

TAMPEREEN AMMATTIKORKEAKOULU  
Tietotekniikan koulutusohjelma  
Ohjelmistotekniikka  
Paavo Mäkelä

Opinnäytetyö

## **Sulautetun Linux-järjestelmän rakentaminen**

Työn ohjaaja  
Työn teettäjä  
Tampere 2/2009

Lehtori Tony Torp  
Ixonos Oyj, valvojana projektipäällikkö Matti Salmi

Author(s)	Paavo Mäkelä
Name of the report	Embedded Linux Platform Building
Number of pages	37 pages
Graduation time	4/2009
Thesis supervisor	Lecturer Tony Torp
Commissioned by	Ixonos Plc, project manager Matti Salmi

---

## **ABSTRACT**

The purpose of this work was to increase personal competence in the field of embedded Linux platform building. This was done by studying general principles and components of an embedded Linux system. In addition also adapting knowledge gained in working in a Linux-based smart phone platform development project at Ixonos Plc. Goal of this work was to implement a custom embedded Linux system for Nokia N800 Internet Tablet device.

This thesis goes through backgrounds of using Linux in an embedded system. After background information, components of an embedded Linux platform are described and explained, components are: bootloader, the Linux kernel and root filesystem. After component descriptions, thesis handles some basic tools and methods of the build process. Finally, in the last part, example of an embedded Linux platform implementation using Poky platform builder is explained.

Tampereen ammattikorkeakoulu  
Tietotekniikka  
Ohjelmistotekniikka

Tekijä(t)	Paavo Mäkelä
Työn nimi	Embedded Linux Platform Building
Sivumäärä	37 sivua
Valmistumisaika	4/2009
Työn ohjaaja	Lehtori Tony Torp
Työn tilaaja	Ixonos Oyj, valvojana projektipäällikkö Matti Salmi

---

## TIIVISTELMÄ

Työn tarkoituksena oli lisätä henkilökohtaista tietämystä sulautetun Linux-järjestelmän rakentamisesta. Tietämyksen lisääminen tapahtui opiskelemalla sulautetun Linux-järjestelmän peruseriaatteita ja komponentteja. Lisäksi sovellettiin tietämystä, jota saatiin työskennellessä Linux-pohjaisen älypuhelimien kehitysprojektissa Ixonos Oyj:ssä. Työn tavoitteena oli toteuttaa sulautettu Linux-järjestelmä Nokia N800 Internet Tablet -päätelaitteelle.

Työn alkuosa käsittelee Linuxin käyttöä sulautetussa järjestelmässä ohjelmiston ja rautatuen näkökulmasta. Alkuosan jälkeen työ käsittelee sulautetun Linux-järjestelmän tärkeimpiä komponentteja, jotka ovat: alkulatausohjelma, Linux-ydin ja juuren tiedostojärjestelmä. Komponenttien käsittelyn jälkeen käydään läpi muutamia työkaluja ja tapoja, joita tarvitaan sulautetun Linux-järjestelmän rakentamisessa. Lopuksi toteutetaan esimerkki sulautetun Linux-järjestelmän rakentamisesta, käyttäen Poky Platform Builder -työkalua.

---

Avainsanat                      Linux, sulautettu järjestelmä, alkulatausohjelma, juuren  
tiedostojärjestelmä, poky, arm-arkkitehtuuri

## **Esipuhe**

Tämä tutkintotyö kirjoitettiin syksyllä 2008 ja keväällä 2009 Ixonos Oyj:lle. Työn tekeminen kasvatti tietämystäni sulautettujen Linux-järjestelmien rakentamisesta ja antoi minulle valmiudet työskennellä sulautettujen Linux-järjestelmien parissa.

Kiitän projektipäällikkö Matti Salmea työn ohjaamisesta ja aiheen rajauksesta.

Tampereella helmikuussa 2009

Paavo Mäkelä

## Sisällysluettelo

1 Johdanto.....	9
2 Sulautettu Linux.....	10
2.1 GNU General Public License.....	10
2.2 Standardointi.....	11
2.2.1 Linux Standard Base.....	11
2.2.2 File System Hierarchy Standard.....	11
2.3 Laitteistotuki.....	14
2.4 Ohjelmistot.....	15
3 Sulautettu Linux-järjestelmä.....	16
3.1 Alkulatausohjelma.....	16
3.2 Linux-ydin.....	17
Ytimeen ladattava moduuli.....	19
3.3 Juuren tiedostojärjestelmä.....	19
4 Järjestelmän rakentaminen.....	21
4.1 GNU-työkaluketju.....	21
4.2 Ristiinkääntäminen.....	22
4.2.1 Linux-ytimen kääntäminen kohdejärjestelmälle.....	22
4.2.2 Ohjelmistopakettien kääntäminen kohdejärjestelmälle.....	23
5 Sulautetun Linux-järjestelmän rakentaminen Poky-työkalulla.....	26
5.1 Nokia N800 Internet Tablet.....	26
5.1.1 Nokia bootloader.....	27
5.1.2 Nokia flasher.....	27
5.2 Järjestelmän ohjelmistokomponentit.....	28
5.2.1 Linux-ydin.....	29
5.2.2 GNOME Mobile Platform.....	29
5.2.3 Sato desktop.....	30
5.3 Poky platform builder.....	30
5.3.1 Laittekonfiguraatio.....	31
5.3.2 Käännöstehtävä.....	32
5.4 Järjestelmän rakentaminen.....	33
5.5 Järjestelmän asentaminen päätelaitteeseen.....	34
6 Yhteenveto.....	37
Lähteet.....	38

## Lyhenteet ja termit

Alkulatausohjelma	Tietokoneohjelma, joka lataa tarvittavat ohjelmat käyttöjärjestelmän käynnistymiseksi.
ARM	ARM Limitedin kehittämä 32-bittinen RISC-proessoriarkkitehtuuri
AVR32	Atmelin kehittämä 32-bittinen RISC-proessoriarkkitehtuuri.
BIOS	Basic Input/Output System. Tietokoneohjelma, joka etsii ja lataa käyttöjärjestelmän.
Busybox	Unix-työkaluja sisältävä ohjelmistokomponentti.
Bluetooth	Avoin standardi laitteiden langattomaan kommunikointiin lähietäisyydellä.
CPU	Central Processing Unit, suoritin tai prosessori.
D-BUS	Desktop Bus. Yksinkertainen prosessien välinen kommunikointimekanismi
Defconfig	Konfiguraatiotiedosto, jossa kuvataan Linuxin ytimen toiminnallisuudet.
Käynnistyskoodi	Perinteinen Linuxin käynnistysjärjestelmä.
NFS	Network File System. Verkossa sijaitseva tiedostojärjestelmä.
Jakelupaketti	Kokoelma ohjelmia, jotka on valmiiksi konfiguroitu ja käännetty.
Sulautettu	Tiettyyn tarkoitukseen tehty tietokonejärjestelmä.
FHS	Filesystem Hierarchy Standard. Juuren tiedostojärjestelmän määrittelevä standardi.
GUI	Graphical User Interface. Graafinen käyttöliittymä
GNU GPL	GNU General Public License. GNU yleinen lisenssi, vapaa ohjelmistolisenssi.
GPS	Global Positioning System. Satelliittipaikannusjärjestelmä.
GSM	Global System for Mobile Communications. Maailmanlaajuisesti käytetty matkapuhelinjärjestelmä.
Intel x86	Intelin kehittämä prosessorikäskykanta.
IPC	Inter-process communication. Prosessien välinen liikennöinti.
I/O	Input/Output. Siirräntä.
Jffs2	Journalling Flash File System version 2. Sulautetuissa järjestelmissä käytetty tiedostojärjestelmä.
Kernel	Useimpien käyttöjärjestelmien keskeisin ohjelmistokomponentti.
LKM	Loadable Kernel Module. Ytimeen ladattava moduuli.
Linux	Yleinen termi, jolla viitataan UNIX-tyyppisiin käyttöjärjestelmiin,

jotka perustuvat Linux-ytimeen.

Kosketusnäyttö	Näyttö, joka voi välittää tietoa kosketuksesta eteenpäin.
LSB	Linux Standard Base. Linuxin standardi.
Maemo	Linuxiin pohjautuva käyttöjärjestelmä.
Matchbox	Sulautetuille järjestelmille suunnattu ikkunointimanageriohjelma.
MMU	Memory Management Unit. Muistinhallintayksikkö
Middleware	Ohjelmisto, joka yhdistää ohjelmistokomponentteja tai -sovelluksia.
MBR	Master Boot Record. Pääkäynnistyslohko.
Motorola 68000	A 16/32-bit CISC microprocessor core designed and marketed by Freescale Semiconductor. Freescale semiconductorin suunnittelema ja markkinoima 16/32-bittinen CISC-mikroprosessori.
M32R	Alunperin Renesas Technologyn kehittämä ja valmistama 32-bittinen sulautettu RISC-mikrokontrolleri.
MIPS	RISC-pohjainen prosessoriarkkitehtuuri.
NAND	Flash-puolijohdemuisti.
nano	Tekstieditori.
nolo	Nokia loader. Nokia Oyj:n kehittämä alkulatausohjelma.
Nokia N810	Linux-pohjainen kämmentietokone
Nokia N800	Linux-pohjainen kämmentietokone
OS	Operating System. Käyttöjärjestelmä.
Platform	Kehys, jossa sovelluksia voidaan ajaa.
PowerPC	Power Performance Computing. RISC-pohjainen prosessoriarkkitehtuuri.
POST	Power On Self Test
Qt	Graafinen käyttöliittymäkirjasto.
RISC	Reduced Instruction Set Computing. Prosessorin suunnittelustrategia.
Root filesystem	Juurihakemistoon kiinnitetty tiedostojärjestelmä.
Super-H	32-bittinen lataa/talleta RISC-pohjainen prosessoriarkkitehtuuri.
SDK	Source software Development Kit. Ohjelmistojen kehitysympäristö
Ubuntu	Debian Linux –käyttöjärjestelmään pohjautuva jakelupaketti.
UNIX	A computer operating system originally developed in 1969 by a group of AT&T employees. AT&T:n työntekijöiden vuonna 1969 kehittämä tietokoneen käyttöjärjestelmä.
UI	User Interface. Käyttöliittymä.
USB	Universal Serial Bus. Sarjaväyläarkkitehtuuri oheislaitteiden liittämiseksi tietokoneeseen.

Usbutils	Ohjelmistopaketti, joka sisältää sovelluksia USB-tietojen esittämiseen.
WLAN	Wireless Local Area Network. Langaton lähiverkko.
zImage	Staattisesti linkitetty suoritettava tiedosto, joka sisältää Linux-ytimen.
3G	Third generation. Kolmannen sukupolven matkapuhelinjärjestelmä.



# 1 Johdanto

Linux-käyttöjärjestelmä on ottanut suuren harppauksen kohti sulautettuja järjestelmiä. Nykyään Linux-ytimeen pohjautuvia sulautettuja järjestelmiä löytyy leivänpaahtimista älypuhelimiin, ja kaikkeen siltä väliltä. Linuxin leviäminen työpöytäkoneilta ja palvelimilta sulautettuihin järjestelmiin jatkuu, eikä loppua ole näkyvissä.

Linuxin suosio sulautetuissa järjestelmissä on luonut työkaluja ja toteutuksia integraation helpottamiseksi. Suosion seurauksena markkinoille on tuotu laitteita, joiden käyttöjärjestelmät pohjautuvat Linuxiin. Ja osa näistä laitteista kykenee suorittamaan muunneltuja Linux-järjestelmiä, jotka eivät tule laitteen mukana.

Avoimen lähdekoodin yhteisön ja työkalujen avulla on mahdollista rakentaa oma muunneltu Linux-järjestelmä sulautettuun laitteeseen. Ennen kuin tämä on mahdollista, tarvitaan hyvä ja vankka tietämys sulautetun Linux-järjestelmän komponenteista ja niiden toiminnasta.

