kääntää, asentaa ja päivittää ohjelmat lähdekoodista käyttäjän tarkkojen määrityksiensä perusteella, optimoi ne käyttäjän laitteistolle ja pitää huolen, että ohjelmien vaihtoehdot ominaisuudet ovat päällä käyttäjän tarpeiden mukaisesti. Portagella voidaan myös pitää järjestelmä ajan tasalla yhdellä komennolla. (Gentoo 2007, viitattu 18.12.2014.)

3.8.5 Fedora

Fedora on Red Hat'in sponsoroima projekti, jota kehitetään Red Hat Enterprise Linuxin (RHEL) ohella. Fedora-projektin tehtävä on ajaa eteenpäin ilmaisia ja avoimia ohjelmistoja ja se pyrkii toimimaan yhteistyössä "upstream" kehitystiimien kanssa, jotta kaikki voisivat hyötyä heidän työstään. Fedoran läheinen työ kehittäjien kanssa takaa heille viimeisimmät päivitykset sovelluksiin usein ennen muita. (Fedora 2013, viitattu 26.11.2014.) Ilmaisissa ja avoimen lähdekoodin projekteissa "upstream" on ohjelmiston projekti, joka niitä kehittää (Fedora 2014, viitattu 11.12.2014). Fedora toimii RHEL:in testauskenttänä eli päivitykset ja ohjelmistot jaetaan RHEL:iin kun ne on todettu toimiviksi ja hyväiksi Fedorassa. Fedora on periaatteessa RHEL, mutta ilman kaupallisia kopiosuojattuja ohjelmia ja ominaisuuksia (Kuutti & Rantala 2007, 13). Fedora käyttää RPM-pakettienhallintajärjestelmää ohjelmien asentamiseen, poistamiseen, varmistamiseen, kyselyihin ja päivittämiseen. RPM on vapaasti saatavilla oleva järjestelmä ja siten se löytyy Fedoran lisäksi useista jakeluista kuten RHEL, SUSE, openSUSE, CentOS, Meego ja Mageia. (rpm 2014, viitattu 19.12.2014.)

Red Hat Enterprise Linux on yksi harvoista kaupallisista Linux-jakeluista ja pohjautuu Fedoraan. Sitä kehittää Red Hat ja se on pääasiassa suunnattu kaupallisiin tarkoituksiin. RHEL:in palvelinversiota julkaistaan x86, x86-64, PowerPC, IBM System z ja Itanium arkkitehtuureille ja työpöytäversiota x86 ja x86-64 arkkitehtuureille. Red Hat myös tarjoaa useita maksullisia koulutuksia ja sertifikaatteja RHEL:in ympärille. Red Hat tarjoaa RHEL:in lähdekoodit vapaasti, mutta tiukoilla tavaramerkkisäännöillä ja ilman kaupallisia ja kopiosuojattuja osia. (distrowatch 2014b, viitattu 19.12.2014.)

CentOS on yhteisön tukema kaupalliseen Red Hat Enterprise Linuxiin pohjautuva jakelu, jonka tarkoitus on olla yhteensopiva RHEL:in kanssa tarjoamalla sen ilmaiset sovellukset. Toisin kuin RHEL, CentOS on ilmainen käyttää ja jakaa ja se ei tarjoa kaupallisia sovelluksia, mutta kuitenkin pyrkii olemaan yritystason käyttöjärjestelmä. CentOS-projekti tarjoaa puitteet mm. kehitysympäristön pilvipalvelujen tarjoajille, ylläpitoyhteisöille ja tieteelliselle datan käsittelylle. CentOS-projekti auttaa useita yhteisöjä kerrostamaan ja jakamaan heidän ohjelmistojaan tehokkaammin heidän luottamallaan alustalla. (CentOS 2014, viitattu 24.11.2014.)

3.9 Sulautetut Linux-järjestelmät

Sulautettu käyttöjärjestelmä on muun kuin tietokoneen käyttöjärjestelmä. Laitte voi olla esimerkiksi puhelin, tukiasema, verkkokiintolevy, älytelevisio tai navigaattori. (Koski 2010, viitattu 12.11.2014.) Tyypillisesti sulautetun laitteen käyttöjärjestelmä suorittaa vain yhtä sovellusta, joka auttaa laitetta tekemään tehtävänsä. Laitteita, jotka pystyvät yhdistämään Internetiin kutsutaan älylaitteiksi ja niitä laitteita jotka eivät voi yhdistää, kutsutaan tyhmiksi. Sulautettujen laitteiden laskentatehot ja muisti ovat erittäin rajoittuneita ja virrankulutus erittäin tiukasti säädettyä, minkä takia ohjelmien kehitys sulautetuille laitteille on hyvin tarkkaa työtä. (Whatis.com 2014, viitattu 12.11.2014.)

Sulautetun Linuxin kehitykseen kuuluu yleisesti ottaen kolme kerrosta: käynnistyslataaja, Linuxin ydin ja graafinen käyttöliittymä. Käynnistyslataaja on ensimmäinen asia mikä suoritetaan laitteella. Työasemilla ja muilla tavanomaisilla tietokoneilla käynnistyslataaja ladataan Master Boot Recordiin (MBR) tai ensimmäiselle Linux-levyn lohkolle. Tietokoneissa BIOS antaa kontrollin

käynnistyslataajalle, mutta sulautetuissa järjestelmissä usein ei ole BIOS:ia jonka takia tarvitaan joko erikoistunutta ohjelmistoa tai erittäin pieni käynnistysohjelma.

Laitteen flash-muistin tiettyyn sijaintiin voidaan asentaa käynnistyslataaja erikoistuneilla ohjelmilla. Flash-muisti on pysyvää eli sitä ei tyhjennetä kun laite käynnistetään tai suljetaan. Joissain sulautetuissa laitteissa on minimaalinen käynnistyskoodi (engl. tiny bootcode), joka alustaa joitakin DRAM-asetuksia ja mahdollistaa seriaal-, usb- tai ethernet-portin kommunikoinnin isäntäohjelmien kanssa. Isäntäohjelmat voivat sitten hyödyntää tätä yhteyttä siirtääkseen käynnistyslataajan kohdelaitteen flash-muistiin.

Kun käynnistyslataaja on asennettu ja sille on annettu kontrolli, se alustaa järjestelmän käynnistyksessä prosessorin kellotaajuuden, muistit, mahdolliset serial-portit, suorittaa POST:in, asettaa pino-osoittimen ja muuttuja-alueita, rakentaa muuttujarakenteita ja -merkkejä, tarjoaa tuen virranhallinnalle keskeyttämiseen ja jatkamiseen ja lopulta siirtyy ytimen käynnistämiseen. Kun ydin on ladattu, käynnistyslataaja siirtää järjestelmän kontrollin sille.

Tiedostojärjestelmän valintaperusteet eivät ole samat sulautetuille järjestelmille kuin tavallisille tietokoneille. (IBM 2002, viitattu 18.11.2014). Sulautetuissa järjestelmissä käytetään usein flash-muistia, minkä avulla saadaan tavanomaisiin kovalevyihin verrattuna useita etuja. Flash-muistit ovat pienempiä, vähävirtaisempia, kestävämpiä, nopeampia ja eivät sisällä liikkuvia osia. (Teach-ICT.com 2015, viitattu 6.3.2015.) Flash-muisteissa on kuitenkin haittojakin, joita voi lieventää käyttämällä eri tiedostojärjestelmää. Flash-muisteissa on rajoittunut elinkaari, joka on suurin piirtein 100000 pyyhintää. Sen jälkeen flash-muistia ei voi käyttää. Tämän takia esimerkiksi Ext2fs ei ole hyvä valinta flash-muistin tiedostojärjestelmäksi, koska kun ext2fs-tiedostojärjestelmässä kirjoitetaan tai pyyhitään tietoa pois, koko sektori kopioidaan muistiin, pyyhitään ja sitten uudelleen kirjoitetaan, mikä aiheuttaa ylimääräistä kulumista flash-muistiin. Samalla ext2fs ei sovellu järjestelmiin, joissa saattaa tulla yhtäkkisiä virtakatkoksia ja se ei jaa rasi- tai tasaisesti flash-muistissa vähentäen muistin elinikää. Vaihtoehtona ext2fs:lle on kehitetty erityisesti sulautetuille järjestelmille tarkoitettu JFFS2. JFFS2 ei kirjoita kokonaisia sektoreita uusiksi vaan lisää loppuun. Samalla JFFS2 on paljon varmempi kaatumisia ja virtakatkoksia vastaan (IBM 2002, viitattu 18.11.2014.)

