

BLACK BIBLE



Linux From
Scratch



TIIVISTELMÄ

Tekijä(t) ja luokkatunnus: Waris Milla-Roosa, TTK15SD; Joona Tolonen

Työn nimi: Linux From Scratch

Kustantaja: Kajaanin Ammattikorkeakoulu

Sivumäärä: 421

Julkaisuvuosi: 2017

ISBN: 978-952-7219-18-8

Asiasanat: Linux, LFS

Sisällys

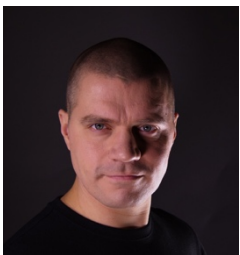
1 JOHDANTO	4
2 ASENNUKSELAISEN YMPÄRISTÖN LUOMINEN	6
2.1 Isäntäkoneen vaatimukset	6
2.2 Uuden levyjaon luominen	11
2.3 Tiedostojärjestelmän luominen	20
2.4 LFS-muuttujan asettaminen	22
2.5 Uuden partition liittäminen	23
2.6 Paketit ja korjaustiedostot	26
2.7 Asennusympäristön viimeistely	30
2.8 Tilapäisen järjestelmän rakentaminen	36
2.9 Järjestelmän perusohjelmien asentaminen	126
2.10 Muutama sana kirjastoista	127
2.11 Kernelin virtuaalisten tiedostojärjestelmien valmistelemine	128
2.12 Chroot-ympäristöön siirtyminen	133
2.13 Hakemistojen luominen	135
2.14 Tarpeellisten tiedostojen ja symbolisten linkkien luominen	138
2.15 Pakettien asentaminen	144
2.16 Debuggaussymbolit	345
2.17 Stripping	346
2.18 Siistitään	348
3 JÄRJESTELMÄN KONFIGUROIMINEN	350
3.1 System V	351
3.2 LFS-Bootscrip-20150222	352
3.3 Laitteiden ja moduulien käsittely	354
3.4 Laitteiden hallinta	362
3.5 Verkon konfigurointi	369
3.6 System V Bootscript - käyttö ja konfigurointi	375
3.7 Bash Shell Startup -tiedostot	394
3.8 /etc/inputrc -tiedoston luominen	398
3.9 /etc/shells -tiedoston luominen	401

4 LFS-JÄRJESTELMÄN TEKEMINEN BOOTATTAVAKSI.....	402
4.1 /etc/fstab -tiedoston luominen	403
4.2 Linux-4.9.9	406
4.3 GRUBin käyttäminen käynnistysprosessin aloittamiseksi	414
5 LOPPU	418
5.1 Ilmoittaudu LFS-käyttäjäksi	419
5.2 Järjestelmän uudelleen käynnistäminen	420
5.3 Mitä seuraavaksi?	423
6 SANASTOA.....	424
7 PAKETTILista	430
8 LIITTEET	449

TEKIJÄT



Milla-Roosa Waris opiskelee kolmatta vuottaan Kajaanin ammattikorkeakoulussa tietojenkäsittelyn koulutusohjelmassa Datacenter -linjalla. Hän on perehtynyt laajasti Linux/Unix -järjestelmiin ja suorittanut useita sertifikaatteja, mm. Professional Certificate in Java and Android Foundation (Hong Kong University of Technology and Science) sekä Apple Certified Associate - Mac Integration 10.11.



Joona Tolonen työskentelee Lehtorina Kajaanin ammattikorkeakoulussa. Pääsääntöisesti hän opettaa ohjelmoinnin perusteita, johtamista ja Linux/Unix -kursseja. Hän pyrkii opettamaan aihealueita opiskelijoiden kanssa yhdessä tekemällä ja käytäntöön soveltamalla. Onnistuneimpia oppimiskokemuksia saatetaan myös toisten saataville, josta esimerkkinä toimii tämä kirja.

KIRJASTA

Tämä kirja pyrkii kattamaan kaikki Linux-käyttöjärjestelmän alusta asti rakentamisen vaiheet ja lopputuloksena käyttäjällä on käytettävissään oma Linux-käyttöjärjestelmä. Kirja on kirjoitettu parhaan taitomme mukaan ja siinä on yritetty ottaa huomioon myös seikkoja, jotka voivat mennä pieleen. Valitettavasti kaikkia ongelmatilanteita emme ole tässä teoksessa voineet kattaa. Jos huomaat teoksessa selviä asiavirheitä tai muuten haluat olla yhteydessä kirjan osalta – se onnistuu parhaiten lähettämällä sähköpostia osoitteella: joona.tolonen@kamk.fi.

1 JOHDANTO

LFS eli Linux From Scratch on projekti, joka tarjoaa askeleittain ohjeet oman Linux-järjestelmän tekemiseen. Alkuperäinen englanninkielinen LFS-kirja löytyy täältä: <http://www.linuxfromscratch.org/lfs/>. Kyseinen kirja toimii lähtökohtana tälle teokselle, joten siihen voi olla myös hyvä tutustua. Välttämätöntä se ei kuitenkaan ole.

Tämä opas on kirjoitettu LFS-kirjan version 8.0 pohjalta.

LFS:n rakentaminen ei vaadi muuta kuin Linuxin peruskomentojen osaamisen ja motivaatiota.

LFS on erittäin hyvä projekti oppimisen kannalta. Linux-järjestelmän toiminnan kannalta on tärkeää tietää, mitä mikäkin paketti tekee ja miksi käyttäjä tai järjestelmä tarvitsee sitä.

LFS:n voi tehdä joko fyysisellä koneella tai virtuaalikoneella. Järjestelmää ei saisi sammuttaa LFS:n rakennuksen aikana. Muuten joitain toimintoja joutuu suorittamaan uudelleen käynnistyksen jälkeen. Jos järjestelmän sammuttaminen jostain syystä on kuitenkin tarpeen, seuraavasta linkistä voi katsoa apuja siihen, miten uudelleenkäynnistyksen jälkeen edetään:

<http://www.linuxfromscratch.org/hints/downloads/files/stages-stop-and-re-sume.txt>.

Ohjeet ovat vuodelta 2006 ja noudattavat LFS-versiota 6.1.1-3, mutta pätevät suurelta osin edelleen.

Osa skripteistä on liitteenä dokumentin lopussa. Myös sanasto on dokumentin lopussa. Tietotekniikkatermeissä pyritään johdonmukaisuuteen suomennoksessa tai sitten niitä ei ole suomennettu lainkaan. Sanastossa on pyritty selittämään tässä dokumentissa käytetyt termit siten, että peruskäyttäjänkin ne ymmärtää. Myös sanastossa oleviin termeihin on linkki sitä mukaa, kun niitä tulee dokumentissa vastaan.

Terminaaliin eli pääte-emulaattoriin ajettavat komennot ovat tässä teoksessa kirjoitettu aina harmaan laatikon sisälle:

```
sudo apt-get install <xxx>
```

<xxx> ei kirjoiteta sellaisenaan, vaan käyttäjä muokkaa sen vastaamaan omaa tarvettaan.

2 ASENNUSYMPÄRISTÖN PYSTYTTÄMINEN

Tässä kappaleessa käydään läpi asennusympäristön vaatimuksia.

2.1 Isäntäkoneen vaatimukset

Tehoa tarvitaan käännösvaiheessa mielellään liikaa kuin liian vähän. Jotta paketien kääntämiseen ei menisi päiviä, olisi hyvä, jos isäntäkoneessa olisi tehokas prosessori/prosessorit.

Jos on vähän RAM:ia, voi olla tarpeen tehdä [swap](#)-partitio. Helpommalla pääsee, jos swap:ia ei tarvitse tehdä. Swap toimii RAM-muistin laajennoksena, silloin jos RAM loppuu koneesta kesken. Jatkuva swap-osion käyttö hidastaa konetta huomattavasti.

Itse LFS ei vaadi kuin minimissään 6 GB levytilaa. Vähintään sen verran vapaata tilaa pitää siis isäntäkoneella olla. Mielellään kuitenkin väh. 20 GB, jotta on varaa asentaa paketteja vielä LFS:n rakentamisen jälkeenkin, jos haluaa.

Seuraavaksi on lueteltu paketteja, joita LFS:n rakentamiseen tarvitaan. Isäntäkoneen käyttöjärjestelmää ei ole rajattu. Periaatteessa mikä tahansa kriteerit täyttävä käyttöjärjestelmä käy. Ubuntun kanssa monilla on ollut ongelmia ja niin oli tämän kirjan tekijälläkin, joten sitä en suosittelisi, joskin silläkin LFS:n rakentaminen pienen kikkailun myötä todennäköisesti onnistuu. Yleisesti suositellaan Slackwarea, koska siinä tulee jo valmiiksi mukana kaikki tarvittavat paketit. Slackwarea käytettiin tämän dokumentin LFS:n asennusta tehtäessä.

Käyttöjärjestelmän valinta on käyttäjästä kiinni, joten minkään käyttöjärjestelmän asentamista ei tässä dokumentissa sen tarkemmin käsitellä. Dokumentissa lähde-tään liikkeelle sillä oletuksella, että käyttäjällä on valmis toimiva Linux-ympäristö käytössään, jolla LFS:n voi tehdä.

2.1.1 Paketit

Isäntäkoneella pitäisi olla asennettuna seuraavat paketit (nimi - min. versiovaatimus):

- **Bash-3.2** (`/bin/sh` pitäisi olla [symbolinen linkki](#) tai [kova linkki bashiin](#))
- **Binutils-2.17** (Versiot 2.27 ylöspäin eivät ole suositeltuja, koska niiden yhteensopivuutta ei ole testattu)
- **Bison-2.3** (`/usr/bin/yacc` pitäisi olla linkki bisoniin tai pieni skripti, joka suorittaa bisonin)
- **Bzip2-1.0.4**
- **Coreutils-6.9**
- **Diffutils-2.8.1**
- **Findutils-4.2.31**
- **Gawk-4.0.1** (`/usr/bin/awk` pitäisi olla linkki gawk:iin)
- **GCC-4.7** mukaan lukien C++ compiler, g++ (Versiot 6.2.0 ylöspäin eivät ole suositeltuja, koska niiden yhteensopivuutta ei ole testattu)
- **Glibc-2.11** (Versiot 2.24 ylöspäin eivät ole suositeltuja, koska niiden yhteensopivuutta ei ole testattu)
- **Grep-2.5.1a**
- **Gzip-1.3.12**
- **Linux Kernel-2.6.32**
- **M4-1.4.10**
- **Make-3.81**
- **Patch-2.5.4**
- **Perl-5.8.8**
- **Sed-4.1.5**
- **Tar-1.22**
- **Texinfo-4.7**
- **Xz-5.0.0**

Aiemmat versiot listan paketeista saattavat toimia, mutta niitä ei ole testattu. Seuraavalla `version-check.sh` -skriptillä voi tarkastaa onko hostilla kaikki tarvittavat paketit:

[Linkki skriptiin](#)

Skripti kopioidaan sellaisenaan terminaaliiin ja suoritetaan, jolloin tuloksena on jotain seuraavaanlaista:

```
root@dcjumalan_linukka:~# cat > version-check.sh << "EOF"
> #!/bin/bash
> # Simple script to list version numbers of critical development tools
> export LC_ALL=C
> bash --version | head -n1 | cut -d" " -f2-4
> MYSH=$(readlink -f /bin/sh)
> echo "/bin/sh -> $MYSH"
> echo $MYSH | grep -q bash || echo "ERROR: /bin/sh does not point to bash"
> unset MYSH
>
> echo -n "Binutils: "; ld --version | head -n1 | cut -d" " -f3-
> bison --version | head -n1
>
> if [ -h /usr/bin/yacc ]; then
>     echo "/usr/bin/yacc -> `readlink -f /usr/bin/yacc`";
> elif [ -x /usr/bin/yacc ]; then
>     echo yacc is `usr/bin/yacc --version | head -n1`
> else
>     echo "yacc not found"
> fi
>
> bzip2 --version 2>&1 < /dev/null | head -n1 | cut -d" " -f1,6-
> echo -n "Coreutils: "; chown --version | head -n1 | cut -d")" -f2
> diff --version | head -n1
> find --version | head -n1
> gawk --version | head -n1
>
> if [ -h /usr/bin/awk ]; then
>     echo "/usr/bin/awk -> `readlink -f /usr/bin/awk`";
> elif [ -x /usr/bin/awk ]; then
```

Esimerkkisuoritus `version-check.sh` -skriptistä.

```
root@dcjumalan_linukka:~#  
root@dcjumalan_linukka:~# bash version-check.sh  
bash, version 4.2.45(2)-release  
/bin/sh -> /bin/bash  
Binutils: (Linux/GNU Binutils) 2.23.52.0.1.20130226  
bison (GNU Bison) 2.7  
yacc is bison (GNU Bison) 2.7  
bzip2, Version 1.0.6, 6-Sept-2010.  
Coreutils: 8.21  
diff (GNU diffutils) 3.3  
find (GNU findutils) 4.4.2  
GNU Awk 4.1.0, API: 1.0 (GNU MPFR 3.1.2, GNU MP 5.1.3)  
/usr/bin/awk -> /bin/gawk-4.1.0  
gcc (GCC) 4.8.2  
g++ (GCC) 4.8.2  
(GNU libc) 2.17  
grep (GNU grep) 2.14  
gzip 1.6  
Linux version 3.10.17 (root@hive64) (gcc version 4.8.2 (GCC) ) #2 SMP Wed Oct 23  
16:34:38 CDT 2013  
m4 (GNU M4) 1.4.17  
GNU Make 3.82  
GNU patch 2.7  
Perl version='5.18.1';  
sed (GNU sed) 4.2.2  
tar (GNU tar) 1.26  
makeinfo (GNU texinfo) 4.13  
xz (XZ Utils) 5.0.5  
g++ compilation OK  
root@dcjumalan_linukka:~#
```

Symboliset linkitkin voi tarkastaa skriptin tulosteesta. Ne on alleviivattu kuvassa punaisella.

Paketteja ja niiden versionumeroita verrataan edellisellä sivulla olevaan listaan. Puuttuvat paketit pitää asentaa manuaalisesti.

Seuraavista gcc:n käyttämistä [kirjastoista](#) isäntäkoneella saa olla kaikki kolme tai ei yhtään:

- libgmp.la
- libmpfr.la
- libmpc.la

Jos isäntäkoneella on yksi tai kaksi edeltävistä, ne pitää joko uudelleennimetä tai poistaa tai vaihtoehtoisesti asentaa puuttuvat .la -tiedostot.

Seuraavalla `library-check.sh` -skriptillä voi tarkastaa mitä niistä isäntäkoneella on:

[Linkki skriptiin](#)

```
root@dcjumalan_linukka:~# cat > library-check.sh << "EOF"
> #!/bin/bash
> for lib in lib{gmp,mpfr,mpc}.la; do
>   echo $lib: $(if find /usr/lib* -name $lib |
>                 grep -q $lib;then :;else echo not;fi) found
> done
> unset lib
> EOF
root@dcjumalan_linukka:~#
root@dcjumalan_linukka:~# bash library-check.sh
libgmp.la: found
libmpfr.la: found
libmpc.la: found
root@dcjumalan_linukka:~# █
```

gcc-kirjastojen tarkistusskripti ja esimerkki sen suorituksesta.

Skripti kopioidaan terminaaliin ja ajetaan. Kuvan tapauksessa koneessa oli kaikki paketit, joten toimenpiteitä ei tarvita.

2.2 Uuden levyjaon luominen

LFS-järjestelmä tarvitsee oman osion tietokoneen kovalevyiltä. Tässä kappaleessa käydään läpi kyseisen levyjaon luonti.