Tämä tutkintotyö keskittyy olennaiseen tietoon, jota tarvitaan oman muunnellun Linux-järjestelmän rakentamiseen ja asentamiseen Nokia N800 -päätelaitteeseen. Linux-järjestelmän taustatiedot on kuvattu vain yleisellä tasolla. Sulautetun Linux-järjestelmän komponentit ja niiden toiminta on kuvattu yksityiskohtaisesti. Järjestelmän rakentamiseen tarvittavat välttämättömät sovellukset ja toteutustavat on kuvattu lyhyesti. Viimeisessä kappaleessa esitetään esimerkkitoetus sulautetun Linux-järjestelmän rakentamisesta ja asentamisesta Nokia N800 -päätelaitteeseen.

## 2 Sulautettu Linux

Sulautettu Linux tarkoittaa Linux-käyttöjärjestelmän soveltamista sulautetuissa järjestelmissä, kuten matkapuhelimissa, kämmentietokoneissa, mediasoittimissa, kotiteatterijärjestelmissä, verkkolaitteissa, teollisuuden ohjausjärjestelmissä, navigaattori-laitteissa ja lääketieteen sovellutuksissa.

Linuxin käytön etuja verrattuna muihin sulautettujen järjestelmien käyttöjärjestelmiin on monia. Linux-ytimen käytöstä ei tarvitse maksaa tekijänoikeus- tai lisensointimaksua. Linux-ytimen kehittäminen ei rajoitu vain tietyn ohjelmistoyrityksen työntekijöihin, mikä mahdollistaa Linux-ytimen laaja-alaisen kehittämisen ja vapaan jakelun. Linux-ytimellä on aktiivinen kehittäjäverkosto, jonka avulla Linux-ytimeen saadaan nopeasti lisättyä tuki uusille laitteistoarkkitehtuureille, järjestelmille ja laitteille. Lisäksi Linux-järjestelmille on luotu standardeja, jotka parantavat Linux-järjestelmien keskinäistä toimivuutta.

Linux-järjestelmien haittapuoleksi voidaan laskea suurehko muistinkulutus (Linux-ydin ja ohjelmat), käyttäjätilan (user space) ja ydintilan (kernel space) muistin käyttö ja monimutkainen laitteistoajurikehys. /5/

### 2.1 GNU General Public License

Linux-ydin on julkaistu GNU yleinen lisenssillä (GNU GPL). Lisenssin on alun perin kirjoittanut Richard Stallman GNU-projektille vuonna 1989. GNU yleinen lisenssi on vapaan levitysoikeuden omaava lisenssi ohjelmistoille ja muille töille. Lisenssi vaatii siitä periytyvien töiden julkaisemista saman lisenssin alaisena, tämä takaa GNU yleinen lisenssin pysyvyyden, vaikka lisenssin alaista työtä olisi muutettu tai lisätty.

Seuraava lista kuvaa muutamia GNU GPL -lisenssin pääperiaatteita

- Lisenssi on monistuva.
- Lisenssi antaa käyttäjälle oikeuden suorittaa ohjelmaa.
- Lisenssi antaa käyttäjälle oikeuden opiskella ja muokata lähdekoodia.
- Lisenssi antaa käyttäjälle luvan jakaa alkuperäistä lähdekoodia ja mahdollisia muutoksia.
- Lisenssi takaa samat oikeudet kenellä tahansa, jolle GPL-lähdekoodia jaetaan.

GNU GPL -lisenssi takaa seuraavat neljä vapautta.

- Vapaus käyttää ohjelmaa mihin tahansa tarkoituksiin.
- Vapaus muunnella ohjelmaa omien tarpeiden mukaan.
- Vapaus jakaa ohjelmistoa vapaasti.
- Vapaus jakaa ohjelmaan tehtyjä muutoksia.

/9/

## **2.2 Standardointi**

### **2.2.1 Linux Standard Base**

Luultavammin yksi kaikkein merkityksekkäin standardi Linux-maailmassa on Linux Standard Base (LSB). LSB-standardi kehitettiin ratkaisemaan eri Linux-järjestelmien välisiä yhteensopivuusongelmia ja siten vähentämään Linux-järjestelmän ylläpidosta syntyviä kuluja. LSB-standardi alentaa merkittävästi sovelluksien siirtämisestä eri alustoille syntyviä kuluja.

Tällä hetkellä LSB-standardi kattaa monia eri prosessoriarkkitehtuureja, kuten:

IA32/64, PowerPC32- ja 64-bit, ja muita. Standardi on hajoitettu ydinkomponenttiin ja yksittäisiin arkkitehtuurikomponentteihin.

LSB-standardi määrittelee Linux-jakelun yleisiä ominaisuuksia mukaan lukien:

objektiformaatit, standardit kirjastojen rajapinnat, minimimäärän komentoja ja niiden toiminnallisuuden, tiedostojärjestelmän rakenteen (File System Hierarchy Standard), järjestelmän alustuksen. /1/

### **2.2.2 File System Hierarchy Standard**

File System Hierarchy Standard (FHS) määrittelee päähakemistot ja niiden sisällön.

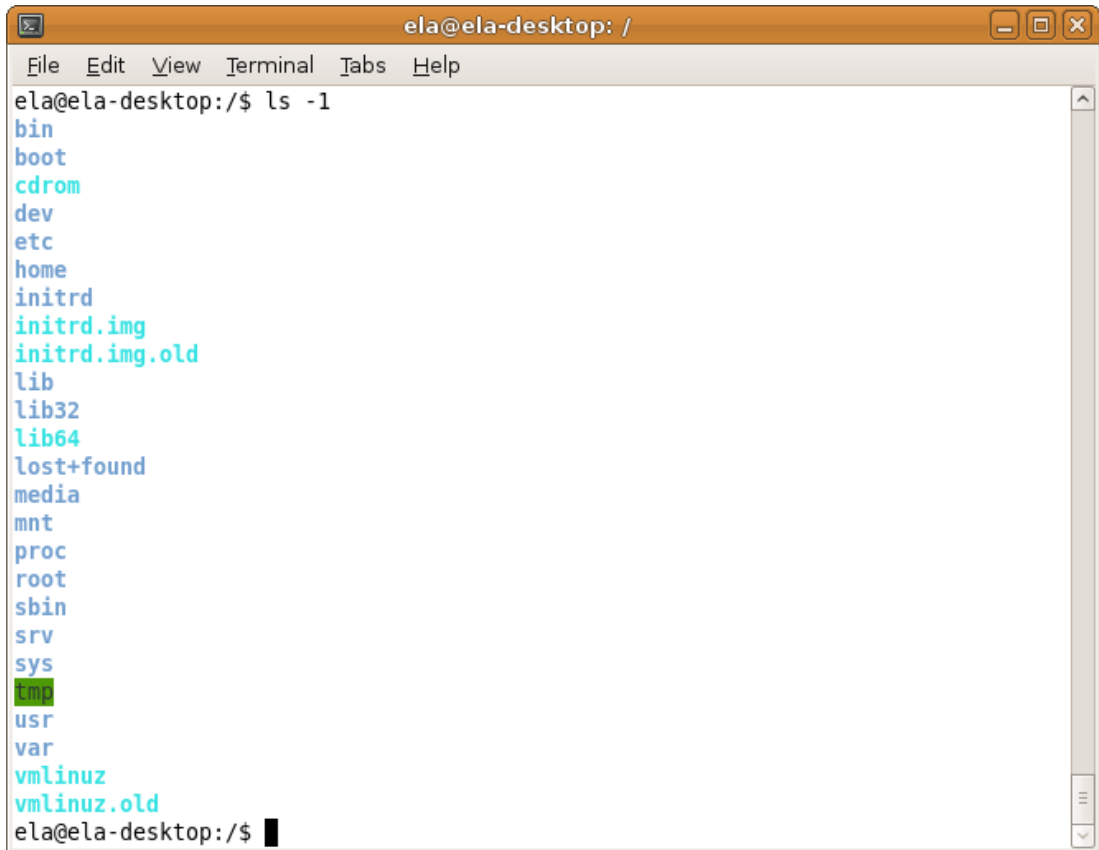
FHS-standardi määrittelee minimitarajan eri Linux-jakeluiden ja sovelluksien yhteensopivuudelle. FHS-standardi kuvaa, millainen juuren tiedostojärjestelmän (root filesystem) pitäisi olla, se myös määrittelee yhteisen rajapinnan eri UNIX- ja Linux-jakeluiden välille. FHS-standardi mahdollistaa ohjelmistokehittäjien ennustaa, missä järjestelmän eri elementit, kuten tiedostot ja hakemistot sijaitsevat. /1/

Taulukko 1: FHS-standardin mukainen juuren tiedostojärjestelmän rakenne.

Hakemisto	Sisältö
/	Päähierarkian juuri ja koko järjestelmän juurihakemisto.
/bin/	Välttämättömien komentojen binäärit.
/boot/	Alkulataajan tiedostot, yleensä eri tiedostojärjestelmä.
/dev/	Välttämättömät laitteet.
/etc/	Järjestelmänlaajuiset asetustiedostot.
/home/	Käyttäjien kotihakemistot.
/lib/	Binäärien tarvitsemat kirjastot.
/media/	Liitoskohta irrotettaville laitteille, kuten CD-ROM-aseuille.
/mnt/	Tilapäisesti liitetyt tiedostojärjestelmät.
/opt/	Valinnaiset ohjelmistopaketit.
/proc/	Virtuaalinen tiedostojärjestelmä, jossa on dokumentoitu kernelin tila.
/root/	Juurikäyttäjän (root) kotihakemisto.
/sbin/	Välttämättömät järjestelmän binäärit.
/srv/	Järjestelmän tarjoama paikkakohtainen tieto.
/tmp/	Väliaikaistiedostot, jotka eivät yleensä säily järjestelmän uudelleen käynnistyksen välissä.
/usr/	Toinen hierarkia käyttäjäkohtaisille tiedoille. Sisältää käyttäjäkohtaiset ohjelmat ja sovellukset.
/var/	Väliaikaistiedostot.

Taulukossa 1 on kuvattu FHS-standardin mukaiset juuren tiedostojärjestelmän hakemistot. /1/

Kuvioissa 1 ja 2 on esitettyä kaksi eri juuren tiedostojärjestelmän hakemistohierarkiaa. Kuviossa 1 on esimerkki sulautetusta Linux-järjestelmästä ja kuviossa 2 on esimerkkityöpöytä Linux-jakelusta.