Tavallisten tietokoneiden tapaan myös sulautetut järjestelmät usein tarvitsevat erilaisia lisälaitteita, jotta niitä voidaan hyödyntää. Näitä ovat muun muassa kosketusnäytöt, hiiret,

näppäimistöt ja sensorit. Jotta näitä laitteita voitaisiin käyttää sulautetuissa järjestelmissä, tarvitaan laiteajureita. Merkittävin laiteajuri on ruutupuskuri, koska järjestelmän ruutu toimii sitä kautta. Se sisältää yleensä kolme tasoa, joista alin on yksinkertainen konsoliajuri, joka tarjoaa osan yleisestä käyttöliittymästä tekstille konsolissa. Sen avulla voidaan tulostaa pelkästään tekstiä ruudulle. Toisen tason ajuri tarjoaa yleisen käyttöliittymän videotilaa varten ja kolmannen tason ajuri on erittäin laitteistoon erikoistunut ajuri, jonka täytyy tukea näytönohjaimen erityisiä ominaisuuksia. Kaikki kolme ajuritasoa ovat keskenään riippuvaisia videotuen tarjoamista ajatellen.

Muita merkittäviä ajureita sulautetuissa järjestelmissä ovat syöttölaitteiden ajurit ja flash MTD-ajurit (Memory Technology Device). Syöttölaitteiden ajurien tehtävä on välittää kosketusnäytöltä, näppäimistöltä, sensoreilta tai muilta niiden kaltaisilta laitteilta tieto ohjelmille. Flash MTD -laitteita ovat flash-sirut, muistikortit ja -tikut. MTD-ajurit ovat erityisesti suunniteltu sulautettuja Linux-järjestelmiä varten. Ilman MTD-ajureita käytettäisiin block-tyyppisten laitteiden ajureita kun MTD-ajurit tarjoavat paremman tuen, hallinnan ja rajapinnan sektori-pohjaisille pyyhkimisille, lukemisille ja kirjoituksille flash-laitteissa.

Useimmilla sulautetuilla järjestelmillä on näyttöpäätte, jonka takia graafinen käyttöliittymä on pakollinen. Graafisen käyttöliittymän pitää olla erityisen tarkka muistin käytöstä, kevyt ja nopea käynnistymään, mutta silti helppo käyttää ja luotettava. Graafista käyttöliittymän voi valita useista ilmaisista sekä että kaupallisista vaihtoehdoista. Yleisimpiä ilmaisen lisenssin omaavia graafisia käyttöliittymiä sulautetuille järjestelmille ovat Xfree86 4.X, Microwindows ja FLTK API Microwindowsille. Kaupallisista vaihtoehdoista tunnetaan Qt/Embedded. (IBM 2002, viitattu 18.11.2014.)



KUVIO 3. Raspberry B+ -malli (Raspberry Pi 2012, viitattu 19.12.2014). Raspberry Pi on suosittu sulautettujen laitteiden harrastamisen ja opiskelun parissa (Linux.com 2014, viitattu 19.12.2014).

4 LINUX-JÄRJESTELMÄN KEHITTÄMINEN

4.1 Linux From Scratch

LFS:n voi riisua jopa alle 100 megatavun kokoiseksi asennukseksi ilman suuria haasteita. Jotkin käyttäjät ovat työskennelleet erittäin pienien sulautettujen LFS-järjestelmien kanssa riisuen ne jopa 8 megatavun kokoisiksi. LFS rakennetaan täysin lähdekoodista, minkä ansiosta käyttäjä voi tarkistaa ja valita kaikki ohjelmistot itse jos niin haluaa.

LFS pyrkii noudattamaan POSIX.1-2008, FHS ja LSB standardeja mahdollisimman tarkasti. FHS-standardissa noudatetaan versiota 3.0 Draft 1. LSB-standardia noudatetaan, jotta yksityisomistuksellisia sovelluksia voitaisiin asentaa ja suorittamaan. LFS-järjestelmä on kuitenkin lähdekoodiin pohjautuva järjestelmä, jolloin käyttäjällä on täysi valta sen suhteen, mitä järjestelmään asennetaan ja siten LSB-standardia ei ole pakko noudattaa. (Beekmans 2014, 10.)

LSB määrittelee järjestelmän rajapinnan käännetyille sovellusohjelmille ja asennuskomentosarjoille minimaalisen tukiympäristön. LSB:n tarkoitus on mahdollistaa yhdenmukaisen teollinen standardi suurille määrälle sovelluksia. (ISO 2012, viitattu 10.11.2014.) LSB-sertifikaatin läpäisevän LFS-järjestelmän kehitys on mahdollista, mutta se vaatii joitain paketteja joita ei käydä läpi LFS-pääkirjassa. Niiden asennus käydään läpi BLFS-kirjassa. (Beekmans 2014, 10.)

4.2 Edellytykset

LFS-järjestelmän rakentaminen ei sovellu aloittelevalle Linux-käyttäjälle. Se vaatii jonkin verran kokemusta Unix-järjestelmien hallinnasta, jotta käyttäjä osaisi ratkaista ongelmia ja noudattaa kirjan ohjeita. Käyttäjän pitäisi vähintään osata käyttää komentotulkkia tiedostojen kopioimiseen, siirtämiseen ja muuhun tiedostonhallintaan. Käyttäjän pitäisi myös osata kääntää ja asentaa sovelluksia. LFS-tukifoorumit eivät välttämättä auta sellaisissa tapauksissa, jossa vika on käyttäjän perustaidoissa. (Beekmans 2014, 16.)

LFS-kirja on tarkoitettu x86- ja x86-64-arkkitehtuurisille järjestelmille, mutta joillakin muokkauksilla LFS-järjestelmän voi kääntää myös ARM- tai Power PC-järjestelmille (Beekmans 2014, 9).

Isäntäjärjestelmän Linux ytimen versio tulisi olla 2.6.32 tai uudempi. Binutils 2.24, GCC 4.9.1 ja Glibc 2.20 versioita uudempia ohjelmia ei suositella käytettävän, koska niitä ei ole testattu. (Beekmans 2014, 17.)

4.3 Pakettienhallinta

LFS:n kirjassa ei käytetä minkäänlaista pakettienhallintaa. Sillä voitaisiin seurata minne asennetut tiedostot menevät myöhempää poistamista ja päivittämistä ajatellen, mutta pakettienhallintajärjestelmän kehittäminen ei kuulu LFS:n idean piiriin. Se veisi liikaa painopistettä pois Linux-järjestelmän kasaamisesta. Pakettienhallinnan toteuttamiselle on useita vaihtoehtoja, mutta ei ole olemassa koko lukijakuntaa miellyttävää vaihtoehtoa. Pakettienhallinnan voi kuitenkin halutessaan toteuttaa mm. asentamalla eri hakemistoihin, symbolisia linkkejä tai aikaleimoja käyttämällä, asennuskomentosarjoja seuraamalla tai pakettiarkistoja luomalla.

LFS:n ohjeita noudattamalla pakettien päivittäminen on mahdollista ilman pakettienhallintajärjestelmää. Ensimmäisenä poikkeuksena on kuitenkin glibc, jonka päivittäminen edellyttää LFS:n uudelleenikäntämisen. Toisena poikkeuksena on jos päivitettävän jaetun kirjaston nimi muuttuu, täytyy kaikki ne paketit kääntää uudelleen, jotka käyttivät tuota jaettua kirjastoa. (Beekmans 2014, 81–84.)

4.4 Laittehallinta

LFS-järjestelmässä laitteet saavat omat solmunsu /dev-hakemiston alle Udev:in menetelmää käyttämällä. Kyseisessä menetelmässä Linuxin ytimeen käännetyt ajurit rekisteröi objektinsa sysfs-tiedostojärjestelmälle sitä mukaan kun ne havaitaan ytimen toimesta. Moduuleina käännetyt ajurit rekisteröidään kun moduuli ladataan. Sysfs-tiedostojärjestelmä liitetään /sys-hakemiston alle ja sitten rekisteröidyt ajurit näkyvät käyttäjätilan prosesseille ja udevd:ille prosessointia varten. Laitetiedostot luodaan ytimen toimesta devtmpfs-tiedostojärjestelmän kautta. Laitteet jotka tarvitsevat laitesolmun, menevät devtmpfs:n kautta tehdäkseen sen. Jos devtmpfs-instanssi liitetään /dev alle, sille annetaan aluksi kiinteä nimi, oikeudet ja omistaja. Udevd luo myöhemmin symboliset linkit laitteelle, muuttaa oikeuksia tai nimen riippuen siitä miten järjestelmä on konfiguroitu. Moduuleiksi käännetyt ajurit sisältävät aliaksen. Aliakset yleensä liittyvät

väyläkohtaisten moduulien tukemien laitteiden tunnisteisiin. Näitä tietoja vertaamalla laiteajurit ladataan jos laite havaitaan.