2.2.1 Teoriaa

Vähimmäisvaatimus LFS-järjestelmälle on 6 GB levytilaa. Kannattaa kuitenkin varata vähän kasvutilaa, jos haluaa asentaa jotain vielä senkin jälkeen, kun LFS on valmis.

Pakettien käännösvaiheessa on hyvä olla tehoa koneessa, jotta käännös sujuu nopeammin eikä vie päiviä. Useampi kuin yksi core auttaa siinä vaiheessa.

Mikäli käytössä ei ole tarpeeksi RAM:ia, kannattaa käyttää pientä partitiota [swap](#):na. [Kernel](#) käyttää swap:ia varastoidakseen sinne harvoin käytettyä dataa jättäen siten enemmän muistia vapaaksi aktiivisille prosesseille. LFS:n swap voi olla sama kuin isäntäkoneen swap, joten erillistä levyjakoa sille ei välttämättä tarvitse tehdä. Swap:in käyttö hidastaa konetta. Jos järjestelmä käyttää swap:ia koko ajan, kannattaa lisätä RAM:in määrää.

Levyjako tehdään esimerkiksi `cfdisk` tai `fdisk` -ohjelmaa käyttäen. Seuraavassa on lueteltuna välttämättömät ja vaihtoehtoiset levyosiot:

Pakolliset:

- `/` (LFS root hakemisto): 10 GB on tarpeeksi

Pakolliset joissain tapauksissa:

- Grub BIOS: Jos boot disk on jaettu käyttäen GPT:tä (GUID Partition Table), pieni 1 MB:n partitiio pitää tehdä, jollei sitä ole jo olemassa. Tätä partitiota ei formatoida. GRUB käyttää sitä käynnistyksen latausohjelman asennuksen aikana. `fdisk`:iä käyttäessä tämän levyjaon nimi on yleensä BIOS

Boot, `gdisk`:iä käyttäessä sen koodi on `EF02`. **HUOM!** GRUB partition pitää olla samalla asemalla kuin mitä BIOS käyttää järjestelmän käynnistämiseen. Se ei ole välttämättä sama asema, millä root LFS on.

- Swap: Suositeltu koko on yleensä tuplasti fyysisen RAM:n määrä. Jos levytilaa on rajallisesti, swap-partitiolle voi laittaa 2 GB ja tarkkailla sen käyttöä.

Valinnaiset:

- `/boot`: Erittäin suositeltava. Tätä partitiota käytetään kerneleiden ja muun käynnistykseen liittyvän datan säilyttämiseen. 100 MB on sopiva koko.
- `/home`: Erittäin suositeltava. Voi jakaa kotihakemiston muiden [distrojen](#) tai LFS buildien kanssa. Koko riippuu käytettävissä olevasta levytilasta.
- `/usr`: Ei tavallisesti tarvita LFS:ssä. `usr` tulee sanoista Unix System Resources tai Unix Source Repository tai User System Resources. Hyödyllinen tarjottaessa serveriä [Thin Clientille](#). 5 GB on riittävä koko.
- `/opt`: 5-10 GB. Hyödyllinen [BLFS](#):ssä, kun halutaan asentaa monia suuria paketteja, kuten KDE tai Gnome.
- `/tmp`: Harvinainen omana osiona. Hyödyllinen konfiguroitaessa kevyttä asiakaspäätettä (eng. Thin Client). 2 GB riittää.
- `/usr/src`: Paikka, jonne varastoidaan BLFS:n [lähdetiedostot](#). Käytetään myös BLFS-pakettien käännöspaikkana. 35-50 GB.

2.2.2 Levyosiointi käytännössä

Esimerkiksi seuraavanlainen levyjako on toimiva:

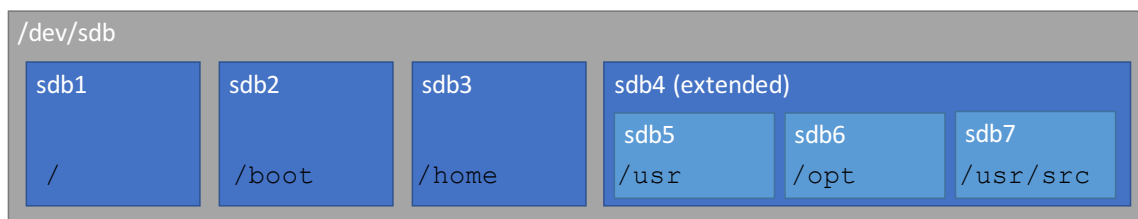
- `/` (LFS root) 15 GB - primary
- `/boot` 500 MB - primary
- `/home` 20 GB - primary

- `/usr` 5 GB
- `/opt` 10 GB
- `/usr/src` 50 GB

Seuraavaksi tehdään vaihe vaiheelta yllä oleva levyjako. Halutessasi voit tehdä omanlaisesi levyjaon dokumentin esimerkkiä seuraten.

Teemme yhteensä kolme [primary-partitionia](#): `/` (LFS root), `/home` ja `/boot`. Lisäksi teemme yhden [extended-partition](#), johon jaamme loput eli `/usr`, `/opt` ja `/usr/src` hakemistot. Jako primary- ja extended osioihin johtuu siitä, että kova-levyn voi jakaa korkeintaan neljään primary-osioon. Jos haluaa useamman kuin neljä osiota, yksi primary-osio on merkattava extended-osioksi, jonka sisään tulevat loput levyosiot (logical).

Kaavio pian tehtävästä levyjaosta:



- Kovalevy
- Looginen (logical) osio
- Ensijainen (primary) osio / Hakemistonimi, johon osio kiinnitetään

Ensin katsotaan levytyökalulla koneelta löytyvät levyt ja levyosiot, sekä niiden nimet.

```
fdisk -l
```

```
root@dcjumalan_linukka:~# fdisk -l

Disk /dev/sda: 214.7 GB, 214748364800 bytes
255 heads, 63 sectors/track, 26108 cylinders, total 419430400 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x011ddef9

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *        2048     20973567     10485760    83   Linux
/dev/sda2                20973568     88082431     33554432    82   Linux swap

Disk /dev/sdb: 161.1 GB, 161061273600 bytes
255 heads, 63 sectors/track, 19581 cylinders, total 314572800 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Sitten valitaan se kovalevy, johon muutoksia halutaan tehdä.

```
fdisk </dev/sdb>
```

```
root@dcjumalan_linukka:~# fdisk /dev/sdb
Welcome to fdisk (util-linux 2.21.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0xd36afcc4.

Command (m for help): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-314572799, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-314572799, default 314572799): +15G
Partition 1 of type Linux and of size 15 GiB is set
```

Yllä olevassa kuvassa on tehty /dev/sdb1 eli LFS root –partitio fdisk-työkalulla.

```
n
```

n aloittaa uuden osion luomisen.

```
p
```

p asettaa uuden osion tyyppiä primary:n.

```
1
```

1 asettaa osion numeroksi 1.

```
[enter]
```

[enter] tarkoittaa vain enter-näppäimen painamista. Tässä kohti default-asetus on hyvä, joten siihen ei tarvitse kirjoittaa mitään.

```
+15G
```

ja +15G asettaa osion kooksi 15 gigaa.

Seuraavaksi tehdään /dev/sdb2 (/boot) osio:

```
Command (m for help): n
Partition type:
  p   primary (1 primary, 0 extended, 3 free)
  e   extended
Select (default p): p
Partition number (1-4, default 2):
Using default value 2
First sector (31459328-314572799, default 31459328):
Using default value 31459328
Last sector, +sectors or +size{K,M,G} (31459328-314572799, default 314572799): +500M
Partition 2 of type Linux and of size 500 MiB is set
```

```
n
```

```
p
```

```
[enter]
```

```
[enter]
```

```
+500M
```

Sitten tehdään /dev/sdb3 (/home) osio:

```
Command (m for help): n
Partition type:
  p   primary (2 primary, 0 extended, 2 free)
  e   extended
Select (default p): p
Partition number (1-4, default 3):
Using default value 3
First sector (32483328-314572799, default 32483328):
Using default value 32483328
Last sector, +sectors or +size{K,M,G} (32483328-314572799, default 314572799): +20G
Partition 3 of type Linux and of size 20 GiB is set
```

```
n
p
[enter]
[enter]
+20GB
```

Seuraavaksi tehdään /dev/sdb4 osio, josta tulee extended-partitio ja jonka sisään tulevat /usr, /opt ja /usr/src.

```
Command (m for help): n
Partition type:
   p   primary (3 primary, 0 extended, 1 free)
   e   extended
Select (default e): e
Selected partition 4
First sector (74426368-314572799, default 74426368):
Using default value 74426368
Last sector, +sectors or +size{K,M,G} (74426368-314572799, default 314572799): +75G
Partition 4 of type Extended and of size 75 GiB is set
```

```
n
e
[enter]
[enter]
+70GB
```

Lopuksi osiointitaulukko kirjoitetaan levyyn komennolla:

```
w
```

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
root@dcjumalan linukka:~#
```

Komennolla

```
fdisk -l
```

katsotaan, että osiointi onnistui.

Lopputulos näyttää jokseenkin tältä:

```
root@dcjumalan_linukka:~# fdisk -l

Disk /dev/sda: 214.7 GB, 214748364800 bytes
255 heads, 63 sectors/track, 26108 cylinders, total 419430400 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x011ddef9

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1 *        2048       20973567    10485760    83   Linux
/dev/sda2          20973568    88082431    33554432    82   Linux swap

Disk /dev/sdb: 161.1 GB, 161061273600 bytes
255 heads, 63 sectors/track, 19581 cylinders, total 314572800 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xd36afcc4

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1          2048       31459327    15728640    83   Linux
/dev/sdb2        31459328    32483327     512000    83   Linux
/dev/sdb3        32483328    74426367    20971520    83   Linux
/dev/sdb4        74426368    231712767    78643200     5   Extended
root@dcjumalan_linukka:~#
```

Seuraavaksi extended-partition alle tehdään [logical-partitioita](#). Avataan jälleen `fdisk`:llä kovalevyn osiointitaulukko:

```
fdisk </dev/sdb>
```

```
root@dcjumalan_linukka:~# fdisk /dev/sdb
Welcome to fdisk (util-linux 2.21.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): n
All primary partitions are in use
Adding logical partition 5
First sector (74428416-231712767, default 74428416):
Using default value 74428416
Last sector, +sectors or +size{K,M,G} (74428416-231712767, default 231712767): +5G
Partition 5 of type Linux and of size 5 GiB is set
```

Tehdään yllä olevan kuvan mukaisesti `/dev/sdb5` (`/usr`) osio:

```
n
[enter]
+5GB
```

Seuraavaksi tehdään /dev/sdb6 (/opt) osio:

```
Command (m for help): n
All primary partitions are in use
Adding logical partition 6
First sector (84916224-231712767, default 84916224):
Using default value 84916224
Last sector, +sectors or +size{K,M,G} (84916224-231712767, default 231712767): +10G
Partition 6 of type Linux and of size 10 GiB is set
```

n

[enter]

+10GB

Lopuksi tehdään vielä /dev/sdb7 (/usr/src) osio:

```
Command (m for help): n
All primary partitions are in use
Adding logical partition 7
First sector (105889792-231712767, default 105889792):
Using default value 105889792
Last sector, +sectors or +size{K,M,G} (105889792-231712767, default 231712767): +50G
Partition 7 of type Linux and of size 50 GiB is set
```

n

[enter]

+50GB

w näppäimen painalluksella kirjoitetaan muutokset levyyn:

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
root@dcjumalan_linukka:~#
```

w

Jälleen komennolla

```
fdisk -l
```

katsotaan, että partitiointi onnistui.

```

root@dcjumalan_linukka:~# fdisk -l

Disk /dev/sda: 214.7 GB, 214748364800 bytes
255 heads, 63 sectors/track, 26108 cylinders, total 419430400 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x011ddef9

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1    *          2048       20973567    10485760    83   Linux
/dev/sda2                20973568    88082431    33554432    82   Linux swap

Disk /dev/sdb: 161.1 GB, 161061273600 bytes
255 heads, 63 sectors/track, 19581 cylinders, total 314572800 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xd36afcc4

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1                2048       31459327    15728640    83   Linux
/dev/sdb2          31459328       32483327         512000    83   Linux
/dev/sdb3          32483328       74426367    20971520    83   Linux
/dev/sdb4          74426368       231712767    78643200     5   Extended
/dev/sdb5          74428416       84914175     5242880    83   Linux
/dev/sdb6          84916224       105887743    10485760    83   Linux
/dev/sdb7         105889792       210747391    52428800    83   Linux
root@dcjumalan_linukka:~# █

```

Levyosiointi näyttää nyt tältä.

2.3 Tiedostojärjestelmän luominen

LFS olettaa, että root-osion tiedostojärjestelmä on tyyppiä [ext4](#). Ext4:n luomiseksi aja seuraava komento:

```
mkfs -v -t ext4 /dev/<xxx>
```

jossa <xxx>:n tilalle tulee levyosion nimi (esim. /dev/sdb1).

```
root@dcjumalan_linukka:~# mkfs -v -t ext4 /dev/sdb1
mke2fs 1.42.8 (20-Jun-2013)
fs_types for mke2fs.conf resolution: 'ext4'
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
983040 inodes, 3932160 blocks
196608 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4026531840
120 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

root@dcjumalan_linukka:~#
```

Jokainen levyosio Extended-partitiota lukuun ottamatta pitää formatoida, eli osiot sdb1-3 ja sdb5-7 formatoidaan yllä mainitulla komennolla.

Käytettäessä jo valmiina olevaa swap-osiota sitä ei tarvitse enää alustaa. Uusi swap-osio pitää ja se onnistuu seuraavalla komennolla:

```
mkswap /dev/<xxx>
```

jossa <xxx>:n tilalle tulee swap:in levyosion nimi.

2.3.1 Levyosioiden uudelleennimeäminen

Osioiden uudelleennimeäminen tehdään komennolla

```
e2label </dev/xxx> <new_name>
```

```
root@dcjumalan_linukka:~# e2label /dev/sdb1 /lfs
root@dcjumalan_linukka:~# e2label /dev/sdb2 /boot
root@dcjumalan_linukka:~# e2label /dev/sdb3 /home
root@dcjumalan_linukka:~# e2label /dev/sdb5 /usr
root@dcjumalan_linukka:~# e2label /dev/sdb6 /opt
root@dcjumalan_linukka:~# e2label /dev/sdb7 /usr/src
root@dcjumalan_linukka:~#
```

Kaikki osiot pitää nimetä erikseen. Nimettävät osiot:

- /dev/sdb1 → /lfs
- /dev/sdb2 → /boot
- /dev/sdb3 → /home
- /dev/sdb5 → /usr
- /dev/sdb6 → /opt
- /dev/sdb7 → /usr/src

Myöhemmässä vaiheessa /dev/sdb1, joka nyt nimetään /lfs osioksi, on / eli LFS root -osio.

2.4 LFS-muuttujan asettaminen

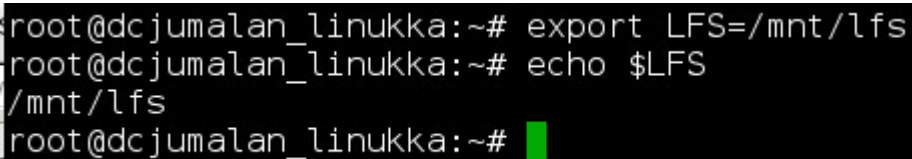
Seuraavaksi asetetaan `LFS`-muuttuja osoittamaan `LFS root` -osioon, jolloin aina, kun käytämme komentoa, jossa on mukana `$LFS`-muuttuja, tekemisemme osoittavat `LFS root` -osioon (`/mnt/lfs`).

`LFS`-muuttujan arvo asetetaan seuraavalla komennolla:

```
export LFS=/mnt/lfs
```

Seuraavalla komennolla voi tarkistaa, että `LFS`-muuttuja osoittaa oikeaan paikkaan:

```
echo $LFS
```



```
root@dcjumalan_linukka:~# export LFS=/mnt/lfs
root@dcjumalan_linukka:~# echo $LFS
/mnt/lfs
root@dcjumalan_linukka:~#
```

`echo $LFS` -komennolla kannattaa tarkastaa polku vähän väliä tämän kirjan toisen luvun aikana. Jos muuttujan arvo ei osoita oikeaan polkuun (`/mnt/lfs`), komennot eivät toimi. Siinä tapauksessa käytä yllä mainittua `export`-komentoa korjataksesi ongelman.

2.5 Uuden partition liittäminen

Kerrataan. Tässä vaiheessa on luotu uusi osio LFS root, jonka tiedostojärjestelmäksi on asetettu `ext4`. `$LFS`-muuttuja osoittaa hakemistoon, johon LFS on tarkoitus rakentaa eli `/mnt/lfs`. Kun komentoon kirjoittaa `$LFS`, terminaali automaattisesti tulkitsee sen muodossa `/mnt/lfs`. Seuraavaksi `/mnt/lfs` -hakemisto liitetään aiemmin luomaamme LFS root -osioon (tässä kirjassa levyosio `/dev/sdb1`).

Tehdään hakemisto seuraavalla komennolla:

```
mkdir -pv $LFS
```

```
root@dcjumalan_linukka:~# mkdir -pv $LFS
mkdir: created directory '/mnt/lfs'
root@dcjumalan_linukka:~#
```

Sitten liitetään levyosio `/dev/sdb1` hakemistoon `/mnt/lfs` seuraavalla komennolla:

```
mount -v -t ext4 /dev/sdb1 $LFS
```

```
root@dcjumalan_linukka:~# mount -v -t ext4 /dev/sdb1 $LFS
/dev/sdb1 on /mnt/lfs type ext4 (rw)
root@dcjumalan_linukka:~#
```

Jos käytetään monia osioita LFS:lle (esim. yksi osio on `/` ja toinen `/usr`), niin jokainen niistä täytyy liittää erikseen. Aiemmin teimme seuraavanlaisen osioinnin:

- `/dev/sdb1` → `/`
- `/dev/sdb2` → `/boot`
- `/dev/sdb3` → `/home`
- `/dev/sdb5` → `/usr`
- `/dev/sdb6` → `/opt`
- `/dev/sdb7` → `/usr/src`

Äsken liitimme `/dev/sdb1` → `/mnt/lfs`, joten seuraavaksi sama täytyy tehdä lopuille osioille.

```
mkdir -v $LFS/<xxx>
mount -v -t ext4 </dev/xxx> $LFS/<xxx>
```

```
root@dcjumalan_linukka:~# mkdir -v $LFS/boot
mkdir: created directory '/mnt/lfs/boot'
root@dcjumalan_linukka:~# mount -v -t ext4 /dev/sdb2 $LFS/boot
/dev/sdb2 on /mnt/lfs/boot type ext4 (rw)
```

```
root@dcjumalan_linukka:~# mkdir -v $LFS/home
mkdir: created directory '/mnt/lfs/home'
root@dcjumalan_linukka:~# mount -v -t ext4 /dev/sdb3 $LFS/home
/dev/sdb3 on /mnt/lfs/home type ext4 (rw)
root@dcjumalan_linukka:~# █
```

```
root@dcjumalan_linukka:~# mkdir -v $LFS/usr
mkdir: created directory '/mnt/lfs/usr'
root@dcjumalan_linukka:~# mount -v -t ext4 /dev/sdb5 $LFS/usr
/dev/sdb5 on /mnt/lfs/usr type ext4 (rw)
root@dcjumalan_linukka:~# █
```

```
root@dcjumalan_linukka:~# mkdir -v $LFS/opt
mkdir: created directory '/mnt/lfs/opt'
root@dcjumalan_linukka:~# mount -v -t ext4 /dev/sdb6 $LFS/opt
/dev/sdb6 on /mnt/lfs/opt type ext4 (rw)
root@dcjumalan_linukka:~# █
```

```
root@dcjumalan_linukka:~# mkdir -v $LFS/usr/src
mkdir: created directory '/mnt/lfs/usr/src'
root@dcjumalan_linukka:~# mount -v -t ext4 /dev/sdb7 $LFS/usr/src
/dev/sdb7 on /mnt/lfs/usr/src type ext4 (rw)
root@dcjumalan_linukka:~# █
```

Seuraavaksi varmistetaan, ettei uuden `$LFS`-hakemiston käyttöoikeudet ole liian rajoittavat. Jos `nosuid` ja/tai `nodev` on asetettu, partitio pitää liittää uudelleen.

```
mount
```

```
dcjumalan_linukka:~# mount
/sda1 on / type ext4 (rw)
on /proc type proc (rw)
s on /sys type sysfs (rw)
s on /dev/shm type tmpfs (rw)
d-fuse on /root/.gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev)
'sdb1 on /mnt/lfs type ext4 (rw)
'sdb2 on /mnt/lfs/boot type ext4 (rw)
'sdb3 on /mnt/lfs/home type ext4 (rw)
'sdb5 on /mnt/lfs/usr type ext4 (rw)
'sdb6 on /mnt/lfs/opt type ext4 (rw)
'sdb7 on /mnt/lfs/usr/src type ext4 (rw)
dcjumalan_linukka:~# █
```

Kuvassa kaikki on kohdallaan, joten voidaan jatkaa eteenpäin.

Jos käyttää swap-partitiota, täytyy varmistaa, että se on käytössä.

```
/sbin/swapon -v /dev/<xxx>
```

<xxx>:n tilalle tulee swap-partition nimi.

2.6 Paketit ja korjaustiedostot

Tässä luvussa ladataan paketit ja korjaustiedostot, joita tarvitaan LFS:n rakentamiseen.

2.6.1 Sources-hakemiston luominen

Paketit pitää asentaa hakemistoon, joka on johdonmukaisesti löydettävissä ja käytettävissä koko käännösvaiheen ajan. Sitä varten luodaan `$LFS/sources` -hakemisto, johon paketit ladataan. Hakemisto tehdään root-käyttäjänä seuraavalla komennolla:

```
mkdir -v $LFS/sources
```

```
root@dcjumalan_linukka:~# mkdir -v $LFS/sources
mkdir: created directory '/mnt/lfs/sources'
root@dcjumalan_linukka:~#
```

Seuraavaksi äsken luodusta sources-hakemistosta tehdään 'writable' ja 'sticky'. Writable tarkoittaa, että hakemistoon voi kirjoittaa. Sticky tarkoittaa, että vaikka usealla käyttäjällä olisi kirjoitusoikeus hakemistoon, ainoastaan tiedoston omistaja pystyy poistamaan tiedoston hakemistosta. Ajetaan seuraava komento oikeuksien muuttamiseksi:

```
chmod -v a+wt $LFS/sources
```

```
root@dcjumalan_linukka:~# chmod -v a+wt $LFS/sources
mode of '/mnt/lfs/sources' changed from 0755 (rwxr-xr-x) to 1777 (rwxrwxrwt)
root@dcjumalan_linukka:~#
```

2.6.2 Pakettien lataaminen

HUOM! Paketeista kannattaa asentaa sama versio, jota tässä kirjassa käytetään. Uudemmat tai vanhemmat versiot voivat aiheuttaa yhteensopivuusongelmia, joihin ei tässä kirjassa esitetä ratkaisua.

Paketit ladataan `wget` -komennolla. Ensin luodaan tekstieditorilla (esim. `nano`) `wget-list` -niminen tiedosto. `wget-list` sisältää kaikkien pakettien latausosoitteet. Tiedoston luomiseksi `nano`-tekstieditorilla ajetaan seuraava komento:

```
nano wget-list
```

```
root@dcjumalan_linukka:~# nano wget-list
```

Tiedoston sisällöksi kopioidaan lista, joka löytyy liitteistä tämän kirjan lopusta. Suora linkki siihen on [tässä](#). Sama lista löytyy myös internetistä: <http://www.linuxfromscratch.org/lfs/view/stable/wget-list>.



```
GNU nano 2.3.2 File: wget-list
http://download.savannah.gnu.org/releases/acl/acl-2.2.52.src.tar.gz
http://download.savannah.gnu.org/releases/attr/attr-2.4.47.src.tar.gz
http://ftp.gnu.org/gnu/autoconf/autoconf-2.69.tar.xz
http://ftp.gnu.org/gnu/automake/automake-1.15.tar.xz
http://ftp.gnu.org/gnu/bash/bash-4.4.tar.gz
http://alpha.gnu.org/gnu/bc/bc-1.06.95.tar.bz2
http://ftp.gnu.org/gnu/binutils/binutils-2.27.tar.bz2
http://ftp.gnu.org/gnu/bison/bison-3.0.4.tar.xz
http://www.bzip.org/1.0.6/bzip2-1.0.6.tar.gz
https://github.com/libcheck/check/releases/download/0.11.0/check-0.11.0.tar.gz
http://ftp.gnu.org/gnu/coreutils/coreutils-8.26.tar.xz
http://dbus.freedesktop.org/releases/dbus/dbus-1.10.14.tar.gz
http://ftp.gnu.org/gnu/dejagnu/dejagnu-1.6.tar.gz
http://ftp.gnu.org/gnu/diffutils/diffutils-3.5.tar.xz
http://dev.gentoo.org/~blueness/eudev/eudev-3.2.1.tar.gz
http://downloads.sourceforge.net/project/e2fsprogs/e2fsprogs/v1.43.4/e2fsprogs-1.43.4$
http://prdownloads.sourceforge.net/expat/expat-2.2.0.tar.bz2
http://prdownloads.sourceforge.net/expect/expect5.45.tar.gz
ftp://ftp.astron.com/pub/file/file-5.30.tar.gz
http://ftp.gnu.org/gnu/findutils/findutils-4.6.0.tar.gz
[ Read 80 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page  ^U UnCut Text ^T To Spell
```

<Ctrl+o> ja [enter] kirjoittaa tiedoston ja <ctrl+x>:llä pääsee pois editorista.

Seuraavaksi ladataan `wget-list`:n sisältämät paketit komennolla:

```
wget --input-file=wget-list --continue --directory-prefix=$LFS/sources
```

```
root@dcjumalan_linukka:~# wget --input-file=wget-list --continue --directory-prefix=$LFS/sources
```

Kestää muutaman minuutin, kun järjestelmä latailee paketteja.

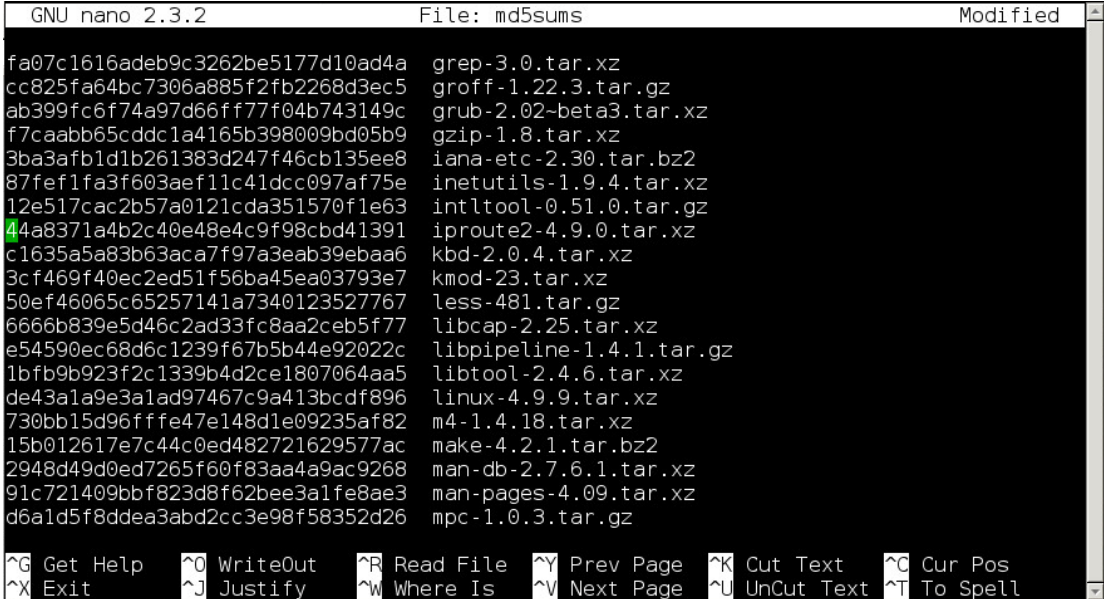
```
FINISHED --2017-03-06 11:42:43--  
Total wall clock time: 4m 46s  
Downloaded: 79 files, 358M in 3m 20s (1.79 MB/s)  
root@dcjumalan_linukka:~#
```

Seuraavaksi on syytä tarkistaa, että kaikkien tarvittavien pakettien lataus onnistui. Ensin luodaan tekstieditorilla (esim. nano) md5sums-niminen tiedosto hakemistoon \$LFS/sources:

```
nano md5sums /mnt/lfs/sources
```

```
root@dcjumalan_linukka:~# nano md5sums /mnt/lfs/sources
```

Md5sums-tiedoston sisällöksi kopioidaan lista, joka löytyy tämän kirjan liitteistä [täältä](#) tai internetistä: <http://www.linuxfromscratch.org/lfs/view/stable/md5sums>.



The screenshot shows the GNU nano 2.3.2 text editor with the file md5sums open. The editor displays a list of package names and their corresponding MD5 checksums. The list includes packages like grep, groff, grub, gzip, iana-etc, inetutils, intltool, iproute2, kbd, kmod, less, libcap, libpipeline, libtool, linux, m4, make, man-db, man-pages, and mpc. The cursor is positioned at the beginning of the line containing the MD5 checksum 4a8371a4b2c40e48e4c9f98cbd41391 for the package iproute2. The editor's status bar at the bottom shows various keyboard shortcuts for navigation and editing.