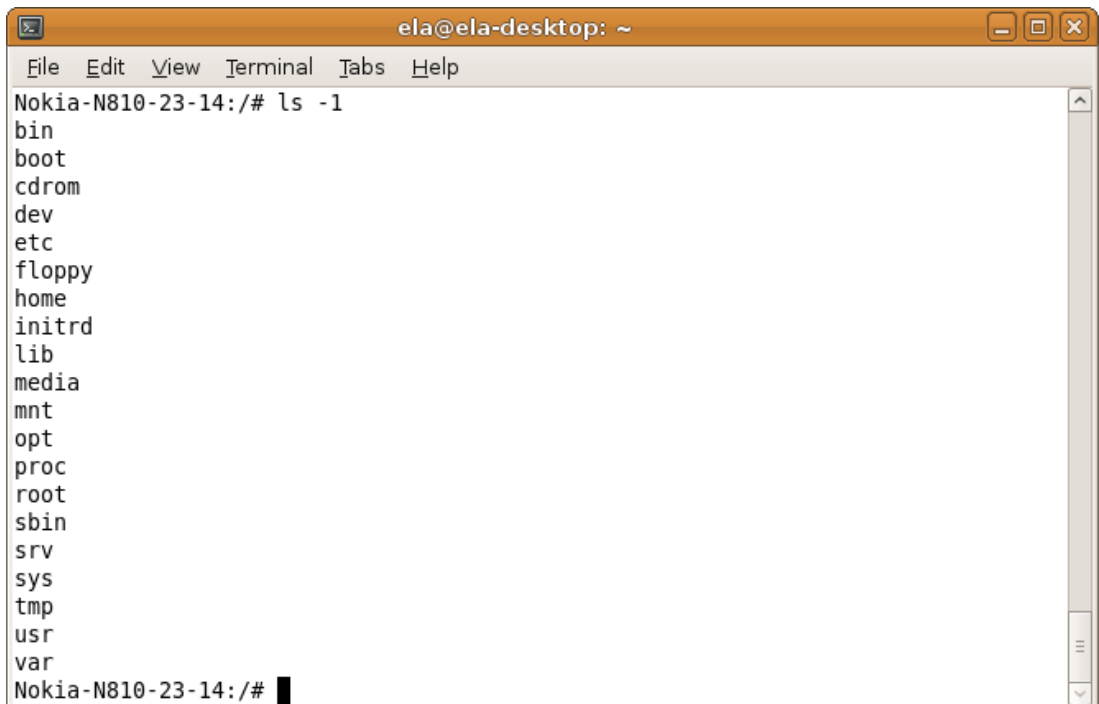


```

ela@ela-desktop: /
File Edit View Terminal Tabs Help
ela@ela-desktop:/$ ls -l
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
initrd.img.old
lib
lib32
lib64
lost+found
media
mnt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
vmlinuz.old
ela@ela-desktop:/$

```

Kuvio 1. Ubuntu Linux 8.10 -jakelupaketin juuren tiedostojärjestelmän rakenne.



```

ela@ela-desktop: ~
File Edit View Terminal Tabs Help
Nokia-N810-23-14:/# ls -l
bin
boot
cdrom
dev
etc
floppy
home
initrd
lib
media
mnt
opt
proc
root
sbin
srv
sys
tmp
usr
var
Nokia-N810-23-14:/#

```

Kuvio 2. Maemo Linux 4.1.2 -käyttöjärjestelmän juuren tiedostojärjestelmän rakenne Nokia N810 -pöytälaiteessa.

Kuten kuvioista 1 ja 2 voidaan huomata, juuren tiedostojärjestelmät eivät suuresti eroa näiden kahden Linux-järjestelmän välillä. Yhteneväisyydet eri järjestelmien välillä ovat FHS-standardin mukaisen toteutuksen ansiota.

## **2.3 Laitteistotuki**

Linux-ydin tukee monia eri prosessoriarkkitehtuureja. Linux-ytimen viimeisin 2.6 – versio tukee vähintään 18 eri prosessoriarkkitehtuureja. Yleisimmin käytetyimmät prosessoriarkkitehtuurit sulautetuissa järjestelmissä ovat ARM, AVR32, Intel x86, M32R, MIPS, Motorola 68000, PowerPC ja Super-H. Yleisimmin käytetty prosessoriarkkitehtuuri mobiili- ja oheislaitteissa on ARM-prosessoriarkkitehtuuri. ARM eli Advanced RISC Machine on ARM Holdings Ltd:n ylläpitämä ja edistämä prosessorituoteperhe. Linux-ytimen versio 2.6 tukee kahdeksaa seuraavaa ARM-pohjaista prosessoria: ARM720T, ARM920T, ARM922T, ARM926EJ-S, ARM1136J(F)-S, ARM1176JZ(F)-S, ARM11 MPCore, Cortex-A8 ja Cortex-A9.

/1/ /2/ /4/

Linux-ydin käsitetään monoliittiseksi ytimeksi, joka sisältää kaikki käyttöjärjestelmätoiminnot samassa osoiteavaruudessa, ydintilassa (kernel space). Monoliittisen rakenteen tuomia etuja ovat käyttöjärjestelmän nopeus sekä helppo kehitysmalli. Huonoina puolina voidaan todeta, että yksikin virhe käyttöjärjestelmän toiminnoissa esimerkiksi muistinhallinnassa, verkkoprotokollassa, levyjärjestelmässä tai laiteajurissa vaikuttaa käyttöjärjestelmän muihin osiin ja saattaa johtaa koko järjestelmän kaatumiseen. Monoliittinen rakenne voi olla myös osittain modulaarinen. Osa käyttöjärjestelmän toiminnoista, kuten laiteajuri, siirretään irralleen ytimestä omaksi käyttöjärjestelmämoduuliksi (Loadable Kernel Module), joka ladataan keskusmuistiin vain tarvittaessa. Modulaarisuus tuo mukanaan monia etuja. Ytimen koko voidaan minimoida, kun toiminnallisuus saadaan ladattua vain tarvittaessa. Tämä nopeuttaa käyttöjärjestelmän käynnistämistä (boot-up), sillä käynnistysvaiheessa ladataan vain sellaiset toiminnot, jotka ovat välttämättömiä. Myöhemässä vaiheessa ladataan toiminnallisuus, jota ei tarvittu käynnistysvaiheessa. Käyttöjärjestelmämoduulit tuovat mukanaan piirteitä, joita voidaan menestyksekkäästi käyttää sulautetuissa järjestelmissä, erityisesti mobiililaitteissa. /7/

## **2.4 Ohjelmistot**

Sulautetut Linux-järjestelmät voivat käyttää samoja ohjelmistoja kuin työpöytäjärjestelmät. Ohjelmistojen käyttöä yleensä rajoittaa kuitenkin prosessorin pieni teho ja muistin rajallinen määrä. Työpöytäjärjestelmissä käytettyjen raskaiden sovelluksien sijaan sulautetut Linux-järjestelmät yleensä käyttävät pienempiä tai riisuttuja versioita. Esimerkkejä sulautetussa Linux-järjestelmässä mahdollisesti käytettävistä ohjelmistoista ovat Busybox (UNIX-työkalukokoelma) ja Matchbox (ikkunointimanageri). Molemmat ohjelmistopakettit saadaan konfiguroitua järjestelmälle sopiviksi ja niiden muistinkulutus voidaan täten minimoida. Yleisesti sulautetuissa Linux-järjestelmissä käytetään tarkoin harkittuja ohjelmistopaketteja. Kantava peruseriaate on, että jos jotain ohjelmistopakettia ei tarvita, sitä ei tulla sisällyttämään järjestelmään.

Joissain sulautetuissa Linux-järjestelmissä riittää prosessoritehoa ja muistia graafisten käyttöliittymien (GUI) suorittamiseen. Linux-järjestelmissä on käytössä kaksi vallitsevaa GUI-kirjastoa, GNOME sekä Qt. Molemmat GUI-kirjastot ovat laajalti käytössä työpöytäkoneisiin suunnatuissa Linux-jakeluissa. Ja niistä löytyy versiot, joita voidaan käyttää sulautetuissa Linux-järjestelmissä. GNOME:sta on GNOME Mobile Platform-mobiilialusta sekä Qt:sta Qt for Embedded Linux -kirjastoversio. Molempien GUI-kirjastojen versiot ovat riittävän kattavia, että niillä voidaan rakentaa sulautetuille järjestelmille sopivia graafisia käyttöliittymiä.

### 3 Sulautettu Linux-järjestelmä

Sulautettu Linux-järjestelmä koostuu kolmesta perusohjelmakomponentista: alkulatausohjelmasta (bootloader), juuren tiedostojärjestelmästä (root filesystem) ja Linux-ytimeistä. Kullakin ohjelmakomponentilla on tärkeä osuus sulautetun Linux-järjestelmän käynnistämisen prosessissa. Kuviossa 3 on esitetty sulautetun Linux-järjestelmän käynnistymisprosessi.



Kuvio 3. Sulautetun Linux-järjestelmän käynnistyssekvenssi.

Käynnistymisvaiheessa alkulatausohjelma on ensimmäinen, joka saa prosessorin käyttöönsä. Alkulatausohjelma suorittaa alemman tason laitteiston alustustoimenpiteitä, jotka mahdollistavat Linux-ytimen lataamisen käyttömuistiin (RAM). Linux-ytimen lataamisen jälkeen, järjestelmänhallinta siirtyy Linux-ytimelle. Linux-ytimen alustustoimenpiteiden jälkeen käyttäjätilan (user space) ohjelmat ladataan juuren tiedostojärjestelmästä. /1/

#### 3.1 Alkulatausohjelma

Alkulatausohjelma on yleensä ensimmäinen osa ohjelmakoodia, joka suoritetaan millä tahansa sulautetussa Linux-järjestelmässä käytössä olevalla laitteistolla. Perinteisissä järjestelmissä, kuten pöytäkoneissa, alkulatausohjelma yleensä ladataan pääkäynnistyslohkoon (master boot record) tai levyaseman ensimmäiseen muistilohkoon. Työpöytäkoneissa ja muissa järjestelmissä BIOS-ohjelma normaalisti siirtää laitteiston hallinnan alkulatausohjelmalle. Sulautetuissa järjestelmissä tämä ei kuitenkaan ole mahdollista, sillä järjestelmissä ei ole käytössä BIOS-ohjelmaa.

BIOS-ohjelman puuttumiseen on käytössä kaksi yleistä tekniikkaa: erityinen ohjelmakomponentti ja pieni käynnistyskoodi.

Erityinen ohjelmakomponentti voi suoraan keskustella flash-muistin kanssa ja asentaa alkulatausohjelman sovittuun muistilohkoon flash-muistilla. Flash-muistit ovat erityisiä



piirejä, jotka käyttäytyvät kuten muut tilalaitteet (storage device) eikä niiden sisältö tuhoudu uudelleenkäynnistyksen yhteydessä.

Joissakin sulautetuissa järjestelmissä on käytössä pieni käynnistyskoodi (bootcode) kooltaan muutamia tavuja. Käynnistyskoodi alustaa tiettyjä DRAM-asetuksia ja mahdollistaa alkulatausohjelman asentamisen sarjaväylän, USB-väylän tai ethernet-verkon kautta isäntäkoneesta (host), riippuen käytössä olevasta laitteistosta. Tämän ansiosta isäntäkoneen ohjelmat tai lataajat voivat asentaa alkulatausohjelman päätelaitteeseen, jossa se kirjoitetaan flash-muistiin. Tekniikkaa sovelletaan esimerkiksi tuotantolinjalla, jossa laitteisiin asennetaan ohjelmistot ensimmäistä kertaa.

Kun alkulatausohjelma on asennettu ja sille on annettu järjestelmänhallintavastuu, suoritetaan seuraavat toimenpiteet:

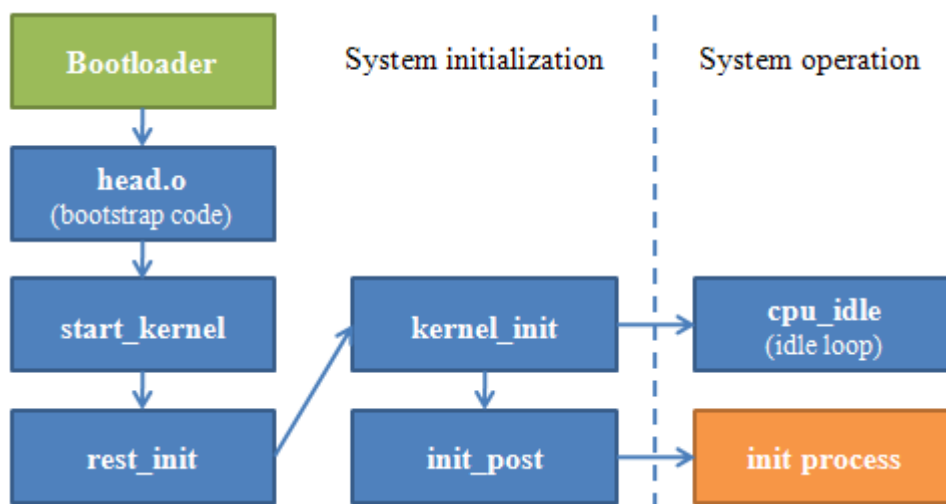
- Asetetaan CPU:n nopeus.
- Muistin alustaminen, johon liittyy seuraavia toimenpiteitä: muistipankkien asettaminen, muistin alustus, rekistereiden konfigurointi ja muita toimenpiteitä.
- Sarjaportin alustus (jos laitteistossa).
- Käsky- ja datapuskurien alustaminen.
- Pino-osoittimen (stack pointer) asettaminen.
- Parametrialueen asettaminen, rakentaminen ja merkitseminen. Linux-ydin käyttää parametrialueen tietoja järjestelmän tunnistamiseen.
- Suorittaa POST-toimenpiteen, jolla tunnistetaan käytössä oleva laitteisto ja raportoidaan laitteistossa ilmenneistä ongelmista.
- Mahdollistetaan virranhallintatuki jäädyttämiselle (suspend) ja uudelleen aloittamiselle (resume).
- Käynnistetään Linux-ydin.