USB-laitteiden kytkemisessä ydin havaitsee sen ja luo uevent:in. Udevd ottaa vastaan uevent:it ja käsittelee ne samaan tapaan kuin muutkin laitteet. (Beekmans 2014, 213–214.)

4.5 LFS-järjestelmän kokoaminen

4.5.1 Kehitysympäristön valmistelu

LFS-järjestelmä rakennetaan valmista Linux-jakelua hyödyntäen. Isäntäjärjestelmä toimii aloituskohtana, jonka täytyy tarjota välttämättömät sovellukset kuten kääntäjän, linkkerin ja komentorivin.

Ensimmäisenä luodaan natiivi Linux-osio ja -tiedostojärjestelmä LFS-järjestelmää varten. Seuraavaksi ladataan paketit ja paikkaukset, mitä tarvitaan LFS-järjestelmässä ja tallennetaan ne oikeisiin sijainteihin hakemistopuussa. Ennen kuin paketteja aletaan asentamaan, LFS:aa varten täytyy rakentaa oikeanlainen työympäristö työkalusarjaa (engl. toolchain) varten. Työkalusarjaa käytetään itse LFS-järjestelmän rakentamiseen ja joidenkin pakettien täytyy ratkaista kiertäviä riippuvuuksia, esimerkiksi kääntäjän kääntämiseen tarvitaan kääntäjä. Työkalusarjan luominen vaatii kaksi kierrosta, ensimmäisellä kierroksella käännetään Binutils ja GCC. Seuraavaksi käännetään vielä C-kirjasto eli Glibc. Sen jälkeen käännetään työkalusarjan toinen kierros, jolloin työkalusarja on dynaamisesti linkitetty uuteen Glibc:aan. Tästä eteenpäin loput paketit käännetään toisen kierroksen työkalusarjalla ja siten LFS-asennus ei ole enää riippuvainen isäntäjärjestelmästä muuten kuin ajossa olevan ytimen kohdalta. (Beekmans 2014, 22.)

Binutils täytyy asentaa ensimmäisenä, koska configure-komento ajaa molemmat GCC ja Glibc suorittaakseen useita testejä kääntäjään ja linkkeriin tietääkseen, mitä ohjelman ominaisuuksia otetaan käyttöön ja mitä ei. Väärin konfiguroitu GCC tai Glibc voi johtaa hieman rikkiinäiseen työkalunippuun ja sitä ei välttämättä huomaa vasta kuin työn loppupuolella kun kasattu järjestelmä ei toimi. Testisarjoilla voidaan havaita ongelmat ennen kun työssä edetään liian pitkälle. (Beekmans 2014, 54.)

4.5.2 Chroot

Chroot-sana tulee sanoista "change root" (Beekmans 2014, 247). Sillä voidaan kirjautua virtuaaliseen ympäristöön uudella komentotulkilla, jonka root-hakemisto osoittaa LFS-osioon. Tämä toimenpide on verrattavissa järjestelmän uudelleenkäynnistykseen, jossa ohjeistetaan ydintä liittämään LFS-osio juuriosioiksi. LFS-järjestelmä ei kuitenkaan ole vielä riittävän valmis, että se voitaisiin oikeasti käynnistää. Sen takia on olemassa chroot, jotta voitaisiin tehdä lisätoita isäntäjärjestelmästä käsin. Chroot:in etuna on, että isäntäjärjestelmää voidaan käyttää normaalisti kun LFS-järjestelmää käännetään. (Beekmans 2014, 22.)

Chroot on ollut olemassa jo Unixista asti ja sitä on käytetty ohjelmien suorittamiseen rajoitetussa ympäristössä. Chroot:ia voidaan käyttää potentiaalisesti vahingollisen ohjelman käyttämiseen turvallisesti tai käyttäjän oikeuksien rajaamiseen järjestelmässä. Chroot:atulle ohjelmalle tai käyttäjälle ei tulisi antaa järjestelmän root-oikeuksia, koska niillä voidaan kiertää rajoitukset. Rajoitetuilla oikeuksilla tämä ei kuitenkaan ole mahdollista. (Unixwiz.net 2015, viitattu 8.3.2015.)

4.5.3 Pakettien kääntäminen

Työkalusarjan kääntämisen jälkeen luodaan Linux-ytimen kommunikointiin käyttämät virtuaalisten tiedostojärjestelmien hakemistot eli dev, proc, sys ja run. Ne eivät varsinaisesti sijaitse levyllä vaan tietokoneen keskusmuistissa. Linux-ydin edellyttää myös laitesolmujen console ja null olemassaolon kun järjestelmää käynnistetään.

Ennen kun keskeneräiseen järjestelmään voidaan chroot:ata, dev-hakemistoon liitetään virtuaaliset tiedostojärjestelmät. Siten laitteet voidaan luoda dynaamisesti sitä mukaan kun niitä havaitaan tai käytetään. Tavanomaisesti tämä tehdään Udev:in toimesta kun järjestelmä käynnistetään, mutta koska järjestelmää ei voida vielä käynnistää, käytetään isäntäjärjestelmän dev-hakemistoa. (Beekmans 2014, 80–81.)

Isäntäjärjestelmän dev-hakemiston liittämisen jälkeen LFS-järjestelmään chroot:ataan. Chroot:illa ohjataan järjestelmä käyttämään väliaikaista työkalunippua siihen asti kunnes pysyvät työkalut ovat tehty. Ensimmäisenä toimenpiteenä järjestelmään luodaan FHS-standardin mukainen hakemistopuu ja välttämättömät tiedostot ja symboliset linkit. Välttämättömiä symbolisia linkkejä

ovat mm. bash, cat, echo, pwd ja stty, jotka viittaavat työkalunipun sovelluksiin. Jotkin sovellukset eivät katso PATH-ympäristömuuttujia vaan ohjelmien polut ovat kovakoodattu niihin. Symboliset linkit korvataan varsinaisilla ohjelmilla sitä mukaan kun ne asennetaan. Käyttäjätilejä ja ryhmiä varten luodaan erikseen omat tiedostot, joita tarvitaan muun muassa root-tilin käyttämiseen. Group-tiedostossa listatut ryhmät eivät noudata mitään erityistä standardia vaan se noudattaa Udev:in vaatimuksia. LSB:n ainoa suositus on se, että root-ryhmän ryhmä-ID 0 ja bin-ryhmän ID 1 ovat olemassa. Muuten hyvin toteutetut ohjelmat eivät ole riippuvaisia ryhmien ID:istä vaan niiden nimistä. (Beekmans 2014, 86–87.)

LFS 7.6:n perusasennus sisältää 64 pakettia alkaen Linux API Header:in ja glibc:n kääntämisestä ja asentamisesta ja loppuu Linux ytimen kääntämiseen ja asentamiseen. Glibc kääntämisen jälkeen työkalunippua muutetaan siten, että se viittaa lopulliseen glibc-asennukseen. Tämän jälkeen käännetään ja asennetaan valtaosa paketeista. Oletuksena suurimpaan osaan ohjelmista sisältyy debug-symboli, millä voidaan saada ohjelman muistin osoitteet ja rutiinien ja muuttujien nimet. Tiedoilla ohjelmaa voi testata ja kehittää, mutta tavanomaisessa käytössä ne suurentavat ohjelman tai kirjaston kokoa jopa moninkertaisesti. LFS-projektissa näitä tietoja ei tarvita, jonka takia ne riisutaan pois pakettien asennuksen jälkeen. Debug-symbolien poistamisessa pitää käyttää väliaikaista työkalunippua, koska ohjelmat eivät saa olla ajossa kun symboleja poistetaan. (Beekmans 2014, 90–208.)

4.5.4 Järjestelmän konfigurointi

LFS-järjestelmä käyttää System V-käynnistysprosessia, jota on käytetty Unix:issa ja sen kaltaisissa järjestelmissä kuten Linuxissa jo vuodesta 1983 (Beekmans 2014, 210). Se sisältää init-ohjelman ja lukuisia komentosarjoja, joilla voidaan tarkkaan määritellä missä järjestyksessä palvelut käynnistyvät. Init määrää järjestelmän käynnistysjärjestyksen ja mitä ohjelmia käynnistetään. (Linux.fi 2008, viitattu 8.3.2015.)

LFS-järjestelmän käynnistys ja sammutus hoidetaan LFS-bootscripits-paketin avulla. Paketin asennuksen jälkeen tarvittaessa luodaan udev-sääntöjä laitteita varten. Säännöillä voidaan määritellä esimerkiksi nimi verkkoadapterille, jolla on tietty MAC-osoite. Udev säännöillä voidaan myös luoda symboliset linkit CD/Dvd-asemiin.