```
GNU nano 2.3.2 File: md5sums Modified  
fa07c1616adeb9c3262be5177d10ad4a grep-3.0.tar.xz  
cc825fa64bc7306a885f2fb2268d3ec5 groff-1.22.3.tar.gz  
ab399fc6f74a97d66ff77f04b743149c grub-2.02-beta3.tar.xz  
f7caabb65cddc1a4165b398009bd05b9 gzip-1.8.tar.xz  
3ba3afb1d1b261383d247f46cb135ee8 iana-etc-2.30.tar.bz2  
87fef1fa3f603aef11c41dcc097af75e inetutils-1.9.4.tar.xz  
12e517cac2b57a0121cda351570f1e63 intltool-0.51.0.tar.gz  
4a8371a4b2c40e48e4c9f98cbd41391 iproute2-4.9.0.tar.xz  
c1635a5a83b63aca7f97a3eab39ebaa6 kbd-2.0.4.tar.xz  
3cf469f40ec2ed51f56ba45ea03793e7 kmod-23.tar.xz  
50ef46065c65257141a7340123527767 less-481.tar.gz  
6666b839e5d46c2ad33fc8aa2ceb5f77 libcap-2.25.tar.xz  
e54590ec68d6c1239f67b5b44e92022c libpipeline-1.4.1.tar.gz  
1bfb9b923f2c1339b4d2ce1807064aa5 libtool-2.4.6.tar.xz  
de43a1a9e3a1ad97467c9a413bcd896 linux-4.9.9.tar.xz  
730bb15d96fffe47e148d1e09235af82 m4-1.4.18.tar.xz  
15b012617e7c44c0ed482721629577ac make-4.2.1.tar.bz2  
2948d49d0ed7265f60f83aa4a9ac9268 man-db-2.7.6.1.tar.xz  
91c721409bbf823d8f62bee3a1fe8ae3 man-pages-4.09.tar.xz  
d6a1d5f8ddea3abd2cc3e98f58352d26 mpc-1.0.3.tar.gz  
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Seuraavaksi ajetaan komennot:

```
pushd $LFS/sources  
md5sum -c md5sums  
popd
```

```
XML-Parser-2.44.tar.gz: OK  
xz-5.2.3.tar.xz: OK  
zlib-1.2.11.tar.xz: OK  
bash-4.4-upstream_fixes-1.patch: OK  
bc-1.06.95-memory_leak-1.patch: OK  
bzip2-1.0.6-install_docs-1.patch: OK  
coreutils-8.26-i18n-1.patch: OK  
glibc-2.25-fhs-1.patch: OK  
kbd-2.0.4-backspace-1.patch: OK  
md5sum: WARNING: 1 line is improperly formatted  
root@dcjumalan_linukka:/mnt/lfs/sources# popd  
/mnt/lfs/sources ~  
root@dcjumalan_linukka:/mnt/lfs/sources#
```

Jos jokin paketti uupuu, se pitää ladata manuaalisesti. Helpoiten se onnistuu etsimällä googlesta paketin latausosoite ja lataamalla se `wget`:iä käyttäen.

```
cd /mnt/lfs/sources  
wget http://<xxx.xxx/xxx.tar.xx>
```

2.7 Asennusympäristön viimeistely

Tässä luvussa tehdään hakemisto tilapäisten työkalujen asentamista varten, luodaan uusi ei-admin-käyttäjä riskien vähentämiseksi ja tehdään käännösympäristö uudelle käyttäjälle.

2.7.1 \$LFS/tools -hakemiston luominen

Kaikki [2.8](#)-kappaleessa käännettävät paketit asennetaan `$LFS/tools` -hakemistoon, jotta ne pysyvät erillään [2.15](#)-kappaleessa asennettavista paketeista. Ohjelmat, jotka käännetään 2.8-luvussa ovat tilapäisiä työkaluja, jotka eivät tule olemaan osa lopullista LFS-järjestelmää. Pitämällä nämä ohjelmat omassa hakemistossaan, ne on helppo heittää menemään käytön jälkeen, eivätkä ne sotkeudu isäntäjärjestelmän hakemistoihin.

Luodaan uusi `/tools`-hakemisto `root`-käyttäjänä.

```
mkdir -v $LFS/tools
```

```
root@dcjumalan_linukka:/mnt/lfs/sources# mkdir -v $LFS/tools
mkdir: created directory '/mnt/lfs/tools'
```

Seuraavaksi luodaan [symbolinen linkki](#) isäntäjärjestelmään.

```
ln -sv $LFS/tools /
```

```
root@dcjumalan_linukka:/mnt/lfs/sources# ln -sv $LFS/tools /
'/tools' -> '/mnt/lfs/tools'
root@dcjumalan_linukka:/mnt/lfs/sources# █
```

Luotu linkki sallii työkaluketjun kääntämisen siten, jotta se viittaa aina `/tools` hakemistoon. Tämä tarkoittaa, että [kääntäjä](#), [assembler](#) ja [linkkeri](#) toimivat sekä tässä kappaleessa (kun käytämme yhä joitakin työkaluja isäntäjärjestelmästä) että seuraavassa kappaleessa (kun toimimme [chroot](#)-ympäristössä).

2.7.2 Uuden käyttäjän luominen

Tehdään uusi `lfs`-käyttäjä ilman pääkäyttäjän (`root`) oikeuksia riskien vähentämiseksi.

Ensin tehdään uusi `lfs`-niminen käyttäjäryhmä:

```
groupadd lfs
```

```
root@dcjumalan_linukka:/mnt/lfs/sources# groupadd lfs
root@dcjumalan_linukka:/mnt/lfs/sources#
```

Sen jälkeen luodaan `lfs`-niminen käyttäjä äsken luotuun ryhmään:

```
useradd -s /bin/bash -g lfs -m -k /dev/null lfs
```

```
root@dcjumalan_linukka:/mnt/lfs/sources# useradd -s /bin/bash -g lfs -m -k /dev/null lfs
root@dcjumalan_linukka:/mnt/lfs/sources#
```

Puretaan komentoa hiukan auki:

- `-s /bin/bash` → Asettaa oletusshelliksi [bash](#):n.
- `-g lfs` → Lisää käyttäjän `lfs`-ryhmään.
- `-m` → Luo käyttäjälle kotihakemiston.
- `-k /dev/null` → Estää tiedostojen kopioinnin [skeleton](#)-hakemistosta (oletus on `/etc/skel`).
- `lfs` → Luodun käyttäjän nimi.

Seuraavaksi asetetaan `lfs`-käyttäjälle salasana, jotta käyttäjällä voidaan kirjautua.

```
passwd lfs
```

```
root@dcjumalan_linukka:/mnt/lfs/sources# passwd lfs
Changing password for lfs
Enter the new password (minimum of 5 characters)
Please use a combination of upper and lower case letters and numbers.
New password:
Re-enter new password:
passwd: password changed.
root@dcjumalan_linukka:/mnt/lfs/sources#
```

Vaihdetaan `$LFS/tools` -hakemiston oikeudet rootilta lfs-käyttäjälle:

```
chown -v lfs $LFS/tools
```

```
root@dcjumalan_linukka:/mnt/lfs/sources# chown -v lfs $LFS/tools
changed ownership of '/mnt/lfs/tools' from root to lfs
root@dcjumalan_linukka:/mnt/lfs/sources#
```

Tehdään sama `$LFS/sources` -hakemistolle:

```
chown -v lfs $LFS/sources
```

```
root@dcjumalan_linukka:/mnt/lfs/sources# chown -v lfs $LFS/sources
changed ownership of '/mnt/lfs/sources' from root to lfs
root@dcjumalan_linukka:/mnt/lfs/sources#
```

Lopuksi kirjaudutaan sisään lfs-käyttäjänä:

```
su - lfs
```

```
root@dcjumalan_linukka:/mnt/lfs/sources# su - lfs
```

“-“ ohjeistaa `su:n` käynnistämään login shellin non-login shellin sijaan. Näiden shellien eroista voi lukea man-sivuilta `bash(1)` ja `info bash`.

2.7.3 Ympäristön pystyttäminen

Seuraavaksi tehdään kaksi startup-tiedostoa bash-shellille, mitä LFS käyttää. Ensin tehdään `.bash_profile` -tiedosto ja sen lisäksi toinen `.bashrc` -tiedosto. Bash tulkki suorittaa käynnistuksen yhteydessä kummatkin tiedostot ja asettaa niihin tallennetut asetukset.

`.bash_profile` -tiedoston luominen:

```
cat > ~/.bash_profile << "EOF"
exec env -i HOME=$HOME TERM=$TERM PS1='\u:\w\$ ' /bin/bash
EOF
```

```
lfs@dcjumalan_linukka:~$ cat > ~/.bash_profile << "EOF"
> exec env -i HOME=$HOME TERM=$TERM PS1='\u:\w\$ ' /bin/bash
> EOF
lfs@dcjumalan_linukka:~$
```

Kun kirjautuu sisään lfs-käyttäjänä, ensimmäinen shell on yleensä login shell, joka lukee isäntäkoneelta `/etc/profile` -tiedoston (skriptin) ja suorittaa siellä olevat komennot (yleensä ne ovat joitakin asetuksia ja ympäristömuuttujia). Seuraavaksi shell hakee käyttäjän kotihakemistosta tiedoston `.bash_profile` ja suorittaa siellä olevat komennot.

`exec env -i.../bin/bash -komento .bash_profile` -tiedostossa korvaa käynnissä olevan shellin uudella täysin tyhjällä ympäristöllä lukuun ottamatta [HOME, TERM ja PS1](#) -muuttujia. Tämä varmistaa, että ei-toivotut ja mahdollisesti harsidit ympäristömuuttujat eivät pääse vuotamaan isäntäjärjestelmästä käännösympäristöön. Toisin sanoen tällä varmistetaan puhdas käännösympäristö.

Shellin uusi instanssi on non-login shell, joka ei suorita `/etc/profile` tai `.bash_profile` -skriptejä. Sen sijaan se suorittaa `.bashrc` -skriptin. Luodaan sellainen seuraavaksi.

.bashrc -tiedoston luominen:

```
cat > ~/.bashrc << "EOF"
set +h
umask 022
LFS=/mnt/lfs
LC_ALL=POSIX
LFS_TGT=$(uname -m)-lfs-linux-gnu
PATH=/tools/bin:/bin:/usr/bin
export LFS LC_ALL LFS_TGT PATH
EOF
```

```
lfs@dcjumalan_linukka:~$ cat > ~/.bashrc << "EOF"
> set +h
> umask 022
> LFS=/mnt/lfs
> LC_ALL=POSIX
> LFS_TGT=$(uname -m)-lfs-linux-gnu
> PATH=/tools/bin:/bin:/usr/bin
> export LFS LC_ALL LFS_TGT PATH
> EOF
lfs@dcjumalan_linukka:~$
```

Puretaan komentoa auki:

- `set +h` → ottaa pois päältä bashin hash-funktionin. Hash-funktionin ollessa päällä bash käyttää hash-table:a, jotta se muistaa polun suoritettaville tiedostoille eli se ei etsi polkua samalle ohjelmalle joka kerta. Tämä pitää ottaa pois päältä, koska kun tehdään uudet työkalut, bash ei saa muistaa edellistä versiota samasta ohjelmasta, joka sijaitsi jossakin toisessa paikassa.
- `umask 022` → Asetimme käyttäjätiedoston maskin 022:een. 022 varmistaa, että uudet tehdyt tiedostot ja hakemistot ovat kirjoitettavissa ainoastaan omistajalla, mutta niitä pystyy lukemaan ja käyttämään kaikki.
- `LFS=/mnt/lfs` → Asettaa LFS-muuttujan valittuun liitospisteeseen ([mount point](#)).

- `LC_ALL=POSIX` → `LC_ALL`-muuttuja kontrolloi tiettyjen ohjelmien paikan-
nusta, jotta se saa viestit seuraamaan tietyn maan käytäntöjä. Kun `LC_ALL`
asetetaan ([POSIX](#)) sillä varmistetaan, että [chroot](#)-ympäristössä kaikki toimii
niin kuin on tarkoitus.
- `PATH=/tools/bin:/bin:/usr/bin` → Kun `/tools/bin` hakemisto-
polku määritellään polkumuuttujaan ennen normaalia työkalupolkua (`bin`),
shell käyttää asennuksen jälkeen ensisijaisesti kaikkia niitä ohjelmia, jotka
tullaan käänösvaiheessa kääntämään. Tämä komento yhdistettynä siihen,
että hash on otettu pois päältä vähentää riskiä, että isäntäkone käyttää van-
hoja ohjelmia, kun sama ohjelma on saatavilla tekemässämme ympäris-
tössä.

Lopuksi ajetaan source-komento `.bash_profile` -skriptille. Tämä saa aikaan sen,
että nykyinen shell suorittaa äsken tekemämme käyttäjäprofiilin alustuksen.

```
source ~/.bash_profile
```

```
lfs@dcjumalan_linukka:~$ source ~/.bash_profile  
lfs:~$
```

2.8 Tilapäisen järjestelmän rakentaminen

Tässä luvussa asennetaan työkaluja, joita tarvitaan seuraavassa luvussa lopullisen LFS-järjestelmän rakentamiseen. Ensimmäisessä vaiheessa asennetaan uusi isäntäkoneesta riippumaton [työkaluketju](#) (kääntäjä, kokoaja, linkkeri, kirjastoja ja muutama hyödyllinen työkalu). Toisessa vaiheessa käännetään muita tarpeellisia työkaluja käyttäen ensimmäisessä vaiheessa luotua työkaluketjua.

Tämän luvun tiedostot asennetaan `$LFS/tools` -hakemistoon, jotta ne ovat erillään seuraavassa luvussa asennettavista tiedostoista. Paketit, jotka käännetään tässä luvussa ovat tilapäisiä, joten emme halua, että ne saastuttavat pian rakennettavaa LFS-järjestelmää.

Tämän luvun lopputuloksena meillä on tilapäinen osasto täynnä työkaluja, jotka voidaan pitää erillään isäntäjärjestelmästä. Chrootia käyttämällä komennot tulevissa luvuissa suoritetaan uuden ympäristön sisällä. Se takaa puhtaan ja ongelmattoman LFS-järjestelmän rakentamisen.

Pakettien asennusjärjestyksellä on väliä, joten tämän kirjan järjestystä kannattaa noudattaa. Kaikkien asennettavien pakettien tarkoitus ja riippuvaisuussuhteet muihin paketteihin on liitteenä kirjan lopussa. Samassa pakettilistassa on myös lyhyesti selitetty, mitä kukin paketti sisältää; listattu pakettien mahdolliset kotisivut, latauslinkit, latauksen koko ja asentamiseen vaadittava levytila sekä kirjattu jokaisen paketin `md5sum`. [Linkki pakettilistaan](#).

Pakettien koko ja siten myös käännoisaika vaihtelee. Suurin paketti on Glibc. Se vie nykyaikaisilla järjestelmillä noin 20 minuuttia, mutta hitaammilla järjestelmillä se voi viedä kolmekin päivää. Pakettilistasta löytyvä SBU (Standard Build Unit) kertoo kunkin paketin kohdalla, kuinka aikaa vievästä käännöksestä on kyse. SBU on mainittu myös tässä luvussa kunkin paketin kohdalla heti alussa.

SBU toimii seuraavasti. Ensimmäinen tässä kappaleessa käännettävä paketti on Binutils. Aikaan, joka Binutilsin kääntämiseen menee, viitataan SBU:lla. Kuvitellaan paketti, jonka kääntämisaika on 4,5 SBU. Jos järjestelmällä kesti 10 minuuttia

kääntää Binutils, sillä kestää noin 45 minuuttia kääntää tämä 4,5 SBU:n esimerkkipaketti.

SBU ei ole täysin tarkka tai aukoton mittayksikkö, koska siihen vaikuttaa moni asia, kuten esimerkiksi isäntäjärjestelmän GCC-versio. SBU antaa suuntaa antavan ajan, mutta joissakin tapauksissa sen pohjalta arvioitu aika ja todellinen kääntämiseen kulunut aika saattavat erota jopa kymmenillä minuuteilla.

Testien ajaminen tässä luvussa ei ole tarpeellista. Kuten jo aiemmin todettiin, tässä luvussa asennettavat työkalut ovat tilapäisiä ja heivataan menemään, kun niitä ei enää tarvita. Testit voi ajaa huvikseen, jos haluaa, mutta todennäköisistä virheistä ei kannata välittää.