/6/

### **3.2 Linux-ydin**

Linux-ydin alustaa ja hallitsee sulautetun järjestelmän laitteistoresursseja. Linux-ydin voidaan asentaa sulautetun järjestelmän NAND flash-muistille tai muulle flash-muistille (esimerkiksi muistikortille). Prosessoriarkkitehtuurista ja käytetystä alkulatausohjelmasta riippuen Linux-ydin voidaan ladata joko vmlinux- tai zImage-formaateista. Molemmat formaatit ovat tulos onnistuneesta Linux-ytimen käännösprosessista.

Linux-ytimen alustusprosessi riippuu pitkälti prosessoriarkkitehtuurista ja sulautetun järjestelmän ominaisuuksista. Kuviossa 4 on esitetty Linux-ytimen alustusprosessi ARM-prosessoriarkkitehtuurissa.



Kuvio 4. Linux-ytimen alustuskaavio ARM-prosessoriarkkitehtuurissa.

Ensimmäinen toimenpide ennen Linux-ytimen käynnistymistä on, että alkulatausohjelma suorittaa Linux-ytimen head.o-esilatausohjelman (bootstrap code), joka on prosessoriarkkitehtuurista riippuvainen alustuskoodi. Esilatausohjelmalla on muutamia toimenpiteitä, joita sen pitää suorittaa ennen kuin Linux-ytimen start\_kernel-funktiota kutsutaan. Esilatausohjelman pitää tarkistaa käytössä oleva prosessoriarkkitehtuuri, prosessori, laitteistotyyppi, konfiguroida muistinhallintayksikkö (MMU), luoda sivutaulukolle (page table) tietorakenne ja mahdollistaa virtuaalimuistin käyttö. Tarkistus- ja konfigurointitoimenpiteiden jälkeen esilatausohjelma kutsuu Linux-ytimen start\_kernel -funktiota.

Linux-ytimen start\_kernel -funktiossa tapahtuu muutamia tärkeitä toimenpiteitä. Ensimmäiseksi Linux-ytimen käynnistysparametrit luetaan alkulatausohjelmalta. Onnistuneen lukemisen jälkeen prosessorin ja laitteiston alustus suoritetaan luettujen parametrien perusteella. Lyhyesti ilmaistuna start\_kernel -funktio alustaa mahdollisimman aikaisessa vaiheessa komentotulkin ja Linux-ytimen alijärjestelmät, lopulta kutsutaan rest\_init -funktiota. rest\_init -funktiossa aloitetaan uusi Linux-ytimen-säie, josta myöhemmin tulee järjestelmän ensimmäinen prosessi (init process), joka on Linux-järjestelmässä kaikkien prosessien äitiprosessi. Linux-ytimen-säikeen luomisen jälkeen alkaa laitteiston alustus kernel\_init -funktiossa. Laiteajurit, jotka on

käännetty ytimeen staattiseksi osaksi, latautuvat `kernel_init` -funktion suorittamisen aikana. Ytimeen ladattavat moduulit (LKM) eivät lataudu tässä vaiheessa. Kun kaikki laitteisto on alustettu, `kernel_init` -funktio siirtyy `cpu_idle` -tilaan odottamaan käyttäjän syöttämiä komentoja. /1/

### **Ytimeen ladattava moduuli**

Jos Linux-ytimeen halutaan lisätä toiminnallisuutta, on luontevin tapa lisätä lähdekooditiedostot Linux-ytimen lähdekoodihakemistoon ja uudelleen kääntää Linux-ydin. Linux-ytimen uudelleen kääntämisprosessissa on tarpeen konfiguroida vain ne tiedostot, jotka lisättiin mukaan käännösprosessiin.

Toiminnallisuutta voidaan lisätä Linux-ytimeen myös toisella tavalla, joka mahdollistaa toiminnallisuuden lisäämisen vaikka Linux-ydin olisi käynnissä. Ohjelmakoodi, joka lisätään edellä mainitulla tavalla, on ytimeen ladattava moduuli (LKM). Moduuleiksi voidaan kääntää paljon erimuotoista toiminnallisuutta, mutta lähes poikkeuksetta ne ovat: laiteajureita, tiedostojärjestelmäajureita tai järjestelmäkutsuja. Linux-ydin osaa eristää ytimeen ladattavien moduulien toiminnallisuuden riittävän hyvin, jotta ne eivät ole monimutkaisesti liitoksissa muuhun Linux-ytimen toiminnallisuuteen. /10/

### **3.3 Juuren tiedostojärjestelmä**

Juuren tiedostojärjestelmällä (root filesystem) viitataan tiedostojärjestelmään, joka on liitetty (mounted) tiedostojärjestelmän juureen, eli `/`-hakemistoon. Juuren tiedostojärjestelmälle on Linux-järjestelmässä asetettu tiettyjä vaatimuksia. Linux-järjestelmä olettaa, että juuren tiedostojärjestelmä sisältää ohjelmat ja sovellukset järjestelmän käynnistämiseksi ja palveluiden, kuten komentokehötteen ja verkkoyhteyksien, alustamiseksi. /1/

Juuren tiedostojärjestelmästä suoritetaan ensimmäisenä käynnistyskriptit hakemistoista `/etc/inittab` ja `/etc/rc.d/rcS`. Molemmat tiedostot määrittelevät, mitkä ohjelmat ja palvelut ladataan järjestelmän käynnistymisen yhteydessä. Jos esimerkiksi halutaan, että jokin ohjelma käynnistyy välittömästi käynnistymisen yhteydessä, ohjelman käynnistyskomento tulee lisätä `/etc/rc.d/rcS` -tiedostoon. Jos taas halutaan, että sarjaporttiliikennöinti mahdollistetaan heti järjestelmän käynnistyessä, tulee palvelun käynnistämiskomento lisätä `/etc/inittab` -tiedostoon.

Sulautetusta Linux-järjestelmästä riippuen juuren tiedostojärjestelmä on useimmiten staattinen kuvatiedosto (jffs2), joka on asennettu (flashed) kohdelaitteen nand-flash-muistille. Juuren tiedostojärjestelmä voi myös sijaita erillisessä järjestelmässä, ja se voidaan ladata kohdelaitteen käyttöön esimerkiksi verkkoyhteyden yli. Verkkoyhteyden yli ladattavan juuren tiedostojärjestelmän vaatimuksena on, että Linux-ytimeen on ladattu NFS-tuki. Verkkoyhteyden yli ladattava juuren tiedostojärjestelmä mahdollistaa nopean sulautetun Linux-järjestelmän kehitystyön. Muutokset voidaan välittömästi ajaa järjestelmässä, eikä järjestelmän kuvatiedostoa tarvitse asentaa uudestaan kohdelaitteeseen jokaisen muutoksen jälkeen.

## 4 Järjestelmän rakentaminen

Toimivan sulautetun Linux-järjestelmän rakentaminen vaatii työkaluja ja toimintatapoja. Tässä kappaleessa esitellään muutamia työkaluja ja tapoja, joiden hallitseminen on välttämätöntä toimivan sulautetun Linux-järjestelmän rakentamiseksi.

### 4.1 GNU-työkaluketju

GNU-työkaluketju (GNU toolchain) on kokoelma työkaluohjelmia (software tools), joita käytetään ohjelmien kääntämiseen tietyille kohdeprosessoriarkkitehtuurille. GNU-työkaluketju on peitetermi (blanket term) GNU-projektissa luodulle kokoelmalle ohjelmointityökaluja. Nämä työkalut muodostavat työkaluketjun, joita käytetään sarjallisella tavalla ohjelmien ja käyttöjärjestelmien kehittämiseen. GNU-työkaluketjulla on merkittävä rooli muun muassa Linux-ytimen, BSD-käyttöjärjestelmän ja sulautettujen järjestelmien ohjelmistojen kehittämisessä. /11/

Taulukko 2: GNU-työkaluketjuun sisältyvät työkalut.

GNU Make	Automaatiotyökalu rakentamiselle ja kääntämiselle.
GNU Compiler Collection (GCC)	Kokoelma kääntäjiä useille ohjelmointikielille.
GNU Binutils	Suite of tools including linker, assembler and other tools Kokoelma työkaluja muun muassa linkkeri, assembler ja muita.
GNU Debugger	Ohjelmakoodin vianselvitystyökalu.
GNU build system (auto tools)	Työkalut ohjelmistopakettien sovittamiseen Unixin kaltaisille järjestelmille.

GNU Make on työkalu, joka ohjaa suoritettavien (executables) ja muiden ei ohjelmakoodia sisältävien tiedostojen luomista. Make saa tiedon ohjelman rakentamiseen makefile-tiedostosta, joka listaa jokaisen ei ohjelmakooditiedoston ja kertoo, miten niitä käsitellään. Kun tietokoneohjelma kirjoitetaan, pitäisi ohjelmalle kirjoittaa makefile-tiedosto, jotta ohjelman kääntäminen ja asentaminen olisi mahdollista Make-työkalulla.

GNU Compiler Collection sisältää kääntäjät ja standardi-kirjastot C, C++, Objective-C, Fortran, Java ja Ada -ohjelmointikielillä. /12/

GNU Binutils on kokoelma binääriin käsitellytyökaluja. Tärkeimmät ovat ld-GNU linker (linkkeri) ja as-GNU assembler. /12/

GNU Debugger (yleensä GDB) on standardi vianselvitystyökalu GNU-ohjelmajärjestelmille. GDB voidaan siirtää muille Unixin kaltaisille järjestelmille. GDB toimii muun muassa Ada, C, C++, FreeBASIC ja Fortran -ohjelmointikielillä. /12/

GNU build system, joka tunnetaan paremmin nimellä Autotools, on GNU-projektin tuottama kokoelma ohjelmointityökaluja. Työkalut on suunniteltu auttamaan monenlaisten ohjelmistopakettien sovittamista Unixin kaltaisille järjestelmille. /12/

## **4.2 Ristiinkääntäminen**

Ristiinkääntäminen (cross compiling) tarkoittaa, että ohjelma tai ohjelmat käännetään yhdessä prosessoriarkkitehtuurissa toiselle. Esimerkiksi ohjelmapaketti voidaan kääntää Intel x86 –prosessoriarkkitehtuurissa ARM-prosessoriarkkitehtuurille sopivaksi. /13/

Ristiinkääntäjä (cross compiler) on kääntäjä, jolla on mahdollista luoda suoritettavaa koodia järjestelmälle, joka on jokin muu kuin missä ristiinkääntäjä itse toimii.

Ristiinkääntäjä-työkalut ovat yleisesti käytössä käännösten tekemiseen sulautetulle järjestelmälle tai useille eri järjestelmille. Ristiinkääntäjä on työkalu, jota pitää käyttää kääntämiseen järjestelmälle, jolle on epäsuotavaa tai mahdotonta kääntää järjestelmällä itsellään. Järjestelmiä, joille kääntämiseen ristiinkääntäjä on tarpeellinen, ovat kaikki ne, joilla ei itse ole riittävää suorituskykyä käännöstehtävän suorittamiseen. Ristiinkääntäjä-työkalun käyttäminen säästää myös paljon aikaa, kun ohjelmaa tai ohjelmia käännetään tehokkaalla moniytimisellä prosessorilla eikä pienitehoisella sulautetun järjestelmän prosessorilla. /13/

GNU-työkaluketjun käyttäminen mahdollistaa ohjelmistokomponenttien ristiinkääntämisen eri prosessoriarkkitehtuureille Intel x86 -työpöytäympäristössä. Ristiinkääntäminen on tarpeellista, sillä kääntäminen pienessä kohdejärjestelmässä on vaikeaa tai mahdotonta.