Laitesääntöjen jälkeen konfiguroidaan verkkoadapttereille IP-asetukset. LFS-kirjassa ei käydä läpi DHCP-palvelun asennusta, joten adapttereille annetaan kiinteät IP-asetukset. System V määrittelee järjestelmän käynnistyksessä IP-asetukset adapttereille jos niin käyttäjä haluaa. Nimipalvelintiedot pitää myös määritellä käsin kun DHCP:ta ei käytetä.

Ensimmäisenä ytimen alustuksen jälkeen järjestelmässä suoritetaan oletuksena init. Sitä varten konfiguroidaan inittab-tiedosto, jossa määritellään käynnistysjärjestys palveluille. Tämän jälkeen määritellään järjestelmän kello, konsolissa käytettävä kieli, fontti ja näppäimistöasettelu ja käynnistyksessä ladattavat tiedostojärjestelmät. (Beekmans 2014, 211–235.)

4.5.5 Linuxin ydin ja käynnistyslataaja

Linuxin ytimen konfigurointi voidaan tehdä joko config-tiedostoa muokkaamalla suoraan tai helpokäyttöisemmin käyttämällä menuconfig-työkalua. Linuxin ydintä kehittävä ryhmä suosittelee, että ytimen hakemistopuu puhdistetaan "make mrproper" -komennolla aina ennen konfigurointia. LFS-kirjassa ydintä ei konfiguroida erityisemmin vaan siihen perehdytään enemmän BLFS:ssä joidenkin pakettien kohdalla. (Beekmans 2014, 237.) Ytimeä kannattaa kuitenkin kytkeä kohdejärjestelmän ajurit käyttöön, jotta ne toimisivat asennetussa järjestelmässä (Linux from scratch 2009, viitattu 9.3.2015). Ytimen konfiguroinnin jälkeen se käännetään ja moduuleina käännetyt ominaisuudet asennetaan. Ydin kopioidaan boot-hakemiston alle. (Beekmans 2014, 237.) Viimeisenä tehtävänä järjestelmään asennetaan käynnistyslataaja. LFS-kirjassa käytetään käynnistyslataajana GRUB:ia, jonka voi asentaa erilliselle levyosiolle, jolloin sitä voi käyttää useampi eri jakelu ilman suurempia ongelmia. GRUB:in asennuksen ja konfiguroinnin jälkeen LFS-järjestelmän voi käynnistää. (Beekmans 2014, 241–242.)

5 LINUX-JÄRJESTELMÄN KOKOAMINEN

5.1 Kehitysympäristö

Kehitysympäristönä toimi VMware player:issä toimiva virtuaalikone. Isäntäjärjestelmän kokoonpano (olennaiset komponentit):

Emolevy: Asrock Z77 Pro3m

Proessori: Intel Xeon 1230v2

Muisti: 8 Gt

Kovalevy jolla virtuaalikone sijaitsee: OCZ Agility 3 120 GB SSD

Virtuaalikoneelle on allokoitu 4 gigatavua muistia, 8 prosessorisäiettä ja kaksi kappaletta 20 gigatavun kokoisia virtuaalisia levyjä. Virtuaalikoneen käyttöjärjestelmänä on 64-bittinen Arch Linux. LFS-kirjassa vaadittujen ohjelmistojen lisäksi virtuaalikoneeseen on asennettu mm. sudo, ssh ja cinnamon-työpöytäympäristö. Itse LFS-järjestelmän asennus tapahtuu pääosin SSH-yhteyden yli käytännön syistä.

5.2 LFS Järjestelmän rakentaminen

Järjestelmän kehitys eteni LFS-kirjan ohjeiden mukaisesti lähes ongelmitta. Muutamissa kohti ilmeni ongelmia, jotka ovat selitettynä kappaleessa 5.2.1.

Kirjassa käytetään termiä SBU (Standard Build Unit) ohjelmistojen käänöksessä menevän ajan mittaamiseen. 1 SBU vastaa sitä aikaa, mitä menee binutils-paketin kääntämisessä. Tässä opinnäytetyössä binutils-paketin kääntämiseen meni aikaa minuutti ja 44,6 sekuntia. Eniten aikaa meni GCC:n kääntämisessä; yli kaksi tuntia.

5.2.1 Pakettien kääntäminen

GCC-4.9.1 kääntämiseen, testaamiseen ja asentamiseen meni eniten aikaa muihin operaatioihin verrattuna. Testeissä ilmeni yksi odottamaton virheilmoitus. LFS-projektiin liittyviä postituslistoja

lukemalla selvisi, että kyseisen virheilmoituksen voi ohittaa, koska yleensä vasta suuri lukumäärä virheilmoituksia viittaa ongelmiin.

Virheilmoitus 1: AddressSanitizer on ohjelmointivirheiden etsinnässä käytetty aputyökalu. Asennettuani GCC:n testasin sen ja tulokset vastasivat odotuksia.

```
"FAIL: g++.dg/asan/asan_test.C -O2 AddressSanitizer_HugeMallocTest
Ident((char*)malloc(size))[-1] = 0 output pattern test"
```

Perl'in asennus ei ollut täysin onnistunut, koska sen ajotiedosto ei ollut korvannut väliaikaista symbolista linkkiä, joka viittasi työkalunippuun. Symbolisen linkin poistamisen ja Perl'in uudelleenasetuksen jälkeen sovellus alkoi toimimaan.

Virheilmoitus 2: XML::Parser-2.42_01:ia asennettaessa ilmeni, että se ei löydä Perl:ia järjestelmästä.

```
Perl thinks: [5.20.0]
Config says: [x86_64-linux]
This may or may not cause problems. Please check your installation of
perl
if you have problems building this extension.
ERROR from evaluation of /sources/XML-Parser-
2.42_01/Expat/Makefile.PL: Failed to opendir
'/tools/lib/perl5/5.20.0/x86_64-linux/CORE' to find header files: No
such file or directory at /tools/lib/perl5/5.20.0/ExtUtils/MM_Any.pm
line 2726.
```

Kmod-18:an testejä tehdessä ilmeni yksi epäonnistunut testi. Kmod-18 ei löytänyt *libz.so.1*:aa vaikka se oli sijoitettu sinne missä sen kuuluukin olla eli */lib/libz.so.1*. Se viittasi symbolisena linkkinä */lib/libz.so.1.2.8*-kohteeseen. Selvää vastausta ei löytynyt, että miksi testi epäonnistui. Asennuksen jälkeen ajettiin komento *ldd /bin/kmod* ja *libz.so.1* löytyi sieltä mistä pitääkin joten on mahdollista, että testi oli virheellinen.

5.2.2 Käynnistyslataaja

Käynnistyslataajan suhteen poikettiin LFS:n ohjeista. GRUB:in asentamisen sijaan käytettiin isäntäjärjestelmän Syslinux-käynnistyslataajaa. GRUB:iin vaihtaminen olisi teettänyt lisää työtä

eikä olisi tarjonnut sillä hetkellä järjestelmän kehittämisen kannalta mitään oleellista uutta tietoa. Syslinuxin konfigurointi on yksinkertaisempaa kuin GRUB:in konfigurointi. Jos LFS-järjestelmä menee rikki, niin sen voi korjata Archin puolelta vanhaa käynnistyslataajaa käyttämällä.

Esimerkki 1. LFS:n käynnistysmerkintä `/boot/syslinux/syslinux.cfg` -tiedostosta.

```
LABEL Linux From Scratch
MENU LABEL Linux From Scratch
LINUX ../vmlinuz-3.16.2-lfs-7.6
APPEND root=/dev/sdb1 ro
```

Käynnistyslataajan näkökulmasta LFS-levy näkyy nimellä `sdb1`. LFS-ydin ladataan isäntäjärjestelmän `/boot`-hakemistosta.

5.2.3 Linuxin ytimen kääntäminen

Linuxin ytimen kääntäminen on suhteellisen helppoa, mutta erilaisten asetuksien määrä on huikea. Ytimen kääntämisessä piti poiketa LFS:n ohjeista ja hakea tietoa muualta. Syynä on se, että järjestelmää ajetaan VMware player:in päällä ja se vaatii muutoksia ytimeen, jotta järjestelmä käynnistyisi. LFS:n ohjeita noudattamalla järjestelmä päätyi käynnistyksessä paniikkiin kun se yritti liittää root-osiota. Syynä olivat puuttuvat laiteajurit.

Linux-ydin optimoitiin VMware Player:ille (liite 1).