Testilogeja voi verrata LFS:n virallisilla sivuilla oleviin logeihin verratakseen ovatko mahdolliset epäonnistumiset odotettavia: <http://www.linuxfromscratch.org/lfs/build-logs/8.0/>

Tässä vaiheessa on syytä tarkistaa, että `$LFS`-muuttuja toimii:

```
echo $LFS
```

Tulosteeksi pitäisi tulla `/mnt/lfs`.

```
lfs:~$ echo $LFS
/mnt/lfs
lfs:~$
```

Aiemmin lueteltujen isäntäkoneen vaatimusten pitää toteutua:

- `bash` on shell, jota käytetään
- `sh` on symbolinen linkki `bash`:iin
- `usr/bin/awk` on symbolinen linkki `gawk`:iin
- `/usr/bin/yacc` pitää olla linkki bisoniin tai pieni skripti, joka suorittaa bisonin

Nykyisen käytössä olevan shellin saa tietää seuraavalla komennolla (mikäli käytössä oleva shell tukee `--version` -flagia, useimmat tukevat):

```
$SHELL --version
```

```
lfs:~$ $SHELL --version
GNU bash, version 4.2.45(2)-release (x86_64-slackware-linux-gnu)
Copyright (C) 2011 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
lfs:~$ █
```

Ennen kuin jatketaan, on syytä ottaa selville työskentelyalustan nimi. Helpoiten sen saa selville ajamalla `config.guess` -skriptin, joka tulee monien pakettien lähdetiedostojen mukana. Puretaan Binutils ja ajetaan skripti.

```
cd $LFS/sources
tar -xf binutils-<xxx>
```

```
lfs:/mnt/lfs$ cd $LFS/sources
lfs:/mnt/lfs/sources$ tar -xf binutils-2.27.tar.bz2
lfs:/mnt/lfs/sources$ █
```

```
cd binutils-<xxx>
./config.guess
```

```
lfs:/mnt/lfs/sources$ cd binutils-2.27
lfs:/mnt/lfs/sources/binutils-2.27$ ./config.guess
x86_64-pc-linux-gnu
lfs:/mnt/lfs/sources/binutils-2.27$ █
```

64-bittisellä järjestelmällä tuloste on `x86_64-pc-linux-gnu`, kuten kuvassa yllä. Esimerkiksi 32-bittisellä Intel-prosessorilla tuloste olisi `i686-pc-linux-gnu`. Nimi kannattaa painaa mieleen mahdollista myöhempää tarvetta varten.

Lisäksi on syytä olla tietoinen alustan [dynaamisesta linkkeristä](#), johon viitataan usein nimellä `dynamic loader` (ei pidä sekoittaa `standard linkkeri ld:hen`, joka on osa Binutilsia). Glibc:n tarjoama dynaaminen linkkeri löytää ja lataa ohjelman tarvitsemat jaetut kirjastot, valmistelee ohjelman ajamista varten ja sitten ajaa sen. 32-bittisellä Intel-koneella dynaamisen linkkerin nimi on `ld-linux.so.2`. 64-bittisellä järjestelmällä nimi on `ld-linux-x86-64.so.2`.

Pomminvarma tapa ottaa selville dynaamisen linkkerin nimi on tutkia satunnaista binääriä isäntäjärjestelmässä ajamalla seuraava komento:

```
readelf -l <name of binary> | grep interpreter
```

Seuraavaksi vähän teoriaa siitä, miten tämän kappaleen käännösmetodi toimii:

- Muuttamalla hieman työskentelyalustan nimeä (muuttamalla "vendor"-kentän target triplet:iä `LFS_TGT` -muuttujaksi) varmistetaan, että Binutilsin ja GCC:n ensimmäinen käännös tuottaa yhteensopivan ristilinkkerin ja ristikäntäjän. Sen sijaan, että ne tuottaisivat binäärejä toiselle arkkitehtuurille, ristilinkkeri ja ristikäntäjä tuottavat binäärejä, jotka ovat yhteensopivia nykyisen laitteiston kanssa.
- Tilapäiset kirjastot ristikäännetään. Koska [ristikäntäjä](#) ei luonnostaan voi nojata mihinkään isäntäjärjestelmästä, tämä metodi estää mahdollisen kohdejärjestelmän saastumisen vähentämällä sitä mahdollisuutta, että headerit tai kirjastot isäntäjärjestelmästä sekoittuvat uusiin työkaluihin. Ristikääntäminen myös mahdollistaa sekä 32-bittisten että 64-bittisten kirjastojen kääntämisen 64-bittisellä laitteistolla.
- GCC:n sourcen harkittu muokkaaminen kertoo kääntäjälle, mitä kohteen dynaamista linkkeriä käytetään.
- Binutils asentaa kääntäjän ja linkkerin kahteen paikkaan, `/tools/bin` ja `/tools/$LFS_TGT/bin`. Työkalut yhdessä paikassa on kovalinkitetty toiseen. Tärkeä puoli linkkeriä on sen kirjaston hakujärjestys. Yksityiskohtaista infoa asiasta saa `ld`:ltä syöttämällä sille `--verbose` -flagin. Esimerkiksi `ld --verbose | grep SEARCH` tulostaa nykyiset hakupolut ja niiden järjestyksen. `gcc dummy.c -Wl,--verbose 2>&1 | grep succeeded` näyttää kaikki tiedostot, jotka avattiin onnistuneesti linkkauksen aikana.
- Binutilsin jälkeen asennetaan GCC. Seuraavanlaista voi näkyä configuren ajamisen aikana:

```
checking what assembler to use... /tools/i686-lfs-linux-gnu/bin/as
```

```
checking what linker to use... /tools/i686-lfs-linux-gnu/bin/ld
```

- Nämä ovat tärkeitä jo aiemmin mainittujen syiden takia. Se myös demonstroi, että GCC:n configure-skripti ei etsi `PATH`-hakemistoja saadakseen selville, mitä työkaluja käyttää. Kuitenkaan gcc:n ajamisen aikana samoja hakupolkuja ei välttämättä käytetä. Saadakseen selville mitä standard linkeriä gcc käyttää, aja seuraava komento: `gcc -print-prog-name=ld`.
- Yksityiskohtaista infoa saa gcc:ltä syöttämällä sille `-v` -flagin dummy-ohjelman kääntämisen aikana. Esimerkiksi `gcc -v dummy.c` näyttää yksityiskohtaista informaatiota esikääntäjästä, [assemblerista](#) ja kääntämisestä mukaan lukien gcc:n sisäiset hakupolut ja niiden järjestyksen.
- GCC:n jälkeen asennetaan Linux API Headersit. Ne sallivat standardin C-kirjaston (Glibc) linkittyä Linux kernelin tarjoamien ominaisuuksien kanssa.
- Linux API Headersien jälkeen asennetaan Glibc. Tärkeimmät pohdinnat liittyen Glibc:n kääntämiseen liittyvät kääntäjään, binäärityökaluihin ja kernel headerseihin. Kääntäjä ei ole yleensä ongelma, koska Glibc käyttää kääntäjää, joka määrittyy `--host` -parametrilla, joka syötetään configure-skriptille. Binäärityökalut ja kernel headersit ovat vähän monimutkaisempi asia. Sen takia ei kannata ottaa riskejä. Kannattaa käyttää saatavilla olevia configure-kytkimiä oikeiden valintojen pakottamiseen. Configuren ajamisen jälkeen, tarkasta `config.make` -tiedoston sisältö `glibc-build` -hakemistossa. Huomaa `CC="i686-lfs-gnu-gcc"`:n käyttö, joka kontrolloi mitä binäärityökaluja käytetään. Huomaa myös `-nostdinc` ja `-isystem` -lippujen käyttö. Ne kontrolloivat kääntäjän hakupolkua. Nämä asiat korostavat Glibc-paketin tärkeää aspektia: se on hyvin itseriittoista, kun ottaa huomioon, että sillä on oma käännöskoneisto ja yleensä se ei nojaa työkaluketjun oletuksiin.
- Binutilsin toisessa osassa hyödynnetään `--with-lib-path` -configure-kytkintä, jolla kontrolloidaan `ld`:n kirjaston hakupolkua.

- GCC:n toisessa osassa sen sourcet pitää modifioida, jotta GCC käyttää uutta dynaamista linkkeriä. Tämän epäonnistuminen johtaa siihen, että GCC:n ohjelmilla itsellään on isäntäjärjestelmän `/lib` -hakemiston dynaamisen linkkerin nimi sulautettuna niihin. Tämä taistelisi tavoitettamme vastaan päästä isäntäjärjestelmästä eroon. Tästä pisteestä eteenpäin työkaluketju on omavarainen. Loput tämän kappaleen paketit käännetään uutta Glibc:tä vastaan `/tools` -hakemistossa.
- [3.3](#)-kappaleessa siirytään [chroot](#)-ympäristöön. Silloin ensimmäinen asennettava iso paketti on Glibc johtuen sen jo aiemmin mainitusta omavaraisesta luonteesta. Kun tämä Glibc on asennettu `/usr` -hakemistoon, suoritetaan työkaluketjun oletuksien nopea vaihto ja sitten jatketaan lopullisen LFS-kohdejärjestelmän rakentamista.

Käännösohjeita:

- Useihin paketteihin asennetaan korjaus ennen kääntämistä, mutta vain silloin kun korjausta tarvitaan ongelman kiertämiseen. Korjauksia tarvitaan usein sekä tässä että seuraavassa kappaleessa, mutta joskus vain toisessa niistä. Ei siis tarvitse huolestua, jos ohjeet ladatun korjauksen asentamiseen puuttuvat. Korjausta asennettaessa saattaa tulla varoitusviestejä, joissa esiintyy sanat *offset* tai *fuzz*. Näistä varoituksista ei tarvitse välittää, koska korjaus on silti asentunut onnistuneesti.
- Pakettien kääntämisen aikana tulee useita varoitusviestejä, jotka vilahtavat ohi nopeasti. Ne ovat tavallisia ja voi ohittaa turvallisin mielin. Nämä varoitukset koskevat vanhentunutta C tai C++ -syntaksin käyttöä. C:n standardit muuttuvat verrattain useasti ja jotkut paketit käyttävät silti vanhempia standardeja. Se ei ole ongelma, mutta johtaa varoitukseen käännöksen aikana.

Tässä vaiheessa on viimeisen kerran syytä tarkistaa, että `$LFS` -muuttuja on asetettu oikein:

```
echo $LFS
```

Tulosteen pitäisi olla `/mnt/lfs`.

Jokaisen paketin kohdalla toistuva käännösprosessin kulku on tärkeä ymmärtää:

1. Laita kaikki lähdekoodit ja korjaustiedostot hakemistoon, joka on saatavilla [chroot](#)-ympäristössä, kuten `/mnt/lfs/sources`.
ÄLÄ laita sourceja esim. `/mnt/lfs/tools` -hakemistoon.
2. Mene sources-hakemistoon.
3. Toimi jokaisen paketin kohdalla seuraavasti:
 - a. Käytä `tar`-ohjelmaa paketin purkamiseen. Tässä kappaleessa varmista, että olet `lfs`-käyttäjänä, kun purat paketin.
 - b. Vaihda hakemistoon, joka luotiin, kun paketti purettiin.
 - c. Seuraa tämän kirjan ohjeita paketin kääntämiseen.
 - d. Vaihda takaisin sources-hakemistoon.
 - e. Poista purettu lähdehakemisto, ellei toisin ohjeisteta.

Äskeisen ymmärtäminen on siis tärkeää, koska se säästää monilta ongelmilta jatkossa.

2.8.1 Binutils-2.27 osa 1

Arvioitu käännösaika: 1 SBU (Huomioi, että tässä 2.8 -kappaleessa SBU-arvot eivät sisällä testejä.)

Ensimmäiseksi asennetaan Binutils, koska sekä Glibc että GCC suorittavat testejä, joissa Binutilsia tarvitaan.

Äsken otimme selville työskentelyalustasta nimen Binutils-pakettia hyödyntäen. Paketti on siis jo purettu, joten voimme suoraan siirtyä kääntämiseen.

Siirrytään binutils-hakemistoon ja tehdään sen alle uusi `build/` -hakemisto:

```
cd binutils-2.27/  
mkdir -v build
```

```
lfs:/mnt/lfs/sources/binutils-2.27$ mkdir -v build  
mkdir: created directory 'build'  
lfs:/mnt/lfs/sources/binutils-2.27$
```

Jos haluat ottaa hyödyn irti SBU-arvoista, kannattaa mitata, kauan Binutilsiin menee. Laita kaikki komennot time-komennon sisälle seuraavasti: `time { ./configure ... && ... && make install; }`

Siirrytään build-hakemistoon ja valmistellaan paketti kääntämistä varten (ts. ajetaan `configure`-skripti):

```
cd build/  
../configure ---prefix=/tools --with-lib-path=/tools/lib --  
target=$LFS_TGT --disable-nls --disable-werror
```

```

lfs:/mnt/lfs/sources/binutils-2.27/build$ ../configure --prefix=/tools
>          --with-sysroot=$LFS          \
>          --with-lib-path=/tools/lib   \
>          --target=$LFS_TGT            \
>          --disable-nls                 \
>          --disable-werror
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking target system type... x86_64-lfs-linux-gnu
checking for a BSD-compatible install... /bin/ginstall -c
checking whether ln works... yes
checking whether ln -s works... yes
checking for a sed that does not truncate output... /bin/sed
checking for awk... awk

```

Puretaan auki:

- `--prefix=/tools` → Kertoo configure-skriptille, että valmistaa Binutilsin asentamisen `/tools`-hakemistoon
- `--with-sysroot=$LFS` → Tämä asetus kertoo ristikäntämistä varten käännösjärjestelmää etsimään `$LFS`-hakemistosta kohdejärjestelmän kirjastoja tarvittaessa.
- `--with-lib-path=/tools/lib` → Kertoo, mitä kirjastopolkua linkkerin pitäisi käyttää.
- `--target=$LFS_TGT` → Koneen kuvaus on `LFS_TGT` -muuttujassa hie-
man erilainen kuin arvo, jonka `config.guess` -skripti palauttaa. Sen vuoksi
tämä asetus kertoo configure-skriptille säätämään Binutilsin käännösjärjes-
telmää kääntämään ristikäntäjä.
- `--disable-nls` → Tämä asetus poistaa käytöstä kansainvälistämisen,
koska se ei ole tarpeellinen tilapäisille työkaluille.
- `--disable-werror` → Estää kääntämisen loppumisen varoituksiin.

Sitten ajetaan [make](#):

```
make
```

Käännös on nyt valmis. Tavallisesti tässä kohtaa ajettaisiin testit, mutta näin varhaisessa vaiheessa testikehys (Tcl, Expect ja DejaGNU) ei ole vielä paikoillaan. Muutenkaan testien ajamisella vielä tässä vaiheessa ei saavuteta mitään hyötyä.