### **4.2.1 Linux-ytimen kääntäminen kohdejärjestelmälle**

Linux-ydin voidaan kääntää monelle eri prosessoriarkkitehtuurille. Ensimmäiseksi Linux-ydin pitää konfiguroida vastaamaan kohdelaitteen ominaisuuksia ja rajoituksia. Linux-ytimen juuressa hakemistossa ”arch/<kohde prosessoriarkkitehtuuri>/configs”,

löytyy valmiita konfiguraatioita (defconfig) tunnetuille järjestelmille. Esimerkiksi Nokia N800 Internet Tablet -päätelaitteen asetustiedosto löytyy hakemistosta ”arch/arm/configs/n800\_defconfig” kernel.org-sivuston linux-omap -haarasta. Linux-ytimen konfigurointi tapahtuu ”make menuconfig” -komennolla tai, jos järjestelmälle löytyy valmis konfiguraatiodiedosto ”make <konfiguraatiodiedoston nimi>”. Molemmat komennot luovat Linux-ytimen juureen ”.config”-tiedoston, joka määrittelee mitkä Linux-ytimen osat käännetään (compile) ja miten osat rakennetaan, joko ytimeen ladattaviksi moduuleiksi tai staattiseksi osaksi Linux-ydintä.

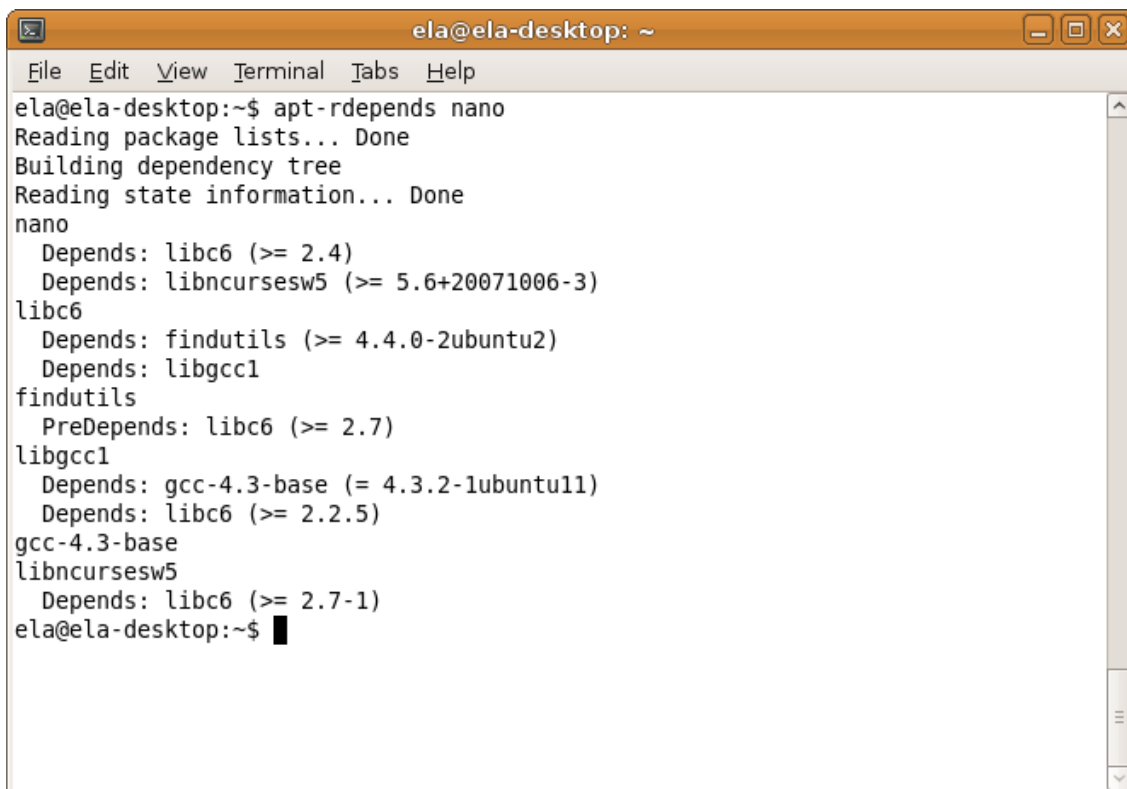
Kun Linux-ydin on konfiguroitu, pitää se vielä kääntää. Kääntäminen tapahtuu komennolla ”make”, mutta ristiinkääntämisessä, ”make”-komento tarvitsee parametrina ristiinkääntäjän sijainnin, joka lisätään komennon perään seuraavasti: ”make CROSS\_COMPILE=<ristiinkääntäjän sijainti käännöskoneessa>”. Kun ”make”-komento on suoritettu onnistuneesti ilman virheilmoituksia, löytyy Linux-ytimen zImage-binääritiedosto hakemistosta ”arch/<kohde prosessoriarkkitehtuuri>/boot”. Linux-ytimen osat, jotka rakennettiin ytimeen ladattaviksi moduuleiksi, voidaan asentaa komennolla ”make INSTALL\_MOD\_PATH=<kohdehakemisto>”.

#### 4.2.2 Ohjelmistopakettien kääntäminen kohdejärjestelmälle

Ohjelmistopakettien kääntäminen kohdejärjestelmällä on mahdollista ristiinkääntäjän avulla. Ohjelmistopakettien kääntäminen erillisellä käännöskoneella luo ideaaliset puitteet ohjelmistopakettien kääntämiselle kohdejärjestelmälle. Etuja erillisen käännöskoneen käyttämiselle ovat nopeus, hallittavuus, luotettavuus.

Normaalin kääntämisen lisäksi, ristiinkääntäminen tuo muutamia omia ongelmiaan mukaan vaikeuttamaan käännösprosessia. Ongelmia aiheutuu jos käännösjärjestelmä ei osaa riittävällä tarkkuudella emuloida kohdeympäristöä, kyseinen ongelma tulee vastaan kun ohjelmistopakettien konfigurointiin liittyy ehtoja, jotka määrittelevät erittäin tarkasti kohdejärjestelmään liittyviä muistinhallinnallisia asioita, kuten esimerkiksi tavujärjestyksen (endianness). Ohjelmistopakettien konfigurointia vaikeuttaa myös se, että käännöskoneen käännösjärjestelmä ei ole täysin identtinen kohdejärjestelmän kanssa. Tämä saattaa aiheuttaa tilanteen, että jokin paketti on kohdejärjestelmässä, mutta ei käännösympäristössä, puuttuva paketti saattaa estää jonkin ohjelmistopakettien kääntymisen.

Joillakin ohjelmistopaketeilla on riippuvuuksia toisiin ohjelmistopaketteihin ja tämä asettaa haasteita käännösympäristölle. Ohjelmistopakettien välisten riippuvuuksien takia, ohjelmistopaketit pitää kääntää ja rakentaa niiden riippuvuusketjun (dependency chain) mukaan. Käännösprosessin pitää tapahtua siten, että riippuvuusketjun alimmat osat rakennetaan ensimmäiseksi, jotta paketit jotka riippuvat niistä pystytään rakentamaan. Käytännössä käännetyt paketit lisätään osaksi ristiinkääntäjää, jolloin ohjelmistopaketit ovat käännösympäristön käytettävissä. Kuviossa 5 on esitetty nano-tekstieditorin vaatimien ohjelmistopakettien väliset riippuvuussuhteet.



```

ela@ela-desktop: ~
File Edit View Terminal Tabs Help
ela@ela-desktop:~$ apt-rdepends nano
Reading package lists... Done
Building dependency tree
Reading state information... Done
nano
  Depends: libc6 (>= 2.4)
  Depends: libncursesw5 (>= 5.6+20071006-3)
libc6
  Depends: findutils (>= 4.4.0-2ubuntu2)
  Depends: libgcc1
findutils
  PreDepends: libc6 (>= 2.7)
libgcc1
  Depends: gcc-4.3-base (= 4.3.2-1ubuntu11)
  Depends: libc6 (>= 2.2.5)
gcc-4.3-base
libncursesw5
  Depends: libc6 (>= 2.7-1)
ela@ela-desktop:~$ █

```

Kuvio 5. Nano-tekstieditorin vaatimien ohjelmistopakettien väliset riippuvuussuhteet.

Ennen kuin nano-tekstieditori voidaan ristiinkääntää halutulle kohdeproessoriarkkitehtuurille, pitää ensin ristiinkääntää kaikki nano-tekstieditorin tarvitsemat ohjelmistopaketit.



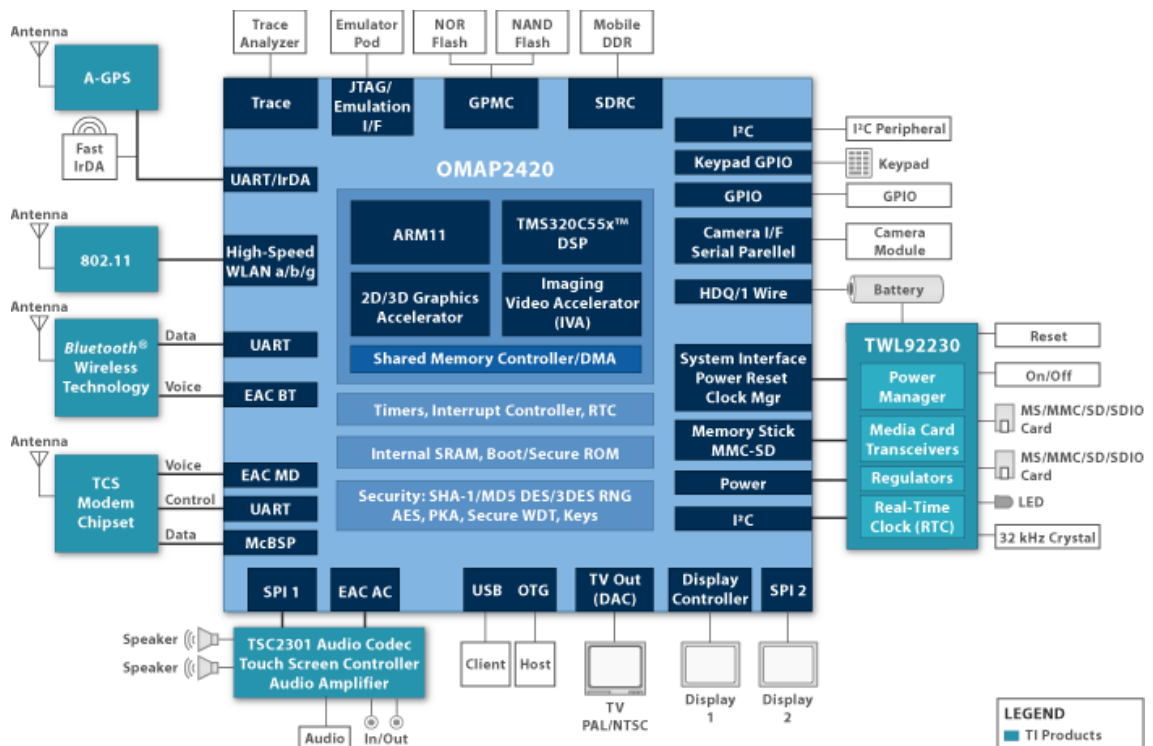
Ohjelmistopakettien välisten riippuvuussuhteiden hallinta käännösympäristössä, ilman hallintatyökaluja on erittäin haasteellista ohjelmistopakettien määrästä ja niiden riippuvuusketjun laajuudesta riippuen. Tästä syystä ohjelmistopakettien hallintaan Linux-järjestelmissä on luotu kehittyneitä ohjelmistopakettien hallintajärjestelmiä. Kaikkein merkittävin hallintajärjestelmä on apt (advanced package tool), joka on käytössä Debian Linux -pohjaisissa käyttöjärjestelmissä.