Oikean verkkoajurin selvittämiseen oli kaksi keinoa: katsoa isäntäjärjestelmästä `lspci -v`-komennolla mitä ajuria ydin käyttää tai jos isäntäjärjestelmä ei ole käytettävissä, voidaan kytkeä ytimen asetuksista kaikki verkkoajurit käyttöön moduuleina, kääntää ja käynnistää järjestelmä niillä. Sitten katsotaan `lsmod`-komennolla, minkä moduulin ydin on ottanut käyttöön ja seuraavalla ytimen kääntökerralla otetaan vain tuo moduuli mukaan kääntöprosessiin.

5.3 Reitittimen kehittäminen

5.3.1 Tarvittavat ohjelmistot

Ensimmäisenä asennettiin paketit OpenSSL ja OpenSSH. OpenSSL tarvitaan OpenSSH:ta varten ja OpenSSH:lla voi ottaa SSH-yhteyden virtuaalikoneeseen, mikä helpottaa työskentelyä merkittävästi. (Beyond Linux From Scratch 2014a, viitattu 10.3.2015; Beyond Linux From Scratch 2014b, viitattu 10.3.2015.) Bootscripts-paketilla voi asentaa ohjelmat käynnistymään järjestelmän kanssa (Beyond Linux From Scratch 2014c, viitattu 10.3.2015).

Dnsmasq hoitaa DHCP- ja DNS-palvelimen virkaa. Uncomplicated Firewall (UFW) toimii palomuurina ja sillä voidaan myös reitittää liikennettä verkkojen välillä. Dnsmasq ja UFW ei kuulu BLFS-kirjan piiriin, joten niiden asentamiseksi noudatettiin paketin mukana toimitettuja tai kehittäjän sivuilta löytyviä ohjeita. Niiden käyttämät riippuvuudet kuitenkin löytyvät BLFS-kirjasta.

Dnsmasq tarvitsee neljä pakettia toimiakseen, joista glibc ja gmp kuuluvat LFS:n peruasennukseen. Libdbus ja nettle kuuluvat BLFS-kirjan piiriin (Beyond Linux From Scratch 2014d, viitattu 10.3.2015; Beyond Linux From Scratch 2014e, viitattu 10.3.2015). UFW tarvitsee toimiakseen Iptables ja Python2 paketit (Beyond Linux From Scratch 2014f, viitattu 10.3.2015; Beyond Linux From Scratch 2014g, viitattu 10.3.2015). Dhcpd ei vaadi mitään paketteja toimiakseen (Beyond Linux From Scratch 2014h, viitattu 10.3.2015).

VMware player:issä virtuaalikoneelle lisättiin toinen verkkoadapteri ja se liitettiin omaan LAN-segmenttiin. Testi-asiakaskoneeksi asennettiin Windows 7-virtuaalikone ja se kytkettiin Linux-reitittimen kanssa samaan LAN-segmenttiin. Siten se sai Internet-yhteyden ainoastaan Linux-reitittimen kautta.

5.3.2 Verkkoadapterien konfigurointi

Ensimmäinen verkkoadapteri asetettiin hakemaan IP-asetukset DHCP palvelimelta. Ajatuksena oli se, että tämä adapteri olisi kytkettynä Internettiin.

Esimerkki 2. Tiedosto /etc/sysconfig/ifconfig.eth0

```
-bash-4.3# cat /etc/sysconfig/ifconfig.eth0
ONBOOT=yes
IFACE=eth0
SERVICE="dhcpcd"
DHCP_START="-b -q"
DHCP_STOP="-k"
```

Toiselle verkkoadapterille annettiin kiinteät IP-asetukset. Se on yhteydessä lähiverkkoon (LAN-segmentti) ja siinä liitännässä operoi myös DHCP- ja DNS-palvelut.

Esimerkki 3. Tiedosto /etc/sysconfig/ifconfig.eth1

```
-bash-4.3# vi /etc/sysconfig/ifconfig.eth1
ONBOOT=yes
IFACE=eth1
SERVICE=ipv4-static
IP=192.168.5.1
PREFIX=24
BROADCAST=192.168.5.255
```

5.3.3 Dnsmasq

Dnsmasq:in asennus ei vaadi paketin ohjeiden mukaan muuta kuin *make install* ja konfiguraatitiedoston kopiointiin */etc*-hakemiston alle (thekelleys.org.uk 2014, viitattu 11.3.2014).

Jotta Dnsmasq käynnistyisi järjestelmän kanssa, sille täytyi tehdä käynnistyskomentosarja. Dnsmasq ei kuulu BLFS-kirjan piiriin, jonka takia oli turhaa etsiä valmista komentosarjaa. Etsimisen sijaan kopioitiin toisen ohjelman käynnistyskomentosarja ja muokattiin se Dnsmasq:ille sopivaksi. Pohjana käytettiin sellaista komentosarjaa, joka oli mahdollisimman yksinkertainen. Tässä tapauksessa *acpid*-paketin komentosarja on hyvä, koska se vain käynnistää ja sulkee ohjelman eikä tee muuta. Komentosarjaan muokattiin vain polut ja käynnistysparametrit Dnsmasq:ille sopiviksi.

Esimerkki 4. Kopioitu *acpid*-käynnistyskomentosarja `~/sources/blfs-bootscripts-20140919/blfs/init.d/dnsmasq`

```
case "$1" in
  start)
    log_info_msg "Starting dnsmasq..."
    start_daemon /usr/sbin/dnsmasq -k --enable-dbus
```

```

    sleep 1
    pidofproc -p "/run/dnsmasq.pid" > /dev/null
    evaluate_retval
    ;;
stop)
    log_info_msg "Stopping dnsmasq..."
    killproc -p "/run/dnsmasq.pid" /usr/sbin/dnsmasq
    evaluate_retval
    ;;
restart)
    $0 stop
    sleep 1
    $0 start
    ;;
status)
    statusproc /usr/sbin/dnsmasq
    ;;
*)
    echo "Usage: $0 {start|stop|restart|status}"
    exit 1
    ;;
esac

```

Bootscripts-paketin Makefile:en täytyi myös lisätä merkinnät Dnsmasq:ia varten. Dnsmasq:in täytyy käynnistyä vasta kun verkkoadaptereilla on IP-asetukset. Verkkoadapterin asetukset tulevat käynnistysprosessissa vasta viimeisimpänä, käynnistystasolla 3. Kaikilla käynnistystasoilla on oma hakemistonsa jonne asetetaan symbolinen linkki, joka viittaa ohjelman käynnistyskomentosarjaan. Symbolisessa linkissä on vielä järjestysnumero, jolla voi määritellä missä järjestyksessä ohjelmat käynnistyvät käynnistystasolla. Dnsmasq:in täytyy käynnistyä vasta verkkoasetusten ja dbus:in käynnistämisen jälkeen, joten Dnsmasq:in symboliselle linkille annettiin luku, joka on suurempi kuin dbus:in ja verkkoasetusten linkeissä. Viimeisenä ajettiin asennuskomentosarja. *\$ make install-dnsmasq.*

Esimerkki 5. ~/sources/blfs-bootscripts-20140919/Makefile -tiedostoon lisätyt rivit

Rivit 52-56

```

install-dnsmasq: create-dirs
    install -m ${MODE} blfs/init.d/dnsmasq          ${EX-
TDIR}/rc.d/init.d/
    ln -sf ../init.d/dnsmasq ${EXTDIR}/rc.d/rc3.d/S30dnsmasq
    ln -sf ../init.d/dnsmasq ${EXTDIR}/rc.d/rc4.d/S30dnsmasq
    ln -sf ../init.d/dnsmasq ${EXTDIR}/rc.d/rc5.d/S30dnsmasq

```

Rivit 586-590

```

uninstall-dnsmasq:
    rm -f ${EXTDIR}/rc.d/init.d/dnsmasq
    rm -f ${EXTDIR}/rc.d/rc3.d/S30dnsmasq
    rm -f ${EXTDIR}/rc.d/rc4.d/S30dnsmasq
    rm -f ${EXTDIR}/rc.d/rc5.d/S30dnsmasq

```

Rivi 1039

```
install-dnsmasq \  
Rivi 1085  
uninstall-dnsmasq \  

```

Esimerkki 6. Dnsmasq:in konfiguraatiodostosto /etc/dnsmasq.conf

```
interface=eth1  
bind-interfaces  
dhcp-range=eth1,192.168.5.100,192.168.5.200,255.255.255.0,48h  
listen-address=127.0.0.1  
listen-address=192.168.5.1  
resolv-file=/etc/resolv.dnsmasq.conf
```

Dnsmasq kuuntelee eth1-verkkoliitännää ja tarjoaa DHCP- ja DNS-palvelut vain ja ainoastaan sitä kautta. IP-osoitteita jaetaan väliltä 192.168.5.100 - .200 ja laina-aika on 48 tuntia. Dhcpd (hakee ip- ja dns-asetukset) konfiguroitiin siten, että se ei koskaan ylikirjoita *resolv.conf*-tiedostoa.