Jos käännetään x86_64 -järjestelmällä, luodaan symbolinen linkki työkaluketjun eheyden varmistamiseksi:

```
case $(uname -m) in
    x86_64) mkdir -v /tools/lib && ln -sv lib /tools/lib64
;;
esac
```

```
lfs:/mnt/lfs/sources/binutils-2.27/build$ case $(uname -m) in
>   x86_64) mkdir -v /tools/lib && ln -sv lib /tools/lib64 ;;
> esac
mkdir: created directory '/tools/lib'
'/tools/lib64' -> 'lib'
lfs:/mnt/lfs/sources/binutils-2.27/build$
```

Sitten asennetaan:

```
make install
```

```
lfs:/mnt/lfs/sources/binutils-2.27/build$ make install
make[1]: Entering directory `/mnt/lfs/sources/binutils-2.27/build'
/bin/sh ../mkinstalldirs /tools /tools
make[2]: Entering directory `/mnt/lfs/sources/binutils-2.27/build/bfd'
make install-recursive
make[3]: Entering directory `/mnt/lfs/sources/binutils-2.27/build/bfd'
```

Lopuksi poistetaan purettu lähdehakemisto sources-hakemistosta.

```
cd $LFS/sources
rm -rf binutils-2.27
```

```
lfs:/mnt/lfs/sources/binutils-2.27$ cd ..
lfs:/mnt/lfs/sources$ rm -rf binutils-2.27
lfs:/mnt/lfs/sources$
```

2.8.2 GCC-6.3.0 osa 1

Arvioitu käännösaika: 8,4 SBU

Ensin puretaan GCC.

```
cd $LFS/sources
tar xf gcc-<xxx>
```

```
lfs:/mnt/lfs/sources/binutils-2.27/build$ cd $LFS/sources
lfs:/mnt/lfs/sources$ tar xf gcc-6.3.0.tar.bz2
lfs:/mnt/lfs/sources$
```

Sen jälkeen puretaan GCC:n vaatimat muut paketit: GMP, MPFR ja MPC. Ne puretaan GCC:n source-hakemistoon ja uudelleennimetään, jotta GCC automaattisesti käyttää niitä käännöksessä.

```
cd gcc-6.3.0
tar xf ../mpfr-3.1.5.tar.xz
mv -v mpfr-3.1.5 mpfr
tar xf ../gmp-6.1.2.tar.xz
mv -v gmp-6.1.2 gmp
tar xf ../mpc-1.0.3.tar.gz
mv -v mpc-1.0.3 mpc
```

```
lfs:/mnt/lfs/sources$ cd gcc-6.3.0
lfs:/mnt/lfs/sources/gcc-6.3.0$ tar xf ../mpfr-3.1.5.tar.xz
lfs:/mnt/lfs/sources/gcc-6.3.0$ mv -v mpfr-3.1.5 mpfr
'mpfr-3.1.5' -> 'mpfr'
lfs:/mnt/lfs/sources/gcc-6.3.0$ tar xf ../gmp-6.1.2.tar.xz
lfs:/mnt/lfs/sources/gcc-6.3.0$ mv -v gmp-6.1.2 gmp
'gmp-6.1.2' -> 'gmp'
lfs:/mnt/lfs/sources/gcc-6.3.0$ tar xf ../mpc-1.0.3.tar.gz
lfs:/mnt/lfs/sources/gcc-6.3.0$ mv -v mpc-1.0.3 mpc
'mpc-1.0.3' -> 'mpc'
lfs:/mnt/lfs/sources/gcc-6.3.0$
```

Seuraava komento vaihtaa GCC:n dynaamisen oletuslinkkerin paikan siihen, joka on asennettu /tools:iin. Se myös poistaa /usr/include:n GCC:n hakupolulta.

```
for file in \
$(find gcc/config -name linux64.h -o -name linux.h -o -
name sysv4.h)
do
    cp -uv $file{,.orig}
    sed -e 's@/lib\ (64\)\ ??\ (32\)\ ??/ld@/tools&@g' \
        -e 's@/usr@/tools@g' $file.orig > $file
    echo '
#undef STANDARD_STARTFILE_PREFIX_1
#undef STANDARD_STARTFILE_PREFIX_2
#define STANDARD_STARTFILE_PREFIX_1 "/tools/lib/"
#define STANDARD_STARTFILE_PREFIX_2 ""' >> $file
    touch $file.orig
done
```

```
lfs:/mnt/lfs/sources/gcc-6.3.0$ for file in gcc/config/{linux,i386/linux{,64}}.h
> do
>   cp -uv $file{,.orig}
>   sed -e 's@/lib\ (64\)\ ??\ (32\)\ ??/ld@/tools&@g' \
>       -e 's@/usr@/tools@g' $file.orig > $file
>   echo '
> #undef STANDARD_STARTFILE_PREFIX_1
> #undef STANDARD_STARTFILE_PREFIX_2
> #define STANDARD_STARTFILE_PREFIX_1 "/tools/lib/"
> #define STANDARD_STARTFILE_PREFIX_2 ""' >> $file
>   touch $file.orig
> done
'gcc/config/linux.h' -> 'gcc/config/linux.h.orig'
'gcc/config/i386/linux.h' -> 'gcc/config/i386/linux.h.orig'
'gcc/config/i386/linux64.h' -> 'gcc/config/i386/linux64.h.orig'
lfs:/mnt/lfs/sources/gcc-6.3.0$
```

Puretaan auki:

Ensin etsitään kaikki tiedostot gcc/config -hakemistosta, joiden nimi on joko linux.h, linux64.h tai sysv4.h. Kaikki löydetty tiedostot kopioidaan ja niiden loppuun lisätään ".orig". Ensimmäinen sed-lauseke lisää hakemistojen /lib/ld, lib64/ld ja /lib32/ld alkuun /tools. Toinen sed-lauseke korvaa /usr:n [kovakoodatut](#) instanssit. Seuraavaksi lisätään määrittelylauseet, jotka

muokkaavat käynnistystiedoston prefixiä tiedoston loppuun. Lopuksi käytetään touch-komentoa päivittämään aikaleima kopioituissa tiedostoissa. "Cp -u" estää odottamattomat muutokset alkuperäisiin tiedostoihin siinä tapauksessa, että komennot ajetaan epähuomiossa kahdesti.

x86_64 -koneilla asetetaan 64-bittisille kirjastoille oletushakemistonimeksi "lib".

```
case $(uname -m) in
    x86_64)
        sed -e '/m64=/s/lib64/lib/' \
            -i.orig gcc/config/i386/t-linux64
    ;;
esac
```

```
lfs:/mnt/lfs/sources/gcc-6.3.0$ case $(uname -m) in
>     x86_64)
>         sed -e '/m64=/s/lib64/lib/' \
>             -i.orig gcc/config/i386/t-linux64
>     ;;
> esac
lfs:/mnt/lfs/sources/gcc-6.3.0$
```

GCC:n dokumentaatio suosittelee GCC:n kääntämistä build-hakemistoon, joten luodaan sellainen ja siirrytään sinne:

```
cd gcc-6.3.0/
mkdir -v build
cd build/
```

```
lfs:/mnt/lfs/sources/gcc-6.3.0$ mkdir -v build
mkdir: created directory 'build'
lfs:/mnt/lfs/sources/gcc-6.3.0$ cd build/
lfs:/mnt/lfs/sources/gcc-6.3.0/build$
```

Sitten ajetaan configure:

```
../configure --target=$LFS_TGT --prefix=/tools --with-glibc-version=2.11 --with-sysroot=$LFS --with-newlib --without-headers --with-local-prefix=/tools --with-native-system-header-dir=/tools/include --disable-nls --disable-shared --disable-multilib --disable-decimal-float --disable-threads --disable-libatomic --disable-libgomp --disable-libmpx --disable-libquadmath --disable-libssp --disable-libvtv --disable-libstdcxx --enable-languages=c,c++
```

```
lfs:/mnt/lfs/sources/gcc-6.3.0/build$ ../configure
\
> --target=$LFS_TGT
> --prefix=/tools
> --with-glibc-version=2.11
> --with-sysroot=$LFS
> --with-newlib
> --without-headers
> --with-local-prefix=/tools
> --with-native-system-header-dir=/tools/include
> --disable-nls
> --disable-shared
> --disable-multilib
> --disable-decimal-float
> --disable-threads
> --disable-libatomic
> --disable-libgomp
> --disable-libmpx
> --disable-libquadmath
> --disable-libssp
> --disable-libvtv
> --disable-libstdcxx
> --enable-languages=c,c++
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking target system type... x86_64-lfs-linux-gnu
checking for a BSD-compatible install... /bin/ginstall -c
```

Puretaan auki:

- `--with-newlib` → Koska toimiva C-kirjasto ei ole vielä saatavilla, tämä asetus valmistaa, että `inhibit_libc` -vakio on määritelty käännettäessä `libgcc`:tä. Tämä estää sellaisten koodien kääntämisen, jotka vaativat `libc`:tä.
- `--without-headers` → Kun luodaan täydellistä ristikääntäjää, GCC vaatii standardit header-tiedostot, jotka ovat yhteensopivia kohdejärjestelmän

kanssa. Emme tarvitse näitä kuvasutiedostoja. Tämä asetus estää GCC:tä etsimästä niitä.

- `--with-local-prefix=/tools` → Local prefix on paikka järjestelmässä, josta GCC etsii paikallisesti asennettuja include-tiedostoja. Oletus on `/usr/local`. Vaihdetaan se `/tools` -hakemistoon, jotta isäntäjärjestelmän `/usr/local` pysyy pois GCC:n etsintäpolusta.
- `--with-native-system-header-dir=/tools/include` → Oletuksena GCC etsii järjestelmän header-tiedostoja hakemistosta `/usr/include`. Sysroot switchin kanssa se kääntyisi yleensä hakemistoon `$LFS/usr/include`. Kuitenkin seuraavassa kahdessa osiossa asennettavat header-tiedostot menevät hakemistoon `$LFS/tools/include`. Tämä asetus varmistaa, että GCC löytää ne. Seuraavassa GCC:n asennusvaiheessa sama asetus varmistaa, että header-tiedostoja ei löydy isäntäjärjestelmästä.
- `--disable-shared` → Tämä asetus käskee GCC:n linkata sen sisäiset kirjastot staattisesti. Teemme sen välttääksemme mahdollisia ongelmia isäntäjärjestelmän kanssa.
- `--disable-decimal-float`, `--disable-threads`, `--disable-libatomic`, `--disable-libgomp`, `--disable-libmpx`, `--disable-libquadmath`, `--disable-libssp`, `--disable-libvtv`, `--disable-libstdc++` → Nämä asetukset poistavat käytöstä tuen seuraavilta: DFP (Decimal Floating Point) -laajennus, säikeistys, libatomic, libgomp, libmpx, libquadmath, libssp, libvtv ja C++ -standardikirjasto. Nämä ominaisuudet epäonnistuvat käännöksessä, kun käännetään ristikäntäjää, eivätkä ole tarpeellisia ristikäännettäessä tilapäistä libc:tä.
- `--disable-multilib` → x86_64 -järjestelmässä LFS ei vielä tue multilib-konfiguraatiota. Tämä asetus on harmiton x86-järjestelmälle.

- `--enable-languages=c,c++` → Tämä asetus varmistaa, että ainoastaan C ja C++ -kääntäjät käännetään. Ainoastaan näitä kahta kieltä tarvitaan tällä hetkellä.

Seuraavaksi ajetaan make:

```
make
```

Sitten asennetaan:

```
make install
```

Lopuksi poistetaan purettu lähdehakemisto sources-hakemistosta.

```
cd $LFS/sources  
rm -rf gcc-6.3.0
```

```
lfs:/mnt/lfs/sources$ rm -rf gcc-6.3.0  
lfs:/mnt/lfs/sources$
```

2.8.3 Linux-4.9.9 API Headers

Arvioitu käännösaika: alle 0,1 SBU

Ensimmäiseksi puretaan paketti ja siirrytään luotuun hakemistoon.

```
cd $LFS/sources
tar -xf linux-4.9.9.tar.xz
cd linux-4.9.9
```

```
lfs:/mnt/lfs/sources/gcc-6.3.0/build$ cd $LFS/sources
lfs:/mnt/lfs/sources$ tar -xf linux-4.9.9.tar.xz
lfs:/mnt/lfs/sources$ cd linux-4.9.9
lfs:/mnt/lfs/sources/linux-4.9.9$
```

Seuraavaksi varmistetaan, ettei paketissa ole väljähtyneitä tiedostoja.

```
make mrproper
```

```
lfs:/mnt/lfs/sources/linux-4.9.9$ make mrproper
lfs:/mnt/lfs/sources/linux-4.9.9$
```

Puretaan käyttäjälle näkyvät kernelin header-tiedostot lähdehakemistosta. Ne si-
joitetaan paikalliseen välittäjähakemistoon (intermediate directory) ja kopioidaan
tarvittavaan paikkaan, koska extraction-prosessi poistaa kaikki olemassa olevat
tiedostot kohdehakemistosta.

```
make INSTALL_HDR_PATH=dest headers_install
cp -rv dest/include/* /tools/include
```

```
lfs:/mnt/lfs/sources/linux-4.9.9$ make INSTALL_HDR_PATH=dest headers_install
CHK    include/generated/uapi/linux/version.h
UPD    include/generated/uapi/linux/version.h
HOSTCC scripts/basic/fixdep
WRAP   arch/x86/include/generated/asm/clkdev.h
WRAP   arch/x86/include/generated/asm/cputime.h
WRAP   arch/x86/include/generated/asm/dma-contiguous.h
WRAP   arch/x86/include/generated/asm/early_ioremap.h
WRAP   arch/x86/include/generated/asm/early_printk.h
```

Toisella komennolla pitäisi tulla tämän näköistä tulostetta:

```
'dest/include/asm/debugreg.h' -> '/tools/include/asm/debugreg.h'  
'dest/include/asm/unistd_64.h' -> '/tools/include/asm/unistd_64.h'  
'dest/include/asm/mce.h' -> '/tools/include/asm/mce.h'  
'dest/include/asm/swab.h' -> '/tools/include/asm/swab.h'  
'dest/include/asm/hw_breakpoint.h' -> '/tools/include/asm/hw_breakpoint.h'  
'dest/include/asm/sockios.h' -> '/tools/include/asm/sockios.h'
```

Lopuksi poistetaan purettu lähdehakemisto.

```
cd ..  
rm -rf linux-4.9.9
```

```
lfs:/mnt/lfs/sources/linux-4.9.9$ cd ..  
lfs:/mnt/lfs/sources$ rm -rf linux-4.9.9  
lfs:/mnt/lfs/sources$
```

2.8.4 Glibc-2.25

Arvioitu käännösaika: 4,1 SBU

Ensimmäiseksi puretaan glibc-paketti ja siirrytään luotuun hakemistoon.

```
cd $LFS/sources
tar -xf glibc-2.25.tar.xz
cd glibc-2.25/
```

```
lfs:/mnt/lfs/sources$ tar -xf glibc-2.25.tar.xz
lfs:/mnt/lfs/sources$ cd glibc-2.25
lfs:/mnt/lfs/sources/glibc-2.25$
```

Sen jälkeen luodaan build-hakemisto ja siirrytään sinne.

```
mkdir build
cd build/
```

```
lfs:/mnt/lfs/sources/glibc-2.25$ mkdir build
lfs:/mnt/lfs/sources/glibc-2.25$ cd build/
lfs:/mnt/lfs/sources/glibc-2.25/build$
```

Sen jälkeen ajetaan configure.

```
../configure \
  --prefix=/tools \
  --host=$LFS_TGT \
  --build=$(../scripts/config.guess) \
  --enable-kernel=2.6.32 \
  --with-headers=/tools/include \
  libc_cv_forced_unwind=yes \
  libc_cv_c_cleanup=yes
```

```

lfs:/mnt/lfs/sources/glibc-2.25/build$ ../configure
> --prefix=/tools \
> --host=$LFS_TGT \
> --build=$(../scripts/config.guess) \
> --enable-kernel=2.6.32 \
> --with-headers=/tools/include \
> libc_cv_forced_unwind=yes \
> libc_cv_c_cleanup=yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-lfs-linux-gnu
checking for x86_64-lfs-linux-gnu-gcc... x86_64-lfs-linux-gnu-gcc
checking for suffix of object files... o

```

Puretaan auki:

- `--host=$LFS_TGT, --build=$(../scripts/config.guess)` → Näiden asetusten yhdistelmän seuraus on, että Glibc:n käännösjärjestelmä konfiguroi itsensä ristikäntämiseen, käyttäen ristilinkkeriä ja ristikäntäjää `/tools-hakemistossa`.
- `--enable-kernel=2.6.32` → Tämä asetus käskää Glibc:n kääntää kirjasto 2.6.32-version ja sitä myöhempien Linux kernel -versioiden tuella.
- `--with-headers=/tools/include` → Tämä asetus käskää Glibc:n kääntää itsensä äskettäin `/tools-hakemistoon` asennettuja header-tiedostoja vasten, jotta se tietää tarkalleen mitkä ominaisuudet kernelillä on ja voi optimoida itse sen mukaisesti.
- `libc_cv_forced_unwind=yes` → Luvussa [2.8.1 Binutils-2.27 osa 1](#) asennettu linkkeri ristikäännettiin, joten sitä ei voida käyttää ennen kuin Glibc on asennettu. Tämä tarkoittaa sitä, että `configure-testi force-unwind` -tuelle epäonnistuu, koska se tarvitsee toimivan linkkerin. `libc_cv_forced_unwind=yes` -muuttuja informoi `configurea`, että `force-unwind` -tuki on saatavissa ilman, että sen tarvitsee ajaa testiä.
- `libc_cv_c_cleanup=yes` → Samaan tapaan tämä asetus kertoo `configure-skriptille`, että testi ohitetaan ja 'C cleanup handling support' konfiguroidaan.