## 5 Sulautetun Linux-järjestelmän rakentaminen Poky-työkalulla

Tässä kappaleessa esitellään toteutus sulautetun Linux-järjestelmän rakentamisesta Nokia N800 Internet Tablet -päätelaitteelle Poky platform builder -työkalulla

### 5.1 Nokia N800 Internet Tablet

Nokia N800 Internet Tablet on Nokia Oyj:n valmistama mobiilipäätelaitte, joka perustuu Texas Instrumentsin OMAP 2420 -alustaan ja ARM-proessoriarkkitehtuuriin. Kuviossa 6 on esitettyä Texas Instrumentsin OMAP 2420 -alustan lohkokkaavio.



Kuvio 6. Texas Instrumentsin OMAP 2420 -alustan lohkokkaavio /14/

Texas Instrumentsin OMAP 2420 -alusta on tarkoitettu suuren volyymin langattomien laitteiden valmistajille. OMAP 2420 -alustasta löytyy tuki kaikille yleisesti käytössä oleville langattomille tiedonsiirtotekniikoille kuten: GPS, WLAN, Bluetooth ja Texas Instrumentsin TCS-alijärjestelmä, johon voidaan kytkeä GSM/3G-lähetinvastaanotin. Alustaan sisältyy myös TWL92230-apupiiri, josta löytyy älykkäät virranhallintaominaisuudet. Kokonaisuudessaan OMAP 2420 -alusta on minikoossa oleva tietokone, josta löytyy ominaisuuksia erilaisten langattomien laitteiden tarpeisiin. /14/

Vakiona Nokia N800 -pöytälaiteessa on Debian Linux pohjainen Maemo-käyttöjärjestelmä. Järjestelmä on kuitenkin mahdollista korvata toisella Linux-pohjaisella käyttöjärjestelmällä, josta löytyy valmiudet käyttää Nokia N800 -pöytälaitteen OMAP 2420 -pohjaista laitteistoa. Laitteistotuen saaminen on mahdollista sillä Maemossa käytetty Linux-ydin on GPL-lisenssin alainen ja julkisesti saatavilla. Taulukossa 1 on esitetty Nokia N800 -pöytälaitteen teknisiä ominaisuuksia.

Taulukko 1. Nokia N800 -pöytälaitteen tekniset ominaisuudet.

Proessori:	OMAP2420 mikroprosessori 400 MHz nopeudella.
Muisti:	128 MB RAM-muistia ja 256 MiB flash-muistia.
Yhteydet:	IEEE 802.11 b/g, Bluetooth 2.0, and USB 2.0 OTG high-speed.
Näyttö:	800 × 480 resolution, 4.1 in. diagonal, 225 pixels/in., 65536 colors (16-bit)
Syöttö:	Resistiivinen kosketusnäyttö.
Laajennettavuus:	Yksi sisäinen ja yksi ulkoinen MMC-muistikortti paikka.

### 5.1.1 Nokia bootloader

Nokia N800 Internet Tablet käyttää Nokia Oyj:n kehittämää suljettua nolo-alkulatausohjelmaa. nolon lähdekoodeja tai dokumentaatiota ei ole julkisesti saatavilla, mutta nolon toiminta tunnetaan julkisesti yleisellä tasolla. nolo lataa Linux-ytimen sille osoitetusta paikasta.

### 5.1.2 Nokia flasher

Nokia flasher -ohjelma on komentorivityökalu, jolla voidaan asentaa ohjelmakomponentteja Nokia N800 -pöytälaitteeseen. Flasher-työkalun avulla pöytälaitteeseen voidaan asentaa alkulatausohjelma, Linux-ydin ja juuren tiedostojärjestelmä. Kuviossa 7 on esitetty flasher-työkalun käyttövalinnat.

```

ela@ela-desktop: ~/Desktop/flash
File Edit View Terminal Tabs Help
flasher v0.8.1 (Jan 5 2007)

Nothing to do!
Usage: flasher [OPTIONS]
Options:
  --fiasco, -F <arg>      Location of a FIASCO image
  --kernel, -k <arg>     Location of kernel image
  --initfs, -n <arg>     Location of initfs image
  --rootfs, -r <arg>     Location of root JFFS2 image
  --xloader, -x <arg>    Location of X-Loader image
  --secondary, -s <arg>  Location of NOLo secondary bootloader image
  --2nd, -2 <arg>        Location of NOLo cold flasher ("2nd") image
  --unpack, -u [arg]     Unpack a FIASCO image
  --flash, -f            Load and flash all supplied images
  --load, -l            Only load all supplied images
  --boot, -b [arg]      Boot the kernel with optional cmdline
  --reboot, -R          Reboot the board (e.g. after flashing NOLo)
  --read-board-id, -i   Print out the board type
  --serial-port, -S <arg> Serial port used for cold flashing
  --initialize-port, -I <arg> Only initialize the serial port
  --cold-flash, -c      "Cold flash" the device
  --hw-revision, -h <arg> Specify product code and HW revision
  --enable-rd-mode      Enable R&D mode on the device
  --disable-rd-mode     Disable R&D mode on the device
  --set-rd-flags [arg]  Set R&D mode flags on the device
  --clear-rd-flags [arg] Clear R&D mode flags on the device
  --query-rd-mode       Query the device R&D mode status and flags
  --set-root-device <arg> Set the default root device
  --query-root-device   Query the default root device
  --enable-usb-host-mode Set the device into USB host mode
  --disable-usb-host-mode Set the device into USB peripheral mode
  --flash-only <arg>   Flash only certain components

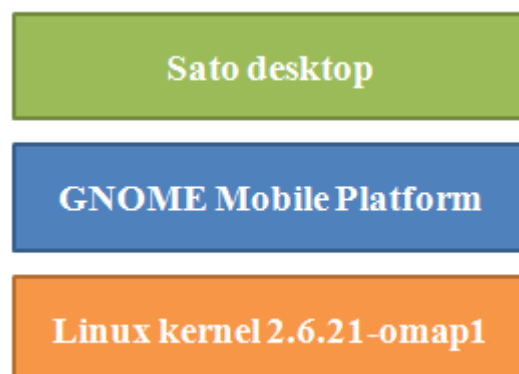
ela@ela-desktop:~/Desktop/flash$ █

```

Kuvio 7. Nokia flasher -työkalun käyttövalinnat.

## 5.2 Järjestelmän ohjelmistokomponentit

Järjestelmään valitut ohjelmistokomponentit on huolellisesti valittu Nokia N800 -päätelaitteen kanssa yhteensopiviksi. Esimerkki sulautettu Linux-järjestelmä rakentuu kolmesta suuresta ohjelmakomponentista Linux-ytimestä, GNOME Mobile Platform -alustasta ja Sato-työpöytäympäristöstä. Kuviossa 8 on esitetty esimerkkijärjestelmän kolme suurinta ohjelmistokomponenttia.



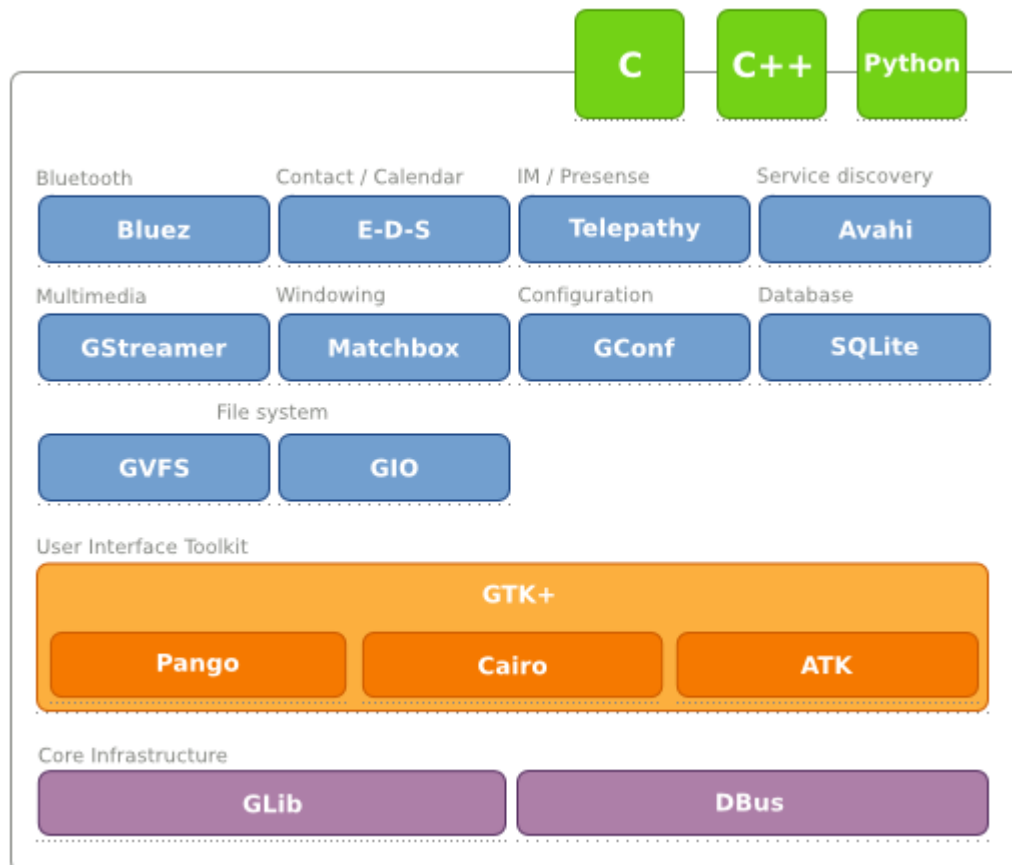
Kuvio 8: Esimerkkijärjestelmän kolme suurinta ohjelmistokomponenttia.

## 5.2.1 Linux-ydin

Linux-ytimestä käytetään versiota 2.6.21-omap1, johon lisätään Nokia Oyj:n julkaisema korjauspaketti ennen Linux-ytimen konfigurointia ja kääntämistä.

## 5.2.2 GNOME Mobile Platform

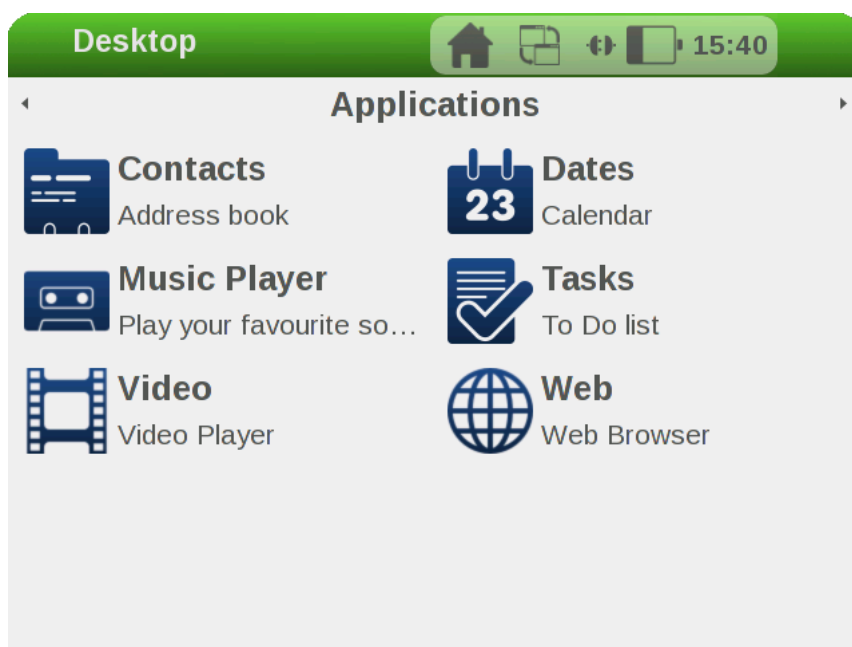
GNOME Mobile Platform -alusta on kokoelma vapaita ohjelmistokomponentteja. Alustaan sisältyy käyttöliittymäkirjastot ja keskiohjelmia (middleware). Yhdessä nämä ohjelmistokomponentit muodostavat riittävän tehokkaissa sulautetuissa järjestelmissä käytettävän ohjelmistokehitysympäristön. Alustaan pohjautuvia sovelluksia voidaan kehittää, joko C, C++ tai Python -ohjelmointikielillä. Kuviossa 9 on esitetty GNOME Mobile Platform -alustan tärkeimmät ohjelmistokomponentit ja käytettävät ohjelmointikielet.