Resolv.conf tiedostoon listattiin dns-palvelimeksi 192.168.5.1, joka kerrotaan LAN-verkon tietokoneille ja *resolv.dnsmasq.conf* tiedostoon listattiin ulkoverkon dns-palvelimet. Tällä järjestelyllä dns-palvelimia saadaan halutessa enemmän kuin kaksi.

Jos verkon laitteella jolla on staattinen IP-osoite halutaan antaa nimiosoite, se pitää lisätä */etc/hosts*-tiedostoon. Dnsmasq hakee nimitietoja ensin *hosts*-tiedostosta ja sitten vasta Internetin nimipalvelimilta, kunhan oma DNS-palvelin on listattu Dnsmasq:issa ennen ulkoverkon palvelimia.

5.3.4 UFW

UFW vaatii toimiakseen python2:n ja iptables:in. Asennusohjeista selviää myös, että Linux-ytimeistä pitää olla aktivoituna *addrtype*, *comment*, *hl (IPv6)*, *limit*, *multiport*, *recent* ja *state* moduuleina. (Launchpad.net 2014, viitattu 11.3.2014.) UFW:n asennuksen jälkeen selvisi, että *conntrack* pitää myös aktivoida. Ytimeistä aktivoitiin lisäksi netfilterin IPv6-moduulit ja IPv6-tuki yleisesti, jotta Linux-reititintä voisi käyttää IPv6-verkossa tarvittaessa.

Dnsmasq:in tapaan UFW:n pitää käynnistyä järjestelmän käynnistyksen yhteydessä. UFW:n käynnistyskomentosarja tehtiin samalla tavalla, mutta */lib/ufw/ufw-init start* käynnistyskomennoksi. Palomuri ajastettiin käynnistymään käynnistystasolla 3 ja järjestysnumerolla 21, mikä on heti verkkoasetusten jälkeen, mutta ennen Dnsmasq:ia.

Seuraavaksi konfiguroitiin palomuuuri reitittämään liikenne verkkojen välillä. Sen toteuttamiseksi `/etc/default/ufw`-tiedostoa muokattiin vaihtamalla `DEFAULT_FORWARD_POLICY`:n `DROP`-arvo `ACCEPT`-arvoksi. Seuraavaksi poistettiin kommentointi `/etc/ufw/sysctl.conf`-tiedoston `net/ipv4/ip_forward=1` kohdasta. Jos halutaan myös reitittää IPv6-verkkoa niin sama pitää myös tehdä alla oleville `IPv6-forwarding` asetuksille. Tässä opinnäytetyössä niin ei tehdä, vaan keskitytään vain IPv4-verkon reitittämiseen. Reititettävien verkkojen määrittely tehtiin `/etc/ufw/before.rules`-tiedostoa muokkaamalla.

Esimerkki 7. `/etc/ufw/before.rules`-tiedosto

```
*nat
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -s 192.168.5.0/24 -o eth0 -j MASQUERADE
:PREROUTING - [0:0]
COMMIT
```

Verkkoliikenne reititetään 192.168.5.0/24-verkon ja eth0-liitännän (Internet) välillä. `MASQUERADE`:n avulla kaikilla lähiverkon laitteilla pääsee Internetiin vain yhtä julkista IP-osoitetta käyttäen. Jos lähiverkon muilla laitteilla, esimerkiksi palvelimella, halutaan ylläpitää jotain verkkopalveluja joihin pitää päästä käsiksi Internetistä, tarvitaan NAT-sääntöjä (Network Address Translation). Nämä säännöt voidaan laittaa `:PREROUTING - [0:0]` alle.

Esimerkki 8. HTTP-protokollan reititys Internetistä IP-osoitteeseen 192.168.5.10.

```
-A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to-destination
192.168.5.10
```

Ennen palomuurin päälle kytkemistä pitää sallia verkkoliikenne lähiverkossa.

Esimerkki 9. Kaiken liikenteen salliminen lähiverkon sisällä.

```
$ ufw allow from 192.168.5.0/24
$ ufw allow 67/udp
```

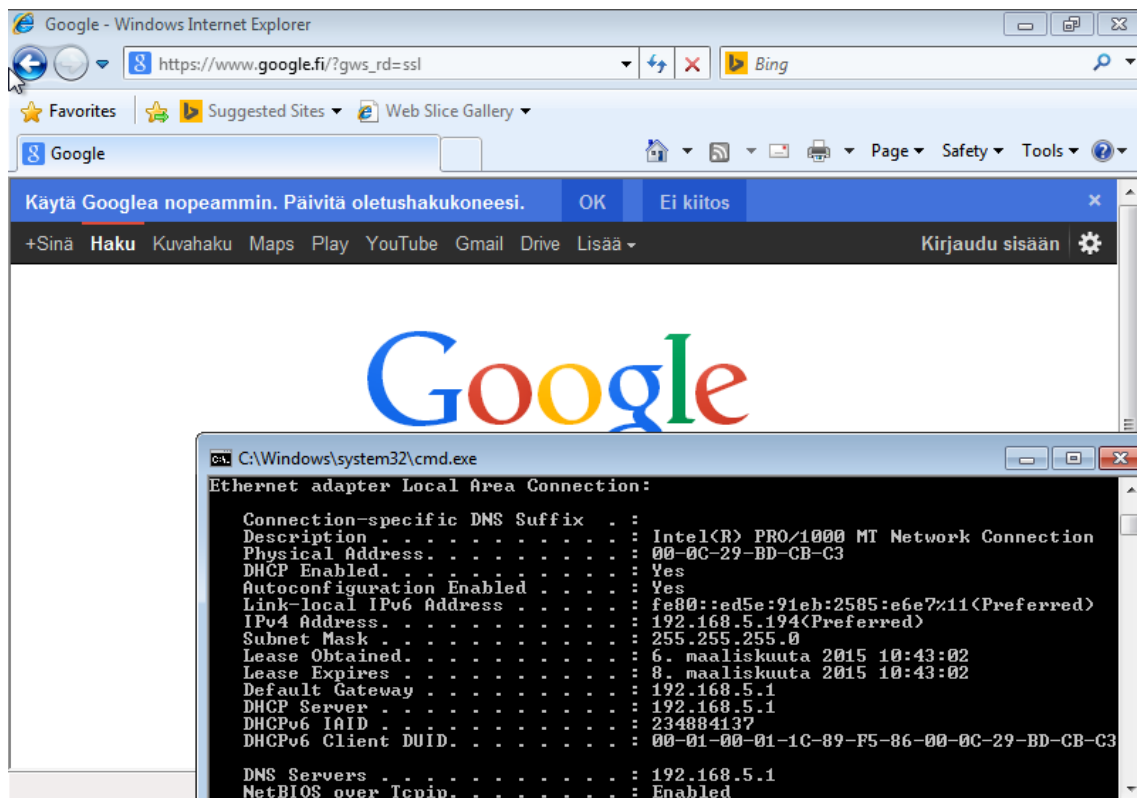
Ensimmäisellä komennolla sallitaan kaikki liikenne reitittimeen 192.168.5.0/24-verkosta. Toisella komennolla sallitaan erikseen DHCP-yhteydenotot. DHCP täytyy sallia erikseen kaikkialta, koska DHCP-yhteydenottoja tehdessä asiakaslaitteella yleensä ei ole vielä IP-osoitetta, minkä

perusteella palomuuuri yleensä sallii tai estää yhteydenottoja. DHCP-palvelu ei kuitenkaan tarjoa IP-osoitteita Internetiin, koska Dnsmasq on konfiguroitu vain kuuntelemaan lähiverkon verkkoliitäntää. Lopuksi palomuuuri kytketään päälle.

Esimerkki 10. UFW:n aktivointi. Tämä täytyi tehdä bootscripitiin lisäksi.

```
$ ufw enable
```

Asiakaslaitteilla pääsee nyt Internetiin.



KUVIO 4: Kuvakaappaus jossa Windows 7-virtuaalikone on hakenut IP-asetukset Linux-reitittimeltä. Taustalla on avattuna Google.

6 POHDINTA

Linux From Scratch -projektin tarjoamat ohjeet ovat laadultaan erinomaiset oman GNU/Linux-järjestelmän rakentamiseen. Kirjassa selitetään mielestäni juuri sen verran jokaisesta aiheesta kuin tavallisen Linux-käyttäjän tarvitsee tietää. Projekti opettaa GNU/Linux-järjestelmän perusosista ja toiminnasta ytimekkäästi ja ymmärrettävästi olettaen, että käyttäjällä on jonkin verran IT-taitoja ja kärsivällisyyttä entuudestaan. LFS sopii hyvin opetteluun tai jonkin erityislaitteen (esim. reititin, NAS-laite) käyttöjärjestelmäksi.