Tässä vaiheessa seuraavanlainen varoitus saattaa ilmaantua:

```
configure: WARNING:
*** These auxiliary programs are missing or
*** incompatible versions: msgfmt
*** some features will be disabled.
*** Check the INSTALL file for required versions.
```

Olematon tai epäsopiva msgfmt on harmiton. msgfmt on osa Gettext-pakettia, joka isäntäkoneen pitäisi tarjota.

Sitten ajetaan make ja asennetaan.

```
make
make install
```

Tässä vaiheessa on hyvä taas tarkastaa toolchainin eheys. Se tehdään seuraavilla komennoilla:

```
echo 'int main(){}' > dummy.c
$LFS_TGT-gcc dummy.c
readelf -l a.out | grep ': /tools'
```

```
lfs:/mnt/lfs/sources/glibc-2.25/build$ echo 'int main(){}' > dummy.c
lfs:/mnt/lfs/sources/glibc-2.25/build$ $LFS_TGT-gcc dummy.c
lfs:/mnt/lfs/sources/glibc-2.25/build$ readelf -l a.out | grep ': /tools'
[Requesting program interpreter: /tools/lib64/ld-linux-x86-64.so.2]
lfs:/mnt/lfs/sources/glibc-2.25/build$
```

Jos kaikki toimii niin kuin pitääkin, viimeisen komennon tuloste on seuraava:

```
[Requesting program interpreter: /tools/lib/ld-linux.so.2]
```

64-bittisillä koneilla tuloste on seuraava:

```
[Requesting program interpreter: /tools/lib64/ld-linux-x86-64.so.2]
```

Mikäli tuloste on jotain muuta tai sitä ei ole lainkaan, ongelma täytyy jäljittää ja korjata ennen kuin voi siirtyä eteenpäin.

Jos kaikki on hyvin, poistetaan testitiedostot:

```
rm -v dummy.c a.out
```

```
lfs:/mnt/lfs/sources/glibc-2.25/build$ rm -v dummy.c a.out  
removed 'dummy.c'  
removed 'a.out'  
lfs:/mnt/lfs/sources/glibc-2.25/build$ █
```

Lopuksi poistetaan purettu hakemisto:

```
cd $LFS/sources  
rm -rf glibc-2.25
```

2.8.5 LIBSTDC++-6.3.0

<i>Arvioitu käännösaika: 0,4 SBU</i>

Libstdc on osa GCC:tä, joten ensin siirrytään sources-hakemistoon ja puretaan gcc-paketti.

```
cd $LFS/sources
tar -xf gcc-6.3.0.tar.bz2
```

```
lfs:~$ cd $LFS/sources
lfs:/mnt/lfs/sources$ tar -xf gcc-6.3.0.tar.bz2
lfs:/mnt/lfs/sources$ █
```

Siirrytään luotuun gcc-hakemistoon ja tehdään build-hakemisto.

```
cd gcc-6.3.0
mkdir -v build
```

```
lfs:/mnt/lfs/sources$ cd gcc-6.3.0
lfs:/mnt/lfs/sources/gcc-6.3.0$ mkdir -v build
mkdir: created directory 'build'
lfs:/mnt/lfs/sources/gcc-6.3.0$ █
```

Siirrytään build-hakemistoon ja tehdään configure:

```
cd build/
../libstdc++-v3/configure \
    --host=$LFS_TGT \
    --prefix=/tools \
    --disable-multilib \
    --disable-nls \
    --disable-libstdcxx-threads \
    --disable-libstdcxx-pch \
```

```
--with-gxx-include-dir=/tools/$LFS_TGT/in-  
clude/c++/6.3.0
```

```
lfs:/mnt/lfs/sources/gcc-6.3.0$ cd build/  
lfs:/mnt/lfs/sources/gcc-6.3.0/build$ ../libstdc++-v3/configure \\  
> --host=$LFS_TGT \\  
> --prefix=/tools \\  
> --disable-multilib \\  
> --disable-nls \\  
> --disable-libstdc++-threads \\  
> --disable-libstdc++-pch \\  
> --with-gxx-include-dir=/tools/$LFS_TGT/include/c++/6.3.0  
configure: WARNING: If you wanted to set the --build type, don't use --host.  
If a cross compiler is detected then cross compile mode will be used.  
checking build system type... x86_64-pc-linux-gnu  
checking host system type... x86_64-lfs-linux-gnu  
checking target system type... x86_64-lfs-linux-gnu
```

Puretaan auki:

- `--host=...` → Osoittaa käyttämään vasta luomaamme ristikäntäjää `/usr/bin` -hakemistossa olevan sijaan.
- `--disable-libstdc++-threads` → Koska emme ole vielä kääntäneet C threads -kirjastoa, C++ threads:a ei voi myöskään kääntää.
- `--disable-libstdc++-pch` → Estää valmiiksi koottujen tiedostojen, joita ei tässä vaiheessa tarvita, asentamisen.
- `--with-gxx-include-dir=/tools/$LFS_TGT/include/c++/6.2.0` → Täältä C++ -kääntäjä hakee oletuksena include-tiedostoja. Normaalisti kyseinen informaatio automaattisesti annetaan Libstdc++ -configureasetuksille ylätasoon hakemistosta. Tässä tapauksessa tämä informaatio pitää erikseen antaa.

Seuraavaksi ajetaan make ja asennetaan.

```
make  
make install
```

Lopuksi poistetaan sources-hakemistosta purettu lähdehakemisto.

```
cd $LFS/sources  
rm -rf gcc-6.3.0
```

```
lfs:/mnt/lfs/sources/gcc-6.3.0/build$ cd $LFS/sources  
lfs:/mnt/lfs/sources$ rm -rf gcc-6.3.0  
lfs:/mnt/lfs/sources$ █
```

2.8.6 Binutils-2.27 osa 2

Arvioitu käännösaika: 1,1 SBU

Jos tämä käännös epäonnistuu, jokin on mennyt pieleen aiemmin [Binutilsissa](#), [GCC:ssä](#) tai [Glibc:ssä](#).

Ensin puretaan Binutils-paketti ja siirrytään luotuun hakemistoon.

```
tar -xf binutils-2.27.tar.bz2
cd binutils-2.27
```

```
lfs:/mnt/lfs/sources$ tar -xf binutils-2.27.tar.bz2
lfs:/mnt/lfs/sources$ cd binutils-2.27
lfs:/mnt/lfs/sources/binutils-2.27$
```

Seuraavaksi luodaan build-hakemisto ja siirrytään sinne.

```
mkdir -v build
cd build/
```

```
lfs:/mnt/lfs/sources/binutils-2.27$ mkdir -v build
mkdir: created directory 'build'
lfs:/mnt/lfs/sources/binutils-2.27$ cd build/
lfs:/mnt/lfs/sources/binutils-2.27/build$
```

Ajetaan configure:

```
CC=$LFS_TGT-gcc \
AR=$LFS_TGT-ar \
RANLIB=$LFS_TGT-ranlib \
../configure \
--prefix=/tools \
--disable-nls \
--disable-werror \
```

```
--with-lib-path=/tools/lib \  
--with-sysroot
```

Puretaan auki:

- `CC=$LFS_TGT-gcc AR=$LFS_TGT-ar RANLIB=$LFS_TGT-ranlib` → Koska tämä on Binutilsin natiivi käännös, näiden muuttujien asettaminen varmistaa, että käännösjärjestelmä käyttää ristikäntäjää ja työkaluja isäntäjärjestelmän vastaavien sijaan.
- `--with-lib-path=/tools/lib` → Tämä käskee configure-skriptiä tarkentamaan kirjastohakupolkua Binutilsin kääntämisen aikana. Se johtaa siihen, että `/tools/lib` annetaan linkkerille. Tämä estää linkkeriä kahlaamasta läpi isäntäjärjestelmän kirjastohakemistoja.
- `--with-sysroot` → Tämä asetus sallii linkkerin löytää jaettuja objekteja, joita muut linkkerin komentorivin jaetut objektit tarvitsevat. Ilman tätä ominaisuutta joidenkin pakettien käännös saattaa epäonnistua joillakin isännillä.

```
lfs:/mnt/lfs/sources/binutils-2.27/build$ CC=$LFS_TGT-gcc \  
> AR=$LFS_TGT-ar \  
> RANLIB=$LFS_TGT-ranlib \  
> ../configure \  
> --prefix=/tools \  
> --disable-nls \  
> --disable-werror \  
> --with-lib-path=/tools/lib \  
> --with-sysroot \  
checking build system type... x86_64-pc-linux-gnu \  
checking host system type... x86_64-pc-linux-gnu \  
checking target system type... x86_64-pc-linux-gnu
```

Seuraavaksi make ja asennus:

```
make  
make install
```

```
lfs:/mnt/lfs/sources/binutils-2.27/build$ make install  
make[1]: Entering directory `/mnt/lfs/sources/binutils-2.27/build'  
/bin/sh ../mkinstalldirs /tools /tools  
make[2]: Entering directory `/mnt/lfs/sources/binutils-2.27/build/bfd'  
make install-recursive  
make[3]: Entering directory `/mnt/lfs/sources/binutils-2.27/build/bfd'
```

Sitten valmistellaan linkkeri seuraavan kappaleen "Re-adjusting" -vaihetta varten:

```
make -C ld clean
make -C ld LIB_PATH=/usr/lib:/lib
cp -v ld/ld-new /tools/bin
```

Puretaan auki:

- `-C ld clean` → Käskee ohjelman poistaa kaikki käännetyt tiedostot ld-alahakemistossa.
- `-C ld LIB_PATH=/usr/lib:/lib` → Tämä asetus kääntää kaiken ld-alahakemistossa uudelleen. `LIB_PATH` Makefile -muuttujan määrittäminen komentorivillä sallii tilapäisten työkalujen oletusarvon ohittamisen ja osoittaa sen oikealle lopulliselle polulle. Tämän muuttujan arvo määrittää linkkerin oletuskirjastohakupolun. Tätä valmistelua hyödynnetään seuraavassa kappaleessa.

```
lfs:/mnt/lfs/sources/binutils-2.27/build$ make -C ld clean
make: Entering directory `/mnt/lfs/sources/binutils-2.27/build/ld'
Making clean in po
make[1]: Entering directory `/mnt/lfs/sources/binutils-2.27/build/ld/po'
rm -f core core.* *.pox ld.po *.old.po cat-id-tbl.tmp
rm -fr *.o
make[1]: Leaving directory `/mnt/lfs/sources/binutils-2.27/build/ld/po'
```

```
lfs:/mnt/lfs/sources/binutils-2.27/build$ make -C ld LIB_PATH=/usr/lib:/lib
make: Entering directory `/mnt/lfs/sources/binutils-2.27/build/ld'
(echo "/* This file is automatically generated.  DO NOT EDIT! */";\
for f in `echo " " eelf_x86_64.o eelf32_x86_64.o eelf_i386.o eelf_iamcu.o ei386linux.o\
eelf_llom.o eelf_klom.o ""`\
| sed -e 's/ e/ ld/g' -e 's/ ld/ /g' -e 's/[.]o//g`; do \
echo "extern ld emulation xfer type ld ${f} emulation:"; \
```

```
lfs:/mnt/lfs/sources/binutils-2.27/build$ cp -v ld/ld-new /tools/bin
'ld/ld-new' -> '/tools/bin/ld-new'
lfs:/mnt/lfs/sources/binutils-2.27/build$
```

Lopuksi poistetaan purettu hakemisto.

```
cd $LFS/sources  
rm -rf binutils-2.27
```

```
lfs:/mnt/lfs/sources/binutils-2.27/build$ cd $LFS/sources  
lfs:/mnt/lfs/sources$ rm -rf binutils-2.27  
lfs:/mnt/lfs/sources$ █
```

2.8.7 GCC-6.3.0 osa 2

Arvioitu käännösaika: 11 SBU

Ensin puretaan gcc-paketti ja siirrytään luotuun hakemistoon:

```
tar -xf gcc-6.3.0.tar.bz2
cd gcc-6.3.0
```

```
lfs:/mnt/lfs/sources$ tar -xf gcc-6.3.0.tar.bz2
lfs:/mnt/lfs/sources$ cd gcc-6.3.0
lfs:/mnt/lfs/sources/gcc-6.3.0$
```

GCC:n ensimmäisessä asennuksessa asennettiin joitakin sisäisiä header-tiedostoja. Tavallisesti yksi niistä, `limits.h`, vuorostaan pitää sisällään vastaavan järjestelmän `limits.h` -headerin, tässä tapauksessa `/tools/include/limits.h`. GCC:n ensimmäisen käännöksen aikana `/tools/include/limits.h` ei ollut olemassa, joten internal header, jonka GCC asensi on vajaa, omavarainen tiedosto ja se ei sisällä järjestelmän headerin laajennettuja ominaisuuksia. Se oli tarpeenmukainen käännettäessä tilapäistä `libc`:tä, mutta tämä GCC:n kääntäminen tarvitsee täydellisen internal headerin. Luodaan täydellinen versio internal headerista käyttämällä komentoa, joka on identtinen siihen, mitä GCC:n käännös tavallisesti käyttää:

```
cat gcc/limitx.h gcc/glimits.h gcc/limity.h > \
  `dirname $($LFS_TGT-gcc -print-libgcc-file-name)'/include-fixed/limits.h
```

```
lfs:/mnt/lfs/sources/gcc-6.3.0$ cat gcc/limitx.h gcc/glimits.h gcc/limity.h > \
> `dirname $($LFS_TGT-gcc -print-libgcc-file-name)'/include-fixed/limits.h
lfs:/mnt/lfs/sources/gcc-6.3.0$
```

Jälleen vaihdetaan GCC:n oletus dynaamisen linkkerin sijaintia. Käytetään sitä, joka on asennettu `/tools` -hakemistoon.

```
for file in gcc/config/{linux,i386/linux{,64}}.h
do
    cp -uv $file{,.orig}
    sed -e 's@/lib\ (64\)\?\ (32\)\?\ /ld@/tools&@g' \
        -e 's@/usr@/tools@g' $file.orig > $file
    echo '
#undef STANDARD_STARTFILE_PREFIX_1
#undef STANDARD_STARTFILE_PREFIX_2
#define STANDARD_STARTFILE_PREFIX_1 "/tools/lib/"
#define STANDARD_STARTFILE_PREFIX_2 ""' >> $file
    touch $file.orig
done
```

```
lfs:/mnt/lfs/sources/gcc-6.3.0$ for file in gcc/config/{linux,i386/linux{,64}}.h
> do
>   cp -uv $file{,.orig}
>   sed -e 's@/lib\ (64\)\?\ (32\)\?\ /ld@/tools&@g' \
>       -e 's@/usr@/tools@g' $file.orig > $file
>   echo '
> #undef STANDARD_STARTFILE_PREFIX_1
> #undef STANDARD_STARTFILE_PREFIX_2
> #define STANDARD_STARTFILE_PREFIX_1 "/tools/lib/"
> #define STANDARD_STARTFILE_PREFIX_2 ""' >> $file
>   touch $file.orig
> done
'gcc/config/linux.h' -> 'gcc/config/linux.h.orig'
'gcc/config/i386/linux.h' -> 'gcc/config/i386/linux.h.orig'
'gcc/config/i386/linux64.h' -> 'gcc/config/i386/linux64.h.orig'
lfs:/mnt/lfs/sources/gcc-6.3.0$
```