Kuvio 9. GNOME Mobile Platform -alustan ohjelmistokomponentit ja käytettävät ohjelmointikielet.

### 5.2.3 Sato desktop

Sato on GNOME Mobile Platform -alustaan pohjautuva työpöytäympäristö. Sato-työpöytäympäristö on suunniteltu keskisuuren näyttöresoluution omaaville kosketusnäyttöisille laitteille, kuten matkapuhelimille, kämmentietokoneille, pienikokoisille kannettaville tietokoneille ja muille. Sato-työpöytäympäristöä kehitettäessä on otettu lähtökohdaksi tehokkuus ja nopeus. Tästä syystä Sato toimii hyvin esimerkiksi sulautetuissa järjestelmissä, joissa on riittävästi tehoa. Sato toimii hyvin millä tahansa alustalla, joka käyttää GNOME Mobile Platform -alustaa. Kuviossa 10 on esitetty kuvakaappaus GNOME Mobile Platform -alustaan pohjautuvasta Sato-työpöytäympäristöstä.



Kuvio 10. GNOME Mobile Platform -alustaan pohjautuva Sato-työpöytäympäristö. /8/

### 5.3 Poky platform builder

Poky on avoimenlähdekoodin Linux-järjestelmän rakennustyökalu, jolla on mahdollista luoda kokonainen sulautettu Linux-järjestelmä. Poky:lla voidaan suunnitella, kehittää, rakentaa ja testata Linux-ytimeen ja GNOME Mobile Platform -alustaan pohjautuvia Intel x86 ja ARM -järjestelmiä. Poky on avoimeenlähdekoodin perustuva vapaaohjelmisto, jota tukee Intel. Poky on pääasiassa Linux-järjestelmän rakennustyökalu, joka rakentaa täyden juuren tiedostojärjestelmän valittujen ohjelmistopakettien perusteella. Poky:ssa löytyy valmiina tuki muutamille laitteille kuten Nokia N800 Internet Tabletille ja Sharp Zaurus -kämmentietokoneelle. /8/

Pokyssa on yli 500 valmiiksi konfiguroitua ohjelmistopakettia, jotka voidaan rakentaa haluttuun järjestelmään. Poky-järjestelmään on mahdollista lisätä omia ohjelmistopaketteja ja täten käyttää Pokya omassa ohjelmistonkehitys prosessissa. Poky käyttää järjestelmän rakentamiseen Bitbake-sovellusta, jossa on automaattinen ohjelmistopakettien riippuvuusketjun hallinta. Bitbakella voidaan luoda juuren tiedostojärjestelmä kuvauksia (image), Linux-ytimiä sekä yksittäisiä ohjelmistopaketteja ipk, deb ja tgz -muodoissa. Lisäksi Bitbake kykenee rakentamaan luodusta järjestelmästä täyden SDK:n, joka sisältää ristiinkääntäjän ja tarpeelliset työkalut. Järjestelmän rakentamista varten Poky tarvitsee laitekonfiguraatitiedoston (machine configuration) ja käännöstehtävän (build task). /8/

### 5.3.1 Laitekonfiguraatio

Poky-järjestelmä vaatii järjestelmän rakentamista varten kuvauksen päätelaitteen teknisistä ominaisuuksista. Kuvaus järjestelmästä toimitetaan laitekonfiguraatitiedostona. Laitekonfiguraatitiedostossa tarvitsee kuvata vähintään seuraavat kohdat: prosessoriarkkitehtuuri, Linux-ydin ja laitteen ominaisuudet (machine features). Vaikka laitekonfiguraatio-tiedostossa tarvitsee kuvata vain edellä mainitut seikat, on kuitenkin erittäin suotavaa kuvata päätelaitteen ominaisuuksia niin paljon kuin suinkin mahdollista. Lisäksi voidaan kuvata muun muassa seuraavia ominaisuuksia: sarjakonsoli (serial console), Linux-ytimen kuvatiedosto (zImage), juuren tiedostojärjestelmän kuvatiedosto (tar.gz tai jffs2), graafisen käyttöliittymän (GUI) tyyppi sekä muita ominaisuuksia. Laitekonfiguraatitiedostossa listattujen ominaisuuksien perusteella rakennetaan päätelaitteelle soveltuva järjestelmä. /15/

Poky-järjestelmässä on valmiina laitekonfiguraatio-tiedosto Nokia N800 -päätelaitteelle. Valmista laitekonfiguraatio-tiedostoa voidaan käyttää esimerkkijärjestelmän rakentamiseen. Kuviossa 11 on esitetty Poky-järjestelmän laitekonfiguraatitiedosto Nokia N800 -päätelaitteelle.

```

#@TYPE: Machine
#@NAME: Nokia 770 internet tablet
#@DESCRIPTION: Machine configuration for the Nokia 800
TARGET_ARCH = "arm"
PACKAGE_EXTRA_ARCHS = "armv4 armv4t armv5te armv6"

PREFERRED_PROVIDER_virtual/xserver = "xserver-kdrive-xomap"
XSERVER = "xserver-kdrive-xomap"

GUI_MACHINE_CLASS = "bigscreen"

# Use tune-arm926 per default. Machine independent feeds should be built with tune-strongarm.
include conf/machine/include/tune-arm926ejs.inc

ROOT_FLASH_SIZE = "174"
EXTRA_IMAGECMD_jffs2_nokia800 = "--pad --little-endian --eraseblock=0x20000 -n"

IMAGE_FSTYPES ?= "jffs2"

SERIAL_CONSOLE = "115200 ttyS0"

PREFERRED_PROVIDER_virtual/kernel = "linux-nokia800"

#use this if you are using the nokia initfs
ROOTFS_POSTPROCESS_COMMAND += " remove_init_link; "

MACHINE_FEATURES = "kernel26 apm alsa bluetooth usb gadget usbhost vfat"
MACHINE_EXTRA_RDEPENDS = "cx3110x usbinit"
MACHINE_ESSENTIAL_EXTRA_RDEPENDS = "nokia770-init"

KERNEL_IMAGE_MAXSIZE = "2097152"

```

Kuvio 11. Poky-järjestelmän laitekonfiguraatitiedosto Nokia N800 -pöytälaiteelle.

Laitekonfiguraatitiedosto koostuu shell-muuttujista, joihin syötetään pöytälaiteelle soveltuvat parametrit.

### 5.3.2 Käännöstehtävä

Kaikki ohjelmistopakettit, joista järjestelmä rakentuu, pitää kuvata Pokylle.

Ohjelmistopakettit kuvataan Pokylle käännöstehtävätiedostossa (build task).

Käännöstehtävätiedosto sisältää listauksen kaikista ohjelmistopaketeista, jotka aiotaan sisällyttää rakennettavaan järjestelmään.

Käännöstehtävässä kuvatuilla ohjelmistopaketeilla pitää jokaisella olla oma käännösresepti (build recipe) -tiedosto. Käännösreseptitiedostossa on kuvattu järjestelmänrakennus työkalu Bitbake, miten kyseinen ohjelmistopaketti pitää rakentaa ja miten järjestelmään asennettava ohjelmistopaketti muodostetaan. Käännösreseptitiedosto on kirjoitettu käyttäen Python-ohjelmointikieltä ja Shell-



ohjelmointiympäristöä. Kuviossa 12 on esitetty usbutils-ohjelmistopakettin käännösreseptitiedosto.

```
DESCRIPTION = "Host side USB console utilities."
SECTION = "base"
DEPENDS += "libusb"
LICENSE = "GPL"
PRIORITY = "optional"
DEFAULT_PREFERENCE = "1"

SRC_URI = "${SOURCEFORGE_MIRROR}/linux-usb/usbutils-${PV}.tar.gz"
S = "${WORKDIR}/usbutils-${PV}"

inherit autotools

EXTRA_OECONF = "--program-prefix="
sbindir = "/sbin"
bindir = "/bin"

FILES_${PN} += "${datadir}/usb*"

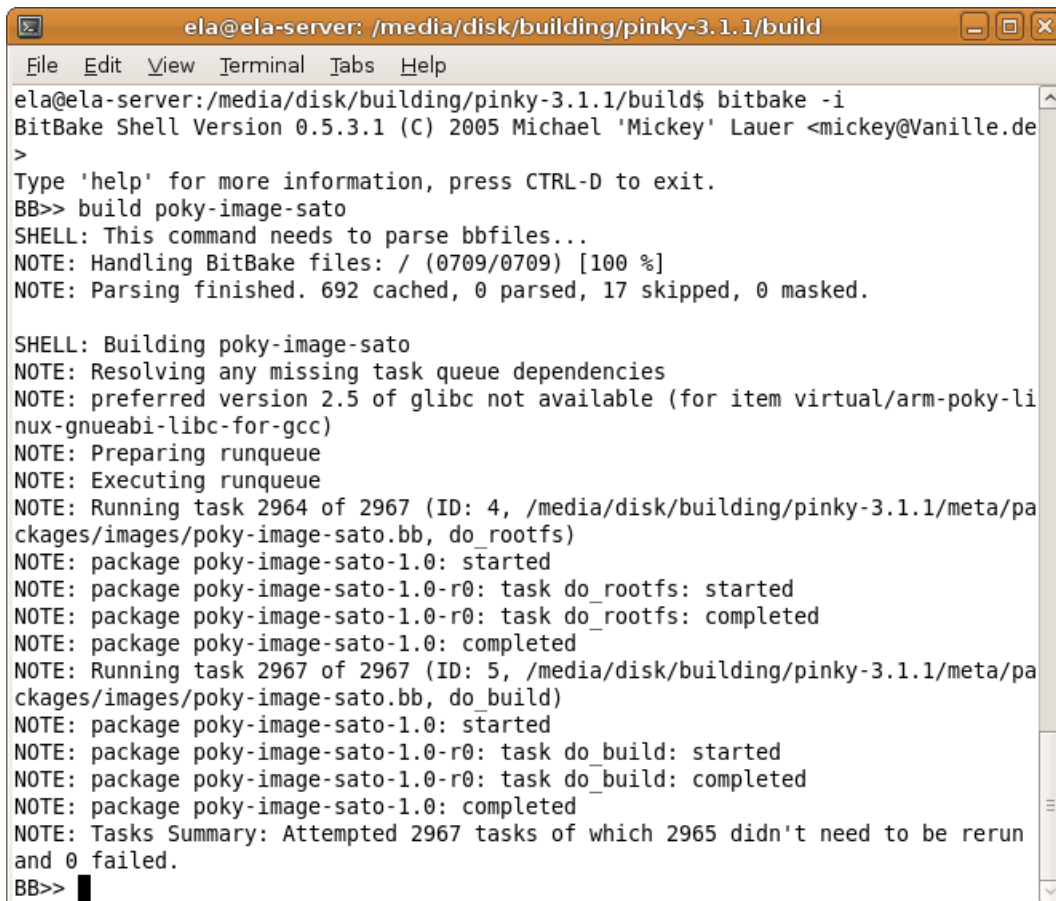
do_configure_prepend() {
    rm -rf ${S}/libusb
}
```

Kuvio 12. Usbutils-ohjelmistopakettin käännösreseptitiedosto.