Opinnäytetyössä tein käytännön tehtävänä LFS-järjestelmän päälle Linux-reitittimen, jonka tavoite on tarjota yksinkertaisen reitittimen toiminnallisuudet eli itse reititys, palomuuuri, dns- ja dhcp-palvelut. LFS-järjestelmän rakennus opetti paljon uusia asioita Linux-järjestelmän osista ja toiminnasta. Erityisesti ohjelmien kääntäminen ja ytimen konfigurointi tuli tutuksi. Aikataulullisesti opinnäytetyöhön meni paljon enemmän aikaa kuin aluksi kuvittelin. Teoriaosuudessa halusin käydä Linux-järjestelmän oleelliset osat läpi, joka vei erityisen paljon aikaa ja teki opinnäytetyöstä sivumäärällisesti pitkän. Käytännön osuuden toteutus meni kuitenkin sujuvasti ja en jäänyt missään kohti jumiin.

Jatkokehitysideoina rakentamani Linux-reitittimen ydintä voisi optimoida ja karsia enemmän reitittimelle optimoituun suuntaan. Samalla voisi tutkia, että tarvitseeko aivan kaikkia LFS-kirjassa asennettavia paketteja lopullisessa järjestelmässä. Reitittimelle voisi myös kehittää web-pohjaisen käyttöliittymän, jolla lisätä palomuuuri- ja reitityssäntöjä. Reitittimeen pystyisi myös lisäämään ohjelmia tietoturvaan liittyen.

LFS-järjestelmän kehittämisessä menee huomattavasti enemmän aikaa ja vaivaa kuin olemassa olevaa jakelua käyttämällä. Opinnäytetyön tuoman kokemuksen pohjalta en siksi suosittelisi LFS:ia työasemiin tai palvelimiin ainakaan tavanomaisessa käytössä. Olemassa olevat jakelut tarjoavat monipuolisia ja pitkään kehitettyjä työkaluja joilla voi suorittaa Linux-järjestelmän ylläpitoa LFS:ia vaivattomammin. GNU/Linux-järjestelmän kehittäminen LFS:n pohjalta opettaa paljon, mutta kuitenkin vain raapaisee pintaa. Aiheeseen voisi perehtyä vuosiksi ja silti ei voisi osata kaikkea.

LÄHTEET

archlinux 2014a, Arh Linux. Viitattu 18.12.2014, https://wiki.archlinux.org/index.php/Arch_Linux

archlinux 2014b, The Arch Way. Viitattu 18.12.2014, https://wiki.archlinux.org/index.php/The_Arch_Way

Beekmans, G. 2014. Linux From Scratch Version 7.6, viitattu 10.11.2014, <http://www.linuxfromscratch.org/lfs/downloads/7.6/LFS-BOOK.pdf>

Beyond Linux From Scratch 2014a, OpenSSL-1.0.2. Viitattu 10.3.2015, <http://www.linuxfromscratch.org/blfs/downloads/7.6/BLFS-BOOK-7.6-nochunks.html#openssl>

Beyond Linux From Scratch 2014b, OpenSSH-6.7p1. Viitattu 10.3.2015, <http://www.linuxfromscratch.org/blfs/downloads/7.6/BLFS-BOOK-7.6-nochunks.html#openssh>

Beyond Linux From Scratch 2014c, BLFS Boot Scripts. Viitattu 10.3.2015, <http://www.linuxfromscratch.org/blfs/downloads/7.6/BLFS-BOOK-7.6-nochunks.html#bootscripts>

Beyond Linux From Scratch 2014d, D-Bus-1.8.8. Viitattu 10.3.2015, <http://www.linuxfromscratch.org/blfs/downloads/7.6/BLFS-BOOK-7.6-nochunks.html#dbus>

Beyond Linux From Scratch 2014e, Nettle-2.7.1. Viitattu 10.3.2015, <http://www.linuxfromscratch.org/blfs/downloads/7.6/BLFS-BOOK-7.6-nochunks.html#nettle>

Beyond Linux From Scratch 2014f, Iptables-1.4.21. Viitattu 10.3.2015, <http://www.linuxfromscratch.org/blfs/downloads/7.6/BLFS-BOOK-7.6-nochunks.html#iptables>

Beyond Linux From Scratch 2014g, Python-2.7.8. Viitattu 10.3.2015, <http://www.linuxfromscratch.org/blfs/downloads/7.6/BLFS-BOOK-7.6-nochunks.html#python2>

Beyond Linux From Scratch 2014h, dhcpcd-6.4.3. Viitattu 10.3.2015, <http://www.linuxfromscratch.org/blfs/downloads/7.6/BLFS-BOOK-7.6-nochunks.html#dhcpcd>

CentOS 2014, About CentOS. Viitattu 24.11.2014, <http://www.centos.org/about/>

Cross Linux From Scratch 2013, CLFS Trac. Viitattu 6.3.2015, <http://trac.cifs.org/>

Debian 2014, Debian Social Contract. Viitattu 9.1.2015, https://www.debian.org/social_contract#guidelines

Debian 2014a, About Debian. Viitattu 12.12.2014, <https://www.debian.org/intro/about>

Debian 2014b, Ports. Viitattu 12.12.2014, <https://www.debian.org/ports/>

Debian 2014c, The Debian GNU/Linux FAQ Chapter 3 - Choosing a Debian distribution. Viitattu 12.12.2014, <https://www.debian.org/doc/manuals/debian-faq/ch-choosing.en.html>

Distrowatch.com 2014a, Search Distributions. Viitattu 12.12.2014, <http://distrowatch.com/search.php?ostype=All&category=All&origin=All&basedon=Debian¬basedon=None&desktop=All&architecture=All&status=All>

Distrowatch.com 2014b, Red Hat Enterprise Linux. Viitattu 19.12.2014, <http://distrowatch.com/table.php?distribution=redhat>

Distrowatch.com 2014c, Package Management Cheatsheet. Viitattu 5.2.2015, <http://distrowatch.com/dwres.php?resource=package-management>

eLinux.org 2012, LogFS. Viitattu 18.11.2014, <http://elinux.org/LogFS>

Fedora 2013, Overview. Viitattu 26.11.2014, <https://fedoraproject.org/wiki/Overview>

Fedora 2014, Staying close to upstream projects. Viitattu 11.12.2014, http://fedoraproject.org/wiki/Staying_close_to_upstream_projects

Fotis Koutoulakis 2014, Distro forking 101: How do you fork a Linux distro? Viitattu 24.11.2014, <http://www.fotiskoutoulakis.com/distro-forking-101/>

fsfe 2015, Mitä vapaat ohjelmistot ovat? Viitattu 6.1.2015, <http://fsfe.org/about/basics/freesoftware.fi.html>

gcc.gnu.org 1999, GCC, GCC Development Mission Statement (1999-04-22). Viitattu 27.10.2014, <https://gcc.gnu.org/gccmission.html>

gcc.gnu.org 2014, GCC, the GNU Compiler Collection. Viitattu 27.10.2014, <https://gcc.gnu.org/>

gentoo linux 2006, The Philosophy of Gentoo. Viitattu 18.12.2014, <https://www.gentoo.org/main/en/philosophy.xml>

gentoo linux 2007, About Gentoo. Viitattu 18.12.2014, <https://www.gentoo.org/main/en/about.xml>

GNU 2014, What is free software? Viitattu 6.1.2015, <http://www.gnu.org/philosophy/free-sw.html>

gnu.org 2013, history. Viitattu 28.10.2014, <http://www.gnu.org/software/hurd/history.html>

gnu.org 2014, The GNU Operating system. Viitattu 27.10.2014, <http://www.gnu.org/>

IBM 2002, Linux system development on an embedded device. Viitattu 18.11.2014, <http://www.ibm.com/developerworks/library/l-embdev/index.html>

IBM 2007, Anatomy of the Linux file system. Viitattu 7.11.2014, <http://www.ibm.com/developerworks/linux/library/l-linux-filesystem/>

IBM 2009, Anatomy of ext4. Viitattu 7.11.2014, <http://www.ibm.com/developerworks/linux/library/l-anatomy-ext4/>

ISO 2012, ISO/IEC 23360-1:2006. Viitattu 10.11.2014, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43781

Koski, R. 2008. Linux tehokäytössä. Readme.fi: Helsinki, 5–11.