Käännösreseptitiedostossa ohjelmistopakettista kuvataan perustietoja, kuten lähdekoodivaraston osoite, kääntäjän tarvitsemat erikoisparametrit, asennuspakettiin sisältyvät tiedostot ja esiasetus toimenpiteet. Käännösreseptien kirjoittaminen vaatii syvällistä tuntemusta ohjelmistopakettien rakenteesta ja GNU build system -järjestelmän käytöstä. Pokyn dokumentaatiosta on kuitenkin saatavilla esimerkki käännösreseptitiedostoja erilaisille ohjelmistopaketeille.

## 5.4 Järjestelmän rakentaminen

Esimerkkijärjestelmän käännöstehtävätiedostossa listattujen ohjelmistopakettien käännösreseptit on sisällytetty Poky-järjestelmään. Kuviossa 11 esitetty laitekonfiguraatitiedosto on valittu rakennettavan järjestelmän kohdelaitteeksi (target device). Järjestelmän rakentamista varten avataan Bitbake-sovellus, jolle on määritelty Poky-järjestelmän sijainti. Järjestelmän rakentamiseen kuluu Intel Atom -tuplaydinprosessorilla noin. 5 tuntia. Kuviossa 13 on esitetty Bitbake-sovelluksen ilmoitus esimerkkijärjestelmän onnistuneesta rakentamisesta Nokia N800 -päätelaitteelle.



```

ela@ela-server: /media/disk/building/pinky-3.1.1/build
File Edit View Terminal Tabs Help
ela@ela-server:/media/disk/building/pinky-3.1.1/build$ bitbake -i
BitBake Shell Version 0.5.3.1 (C) 2005 Michael 'Mickey' Lauer <mickey@Vanille.de>
>
Type 'help' for more information, press CTRL-D to exit.
BB>> build poky-image-sato
SHELL: This command needs to parse bbfiles...
NOTE: Handling BitBake files: / (0709/0709) [100 %]
NOTE: Parsing finished. 692 cached, 0 parsed, 17 skipped, 0 masked.

SHELL: Building poky-image-sato
NOTE: Resolving any missing task queue dependencies
NOTE: preferred version 2.5 of glibc not available (for item virtual/arm-poky-linux-gnueabi-libc-for-gcc)
NOTE: Preparing runqueue
NOTE: Executing runqueue
NOTE: Running task 2964 of 2967 (ID: 4, /media/disk/building/pinky-3.1.1/meta/packages/images/poky-image-sato.bb, do_rootfs)
NOTE: package poky-image-sato-1.0: started
NOTE: package poky-image-sato-1.0-r0: task do_rootfs: started
NOTE: package poky-image-sato-1.0-r0: task do_rootfs: completed
NOTE: package poky-image-sato-1.0: completed
NOTE: Running task 2967 of 2967 (ID: 5, /media/disk/building/pinky-3.1.1/meta/packages/images/poky-image-sato.bb, do_build)
NOTE: package poky-image-sato-1.0: started
NOTE: package poky-image-sato-1.0-r0: task do_build: started
NOTE: package poky-image-sato-1.0-r0: task do_build: completed
NOTE: package poky-image-sato-1.0: completed
NOTE: Tasks Summary: Attempted 2967 tasks of which 2965 didn't need to be rerun and 0 failed.
BB>> █

```

Kuvio 13: Bitbake-sovelluksen ilmoitus onnistuneesta järjestelmän rakentamisesta.

Bitbake-sovelluksen onnistuneen järjestelmän rakentamisen tuloksena saadaan juuren tiedostojärjestelmästä poky-image-sato-nokia800-20081023044859.rootfs.jffs2 -kuvatiedosto ja Linux-ytimeistä zImage-nokia800.bin-binääritiedosto. Molemmat tiedostot ovat Nokia flasher -työkalun tukemia formaatteja.

## 5.5 Järjestelmän asentaminen päätelaitteeseen

Käännösprosessin tuloksena saadut jffs2- ja zImage-tiedostot voidaan asentaa Nokia N800 -pätelaitteeseen käyttäen Nokia flasher -työkalua.

Nokia flasher -työkalu ottaa juuren tiedostojärjestelmän (jffs2) ja Linux-ytimen (zImage) sijainnin parametrina. Kuviossa 13 ja 14 on esitetty Nokia flasher -työkalu asentamassa Linux-ydin ja juuren tiedostojärjestelmä Nokia N800 -pätelaitteeseen.

```

ela@ela-desktop: ~/Desktop/flash
File Edit View Terminal Tabs Help
ela@ela-desktop:~/Desktop/flash$ sudo ./flasher-3.0.amd64 -f -k zImage-nokia800.bin -R
flasher v0.8.1 (Jan 5 2007)

USB device found found at bus 005, device address 008
Found device RX-34, hardware revision 1301
NOLo version 1.1.16
Version of 'sw-release': <no version>
Sending kernel image (1515 kB)...
100% (1515 of 1515 kB, avg. 13532 kB/s)
Flashing kernel... done.
ela@ela-desktop:~/Desktop/flash$ █

```

Kuvio 13. Nokia flasher -työkalu asentamassa Linux-ytimen zImage-tiedostoa Nokia N800 -päätelaitteeseen.

```

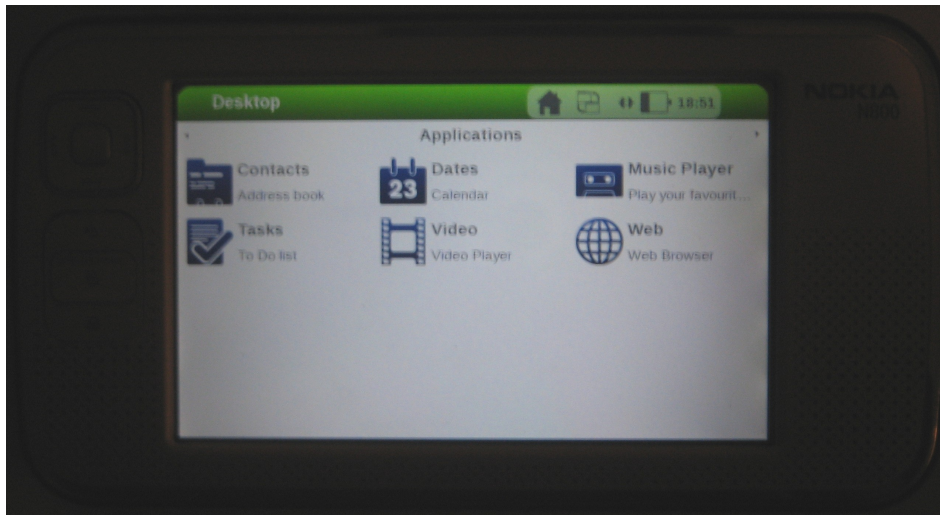
ela@ela-desktop: ~/Desktop/flash
File Edit View Terminal Tabs Help
ela@ela-desktop:~/Desktop/flash$ sudo ./flasher-3.0.amd64 -f -r poky-image-sato-nokia800
-20081023044859.rootfs.jffs2 -R
flasher v0.8.1 (Jan 5 2007)

USB device found found at bus 005, device address 010
Found device RX-34, hardware revision 1301
NOLo version 1.1.16
Version of 'sw-release': <no version>
Sending and flashing rootfs image (39040 kB)...
100% (39040 of 39040 kB, avg. 6857 kB/s)
Finishing flashing... done
ela@ela-desktop:~/Desktop/flash$ □

```

Kuvio 14. Nokia flasher -työkalu asentamassa jffs2 tiedostojärjestelmää Nokia N800 -päätelaitteeseen.

Onnistuneen asennus operaation jälkeen, Nokia N800 -päätelaitte käynnistyy ja Sato-työpöytäympäristö ilmestyy. Kuviossa 15 on esitetty Nokia N800 -päätelaitte suorittamassa rakennettua esimerkkijärjestelmää.



Kuvio 15. Rakennettu esimerkkijärjestelmä Nokia N800 -pöytälaiteessa

## 6 Yhteenveto

Työn tavoitteena oli toteuttaa sulautettu Linux-järjestelmä Nokia N800 -pöytälaitteelle. Työ tehtiin käyttäen Poky platform builder -työkalua. Järjestelmän rakentaminen Pokylla on itsessään suoraviivainen toimenpide. Ongelmia kuitenkin syntyy kun järjestelmän ohjelmistopakettit eivät käänny oikealla tavalla. Epäonnistuneen käännösprosessin tutkiminen ja vian selvittäminen on vaativa prosessi, joka vaatii ymmärrystä käännettävästä järjestelmästä, sekä Poky-työkalun sisäisistä toiminnoista.

Sulautetun Linux-järjestelmän rakentaminen on monimutkainen tehtävä, joka vaatii paljon tietoa sulautetun Linux-järjestelmän eri komponenteista ja siitä miten eri komponentit toimivat. Vaativinta järjestelmän rakentamisessa ovat virhetilanteet ja niiden selvittäminen.

Tutkintotyön tekeminen lisäsi huomattavasti tietouttani sulautetuista Linux-järjestelmistä ja niiden rakentamisesta. Sulautettujen Linux-järjestelmien ymmärtäminen tulee olemaan tulevaisuudessa erittäin tärkeää, kun Linux-käyttöjärjestelmän osuus tulee edelleen kasvamaan sulautettujen järjestelmien pääkäyttöjärjestelmänä ja seurauksena syntyy kysyntää Linux-osaajille. Toivon, että voin tulevaisuudessa jatkaa työskentelyä sulautettujen Linux-järjestelmien parissa.

## Lähteet

### Painetut lähteet

- 1 Hallinan, Christopher 2006. Embedded Linux Primer: A Practical, Real-World Approach.
- 2 Yaghmour, Karim, Masters, Jon, Ben-Yossef, Gilad & Gerum, Phillippe 2008. Building Embedded Linux Systems, Second Edition.

### Sähköiset lähteet

- 3 The Linux Kernel Archives [www-sivu]. [viitattu 1.12.2008] Saatavissa: <http://www.kernel.org/>
- 4 Linux Operating System Download [www-sivu]. [viitattu 2.12.2008] Saatavissa: [http://www.arm.com/products/os/linux\\_download.html](http://www.arm.com/products/os/linux_download.html)
- 5 Linux still top embedded OS [www-sivu]. [viitattu 3.12.2008] Saatavissa: <http://www.linuxdevices.com/news/NS4920597981.html>
- 6 Linux system development on an embedded device [www-sivu]. [viitattu 4.12.2008] Saatavissa: <http://www.ibm.com/developerworks/library/l-embdev.html>
- 7 Anatomy of Linux loadable kernel modules [www-sivu]. [viitattu 5.12.2008] Saatavissa: <http://www.ibm.com/developerworks/linux/library/l-lkm/index.html>
- 8 Poky Platform Builder – About Poky [www-sivu]. [viitattu 6.12.2008] Saatavissa: <http://www.pokylinux.org/about/>
- 9 GNU Operating System – Licenses [www-sivu]. [viitattu 6.12.2008] Saatavissa: <http://www.gnu.org/licenses/licenses.html>
- 10 The Linux Documentation Project - Linux Loadable Kernel Module HOWTO [www-sivu]. [viitattu 7.12.2008] Saatavissa: <http://tldp.org/HOWTO/Module-HOWTO/x73.html>
- 11 Handhelds - GNU Toolchain Projects [www-sivu]. [viitattu 7.12.2008] Saatavissa: <http://www.handhelds.org/projects/toolchain/index.html>
- 12 Software – GNU Project – Free Software Foundation [www-sivu]. [viitattu 7.12.2008] Saatavissa: <http://www.gnu.org/software/>
- 13 Introduction to cross-compiling for Linux [www-sivu]. [viitattu 7.12] Saatavissa: <http://www.landley.net/writing/docs/cross-compiling.html>
- 14 High-Performance – OMAP 2420 [www-sivu]. [viitattu 7.12] Saatavissa: <http://focus.ti.com/general/docs/>
- 15 Poky Handbook [www-sivu]. [viitattu 7.12] Saatavissa: <http://www.pokylinux.org/doc/poky-handbook.html>