Koski, R. 2010, Johdatus Sulautettuun Linuxiin. Viitattu 12.11.2014, http://raimokoski.com/Linux2010/Johdatus_sulautettuun_Linuxiin.pdf

Kuutti, W. & Rantala, A. 2007. Linux. 3. laitos, 1. painos. Jyväskylä: WSOY, 5–32.

Launchpad.net 2014, ~jdstrand/ufw/trunk : contents of README at revision 865. Viitattu 11.3.2015, <http://bazaar.launchpad.net/~jdstrand/ufw/trunk/view/head:/README>

Linux Foundation 2011, FHS. Viitattu 7.11.2014, <http://www.linuxfoundation.org/collaborate/workgroups/lfs/fhs>

Linux From Scratch 2009, kernel configuration. Viitattu 9.3.2015, <http://www.linuxfromscratch.org/hints/downloads/files/kernel-configuration.txt>

Linux From Scratch 2015, BLFS Project Homepage. Viitattu 6.3.2015, <http://www.linuxfromscratch.org/blfs/>

Linux from scratch, 2014, What is Linux From Scratch?. Viitattu 5.11.2014, <http://www.linuxfromscratch.org/lfs/>

Linux Mint 2014, Download Linux Mint Debian. Viitattu 26.11.2014, http://www.linuxmint.com/download_lmde.php

Linux.com 2014, Raspberry Pi Spun Into Embedded Module. Viitattu 19.12.2014, <http://www.linux.com/news/embedded-mobile/mobile-linux/769499-raspberry-pi-spun-into-embedded-module>

Linux.fi 2008, Linuxin käynnistysprosessi. Viitattu 8.3.2015, http://www.linux.fi/wiki/GNU/Linuxin_k%C3%A4ynnistysprosessi

Linux.fi 2008, Swap. Viitattu 5.11.2014, <http://linux.fi/wiki/Swap>

Linux.fi 2009, Paketinhallintajärjestelmä. Viitattu 9.11.2014, <http://linux.fi/wiki/Paketinhallintaj%C3%A4rjestelm%C3%A4>

Linux.fi 2011, Hakemistorakenne. Viitattu 7.11.2014, <http://linux.fi/wiki/Hakemistorakenne>

Linux.fi 2013, Pakettivarasto. Viitattu 9.11.2014, <http://linux.fi/wiki/Pakettivarasto>

Linux.fi 2015, Symbolinen Linkki. Viitattu 16.5.2015, [http://www.linux.fi/wiki/Symbolinen linkki](http://www.linux.fi/wiki/Symbolinen_linkki)

Linux.fi 2014, GNU GPL. Viitattu 9.1.2015, [http://linux.fi/wiki/GNU GPL](http://linux.fi/wiki/GNU_GPL).

McGee, M. 2014. What Is a BSD Operating System? Viitattu 27.10.2014, <http://www.wisegeek.com/what-is-a-bsd-operating-system.htm>

MTD 2008, UBIFS - UBI File-System. Viitattu 18.11.2014, <http://www.linux-mtd.infradead.org/doc/ubifs.html>

PCWorld 2012, RAID Made Easy. Viitattu 28.10.2014, <http://www.pcworld.com/article/194360/raid-made-easy.html>

Polishlinux.org 2005, Live CD. Viitattu 26.11.2014, <http://polishlinux.org/choose/live-cd/>

Raspberry Pi 2012, STARTERS GUIDE for Raspberry PI, A, B and B+. Viitattu 19.12.2014, <http://www.raspberrypi.org/forums/viewtopic.php?t=4751>

Scientific Linux 2014, About. Viitattu 9.11.2014, <https://www.scientificlinux.org/about/>

Sfnet.atk.linux 2005, Mitä ovat työpöytäympäristöt ja ikkunamanagerit. Viitattu 8.11.2014, <http://sal-faq.sourceforge.net/html/sal-faq.html#tyopoyta>

Stallman, R. 2014. The GNU Project. Viitattu 27.10.2014, <http://www.gnu.org/gnu/thegnuproject.html>

Teach-ICT.com 2015, What is flash memory? Viitattu 6.3.2015, http://www.teach-ict.com/technology_explained/flash_memory/flash_memory.html

The FreeBSD Project 2014, Chapter 6. The X Window System. Viitattu 8.11.2014, <https://www.freebsd.org/doc/handbook/x-understanding.html>

The Linux Documentation Project 2004, The Linux Kernel. Viitattu 28.10.2014, <http://www.tldp.org/FAQ/Linux-FAQ/kernel.html#does-linux-run-on-my-computer>

The Linux Information Project 2005, System V Definition. Viitattu 28.10.2014, http://www.linfo.org/system_v.html

The Linux Kernel Archives 2015, The Linux Kernel Archives. Viitattu 11.4.2015, <https://www.kernel.org/>

thekelleys.org.uk 2014, Dnsmasq setup. Viitattu 11.3.2015, <http://www.thekelleys.org.uk/dnsmasq/docs/setup.html>

Top500 2014, List Statistics | TOP500 Supercomputer Sites. Viitattu 16.4.2015, <http://www.top500.org/statistics/list/>

TuxRadar 2009, How to build your own Linux distro. Viitattu 10.11.2014, <http://www.tuxradar.com/content/how-build-your-own-linux-distro>

Ubuntu Suomi 2014, Esittely. Viitattu 24.11.2014, <http://wiki.ubuntu-fi.org/Esittely>

Ubuntu Suomi 2014, Lataa. Viitattu 24.11.2014, <http://www.ubuntu-fi.org/lataa.html>

Unixwiz.net 2015, Best Practices for UNIX chroot() Operations. Viitattu 8.3.2015, <http://www.unixwiz.net/techtips/chroot-practices.html>

W3Techs 2015, Usage statistics and market share of Unix for websites. Viitattu 16.4.2015, <http://w3techs.com/technologies/details/os-unix/all/all>

Whatis.com 2014, embedded device. Viitattu 12.11.2014, <http://whatis.techtarget.com/definition/embedded-device>

Wikipedia 2006, X client server example - X Window system. Viitattu 19.12.2014,
http://fi.wikipedia.org/wiki/X_Window_System#mediaviewer/File:X_client_server_example.svg

ZDNet 2013, The secret origins of Google's Chrome OS. Viitattu 26.11.2014,
<http://www.zdnet.com/the-secret-origins-of-googles-chrome-os-7000012215/>

```

[*] 64-bit kernel (leave blank for x86)
General Setup --->
  [*] Optimize very unlikely/likely branches
Processor type and features --->
  Processor Family (usually Core2/Newer Xeon)
  [*] Linux guest support
Bus Options --->
  < > PCCard (PCMCIA/Cardbus) support
Networking support --->
  [ ] Amateur Radio support
  [ ] Wireless (only if you plan on using nat EXCLUSIVELY)
Device Drivers --->
  Generic Driver Options
  [*] Maintain a devtmpfs filesystem to mount at /dev
Misc Devices --->
  [*] VMware Balloon Driver (manages memory between VM and host)
  [*] VMware VMCI Driver (Virtual Machine Communication Interface -
low-latency access to host memory bus)
SCSI device support --->
  [*] SCSI low-level drivers --->
  <*> VMware PVSCSI driver support (high throughput storage adapt-
er)
  [*] Fusion MPT device support --->
  <*> Fusion MPT ScsiHost drivers for SPI
  [*] Network device support --->
  [*] Ethernet driver support --->
  (kytketään tästä valikosta ainoastaan se mitä käytetään)
  [*] Intel Devices
  <*> Intel(R) PRO/1000 Gigabit Ethernet support
  [ ] Wireless LAN (ONLY if you disabled Wireless networking support
above)
  < > VMware VMXNET3 ethernet driver (PCNet32 is more than enough
for most use cases - enable this only if you have spare cpu cycles to
burn)
Graphics support --->
  <*> Direct Rendering Manager
  < > Intel 8xx/9xx/G3x/G4x/HD Graphics
  <*> DRM driver for VMware Virtual GPU
  [*] Enable framebuffer console support under vmwgfx by default
  <*> Support for frame buffer devices
Console display driver support --->
  <*> Framebuffer Console support
Sound card support --->
  <*> Advanced Linux Sound Architecture --->
  [*] PCI sound devices
  <*> (Creative) Ensoniq AudioPCI 1371/1373
  < > Intel HD Audio
File systems --->
  (enable only those you anticipate using)
  <*> Second extended fs support
  <*> The Extended 4 (ext4) filesystem
  <*> XFS filesystem support
  <*> Btrfs filesystem Unstable disk format
Pseudo filesystems --->
  [*] Tmpfs virtual memory file system support (former shm fs)
  [*] Tmpfs POSIX Access Control Lists

```