Jos käännetään `x86_64` -järjestelmällä, vaihdetaan 64-bittisten kirjastojen oletus-hakemistonimi `"lib"`:in.

```
case $(uname -m) in
    x86_64)
        sed -e '/m64=/s/lib64/lib/' \
            -i.orig gcc/config/i386/t-linux64

        ;;
esac
```

```
lfs:/mnt/lfs/sources/gcc-6.3.0$ case $(uname -m) in
>   x86_64)
>       sed -e '/m64=/s/lib64/lib/' \
>           -i.orig gcc/config/i386/t-linux64
>   ;;
> esac
lfs:/mnt/lfs/sources/gcc-6.3.0$
```

Sitten puretaan mpfr, gmp ja mpc samalla tavalla kuin ensimmäisen GCC:n asennuksen yhteydessä.

```
tar -xf ../mpfr-3.1.5.tar.xz
mv -v mpfr-3.1.5 mpfr
tar -xf ../gmp-6.1.2.tar.xz
mv -v gmp-6.1.2 gmp
tar -xf ../mpc-1.0.3.tar.gz
mv -v mpc-1.0.3 mpc
```

```
lfs:/mnt/lfs/sources/gcc-6.3.0$ tar -xf ../mpfr-3.1.5.tar.xz
lfs:/mnt/lfs/sources/gcc-6.3.0$ mv -v mpfr-3.1.5 mpfr
'mpfr-3.1.5' -> 'mpfr'
lfs:/mnt/lfs/sources/gcc-6.3.0$ tar -xf ../gmp-6.1.2.tar.xz
lfs:/mnt/lfs/sources/gcc-6.3.0$ mv -v gmp-6.1.2 gmp
'gmp-6.1.2' -> 'gmp'
lfs:/mnt/lfs/sources/gcc-6.3.0$ tar -xf ../mpc-1.0.3.tar.gz
lfs:/mnt/lfs/sources/gcc-6.3.0$ mv -v mpc-1.0.3 mpc
'mpc-1.0.3' -> 'mpc'
lfs:/mnt/lfs/sources/gcc-6.3.0$
```

Sen jälkeen luodaan uusi build-hakemisto ja siirrytään sinne:

```
mkdir -v build
cd build/
```

```
lfs:/mnt/lfs/sources/gcc-6.3.0$ mkdir -v build
mkdir: created directory 'build'
lfs:/mnt/lfs/sources/gcc-6.3.0$ cd build/
lfs:/mnt/lfs/sources/gcc-6.3.0/build$
```

Build-hakemistossa ajetaan configure:

```
CC=$LFS_TGT-gcc \
CXX=$LFS_TGT-g++ \
AR=$LFS_TGT-ar \
RANLIB=$LFS_TGT-ranlib \

../configure \
  --prefix=/tools \
  --with-local-prefix=/tools \
  --with-native-system-header-dir=/tools/include \
  --enable-languages=c,c++ \
  --disable-libstdcxx-pch \
  --disable-multilib \
  --disable-bootstrap \
  --disable-libgomp \
```

```
lfs:/mnt/lfs/sources/gcc-6.3.0/build$ CC=$LFS_TGT-gcc \
> CXX=$LFS_TGT-g++ \
> AR=$LFS_TGT-ar \
> RANLIB=$LFS_TGT-ranlib \
> ../configure \
>   --prefix=/tools \
>   --with-local-prefix=/tools \
>   --with-native-system-header-dir=/tools/include \
>   --enable-languages=c,c++ \
>   --disable-libstdcxx-pch \
>   --disable-multilib \
>   --disable-bootstrap \
>   --disable-libgomp \
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking target system type... x86_64-pc-linux-gnu
checking for a BSD compatible install... /bin/install -c
```

Puretaan auki:

- `--enable-languages=c,c++` → Tämä asetus varmistaa, että sekä C että C++ -kääntäjät käännetään.
- `--disable-libstdcxx-pch` → Älä käännä esikäännettyä header-tiedostoa (PCH = pre-compiled header) libstdc++ varten. Se vie paljon tilaa emmekä tarvitse sitä.
- `--disable-bootstrap` → Oletuksena GCC:lle tehdään "bootstrap" -asennus. Se kääntää GCC:n monta kertaa normaalin yhden kerran sijaan. Se käyttää ensimmäisellä kierroksella käännettyjä ohjelmia kääntääkseen itsensä toisen kerran ja jälleen kolmannen. Toista ja kolmatta iterointia verrataan keskenään varmistaakseen, että ohjelma kykenee jäljentää itsensä virheettömästi. Se myös tarkoittaa, että ohjelma on käännetty onnistuneesti. LFS:n käännösmetodin pitäisi tarjota eheä kääntäjä ilman tarvetta tehdä bootstrap joka kerta.

Sitten tehdään make ja asennus normaalisti:

```
make
make install
```

Seuraavaksi luodaan [symbolinen linkki](#). Monet ohjelmat ajavat CC:n GCC:n sijaan. CC:tä käytetään, kun halutaan pitää ohjelmat käytettävänä kaikenlaisissa UNIX-järjestelmissä, joissa GNU C -kääntäjää ei ole aina asennettu. CC:n ajaminen jättää adminille vapauden päättää, minkä C-kääntäjän asentaa.

```
ln -sv gcc /tools/bin/cc
```

```
lfs:/mnt/lfs/sources/gcc-6.3.0/build$ ln -sv gcc /tools/bin/cc
'/tools/bin/cc' -> 'gcc'
lfs:/mnt/lfs/sources/gcc-6.3.0/build$
```

Tässä vaiheessa on hyvä taas tarkastaa työkaluketjun eheys.

```
echo 'int main(){}' > dummy.c
cc dummy.c
readelf -l a.out | grep ': /tools'
```

```
lfs:/mnt/lfs/sources/gcc-6.3.0/build$ echo 'int main(){}' > dummy.c
lfs:/mnt/lfs/sources/gcc-6.3.0/build$ cc dummy.c
lfs:/mnt/lfs/sources/gcc-6.3.0/build$ readelf -l a.out | grep ': /tools'
      [Requesting program interpreter: /tools/lib64/ld-linux-x86-64.so.2]
lfs:/mnt/lfs/sources/gcc-6.3.0/build$
```

Jos kaikki toimii niin kuin pitääkin, viimeisen komennon tuloste on seuraava:

```
[Requesting program interpreter: /tools/lib/ld-linux.so.2]
```

64-bittisillä koneilla tuloste on seuraava:

```
[Requesting program interpreter: /tools/lib64/ld-linux-x86-64.so.2]
```

Mikäli tuloste on jotain muuta tai sitä ei ole lainkaan, *ongelma täytyy jäljittää ja korjata ennen kuin voi siirtyä eteenpäin*.

Jos kaikki on hyvin, poistetaan testitiedostot:

```
rm -v dummy.c a.out
```

```
lfs:/mnt/lfs/sources/gcc-6.3.0/build$ rm -v dummy.c a.out
removed 'dummy.c'
removed 'a.out'
lfs:/mnt/lfs/sources/gcc-6.3.0/build$
```

Lopuksi poistetaan sources-hakemistosta purettu lähdehakemisto:

```
cd $LFS/sources
rm -rf gcc-6.3.0
```

```
lfs:/mnt/lfs/sources/gcc-6.3.0/build$ cd $LFS/sources
lfs:/mnt/lfs/sources$ rm -rf gcc-6.3.0
lfs:/mnt/lfs/sources$
```

2.8.8 TCL-core-8.6.6

Arvioitu käännösaika: 0,4 SBU

Asennettavat ohjelmat: *tclsh* (linkki *tclsh8.6*) ja *tclsh8.6*

Asennettavat kirjastot: *libtcl8.6.so*, *libtclstub8.6.a*

Tämä paketti ja kolme seuraavaa (Expect, DejaGNU ja Check) asennetaan GCC:n, Binutilsin ja joidenkin muidenkin pakettien testaamista varten.

Ensin puretaan tcl-core -paketti.

```
tar -xf tcl-core8.6.6-src.tar.gz
```

```
lfs:/mnt/lfs/sources$ tar -xf tcl-core8.6.6-src.tar.gz
lfs:/mnt/lfs/sources$
```

Sen jälkeen siirrytään unix-hakemistoon ja ajetaan configure.

```
cd tcl8.6.6/unix/
./configure --prefix=/tools
```

```
lfs:/mnt/lfs/sources$ cd tcl8.6.6/unix/
lfs:/mnt/lfs/sources/tcl8.6.6/unix$ ./configure --prefix=/tools
checking whether to use symlinks for manpages... no
checking whether to compress the manpages... no
checking whether to add a package name suffix for the manpages... no
checking for gcc... gcc
checking for C compiler default output file name... a.out
```

Sitten ajetaan make.

```
make
```

Seuraavaksi ajetaan testit. Tämä vaihe ei ole pakollinen ja vie jonkin verran aikaa, joten kannattaa harkita tekeekö sitä ollenkaan. Testissä saattaa ilmaantua virheitä, mutta tässä vaiheessa niistä ei tarvitse huolestua.

```
TZ=UTC make test
```

TZ=UTC -parametri asettaa aikavyöhykkeeksi Coordinated Universal Time (UTC), mutta vain testin ajaksi.

```
Tests ended at Sat Mar 18 13:36:15 UTC 2017
all.tcl:      Total    31259   Passed  29965   Skipped 1294   Failed   0
Sourced 148 Test Files.
Number of tests skipped for each constraint:
          9      !ieeeFloatingPoint
          3      asyncPipeChan
```

Seuraavaksi asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/tcl8.6.6/unix$ make install
Installing libtcl8.6.so to /tools/lib/
Installing tclsh as /tools/bin/tclsh8.6
Installing tclConfig.sh to /tools/lib/
Installing tclOOConfig.sh to /tools/lib/
Installing libtclstub8.6.a to /tools/lib/
```

Muokataan äsken `make install` -komennolla asennettu kirjasto sellaiseksi, että siihen voi kirjoittaa (writable), jotta debuggaussymbolit voidaan poistaa myöhemmin.

```
chmod -v u+w /tools/lib/libtcl8.6.so
```

```
lfs:/mnt/lfs/sources/tcl8.6.6/unix$ chmod -v u+w /tools/lib/libtcl8.6.so
mode of '/tools/lib/libtcl8.6.so' changed from 0555 (r-xr-xr-x) to 0755 (rwxr-xr-x)
lfs:/mnt/lfs/sources/tcl8.6.6/unix$
```

Seuraavaksi asennetaan Tcl:n header-tiedostot. Seuraava asennettava paketti, Expect, vaatii ne.

```
make install-private-headers
```

```
lfs:/mnt/lfs/sources/tcl8.6.6/unix$ make install-private-headers
Installing private header files to /tools/include/
lfs:/mnt/lfs/sources/tcl8.6.6/unix$
```

Luodaan vielä tarpeellinen symbolinen linkki.

```
ln -sv tclsh8.6 /tools/bin/tclsh
```

```
lfs:/mnt/lfs/sources/tcl8.6.6/unix$ ln -sv tclsh8.6 /tools/bin/tclsh  
'/tools/bin/tclsh' -> 'tclsh8.6'  
lfs:/mnt/lfs/sources/tcl8.6.6/unix$ █
```

Lopuksi poistetaan sources-hakemistosta purettu lähdehakemisto.

```
cd $LFS/sources  
rm -rf tcl8.6.6
```

```
lfs:/mnt/lfs/sources/tcl8.6.6/unix$ cd $LFS/sources  
lfs:/mnt/lfs/sources$ rm -rf tcl8.6.6  
lfs:/mnt/lfs/sources$ █
```

2.8.9 Expect-5.45

Arvioitu käännösaika: 0,1 SBU

Asennettava ohjelma: expect

Asennettava kirjasto: libexpect-5.45.so

Ensin puretaan expect-paketti ja siirrytään luotuun hakemistoon.

```
tar xf expect5.45.tar.gz
cd expect5.45
```

```
lfs:/mnt/lfs/sources$ tar xf expect5.45.tar.gz
lfs:/mnt/lfs/sources$ cd expect5.45
lfs:/mnt/lfs/sources/expect5.45$
```

Seuraavaksi pakotetaan Expectin configure-skripti käyttämään /bin/stty -hakemistoa /usr/local/bin/stty sijaan. Tämä varmistaa, että testityökalut pysyvät eheinä työkaluketjun viimeistä käännöstä varten.

```
cp -v configure{,.orig}
sed 's:/usr/local/bin:/bin:' configure.orig > configure
```

```
lfs:/mnt/lfs/sources/expect5.45$ cp -v configure{,.orig}
'configure' -> 'configure.orig'
lfs:/mnt/lfs/sources/expect5.45$ sed 's:/usr/local/bin:/bin:' configure.orig > configure
lfs:/mnt/lfs/sources/expect5.45$
```

Seuraavaksi ajetaan configure.

```
./configure --prefix=/tools \
            --with-tcl=/tools/lib \
            --with-tclinclude=/tools/include
```

```
lfs:/mnt/lfs/sources/expect5.45$ ./configure --prefix=/tools \
> --with-tcl=/tools/lib \
> --with-tclinclude=/tools/include
checking for correct TEA configuration... ok (TEA 3.9)
configure: configuring expect 5.45
checking for Tcl configuration... found /tools/lib/tclConfig.sh
checking for existence of /tools/lib/tclConfig.sh... loading
```

Puretaan auki:

- `--with-tcl=/tools/lib` → Varmistaa, että configure-skripti löytää Tcl:n asennuksen tilapäisten työkalujen hakemistosta (/tools/lib) sen sijaan, että löytäisi mahdollisesti olemassa olevan asennuksen isäntäkoneelta.
- `--with-tclinclude=/tools/include` → Kertoo Expectille, mistä löytää Tcl:n sisäiset header-tiedostot. Tätä asetusta käyttämällä välttää tilanteet, joissa configure epäonnistuu, koska se ei automaattisesti löydä Tcl:n header-tiedostoja.

Seuraavaksi ajetaan make.

make

```
lfs:/mnt/lfs/sources/expect5.45$ make  
gcc -DPACKAGE_NAME=\"expect\" -DPACKAGE_TARNAME=\"expect\" -DPACKAGE_VERSION=\"5.45\"  
-DPACKAGE_STRING=\"expect 5.45\" -DPACKAGE_BUGREPORT=\"\" -DBUILD_expect= -DSTDC_HEADERS=1 -DHAVE_SYS_TYPES_H=1 -DHAVE_SYS_STAT_H=1 -DHAVE_STDLIB_H=1 -DHAVE_STRING_H=1 -DHAVE_MEMORY_H=1 -DHAVE_STRINGS_H=1 -DHAVE_INTTYPES_H=1 -DHAVE_STDINT_H=1 -DHAVE_UNISTD_H=1 -DHAVE_LIMITS_H=1 -DHAVE_CONFIG_H=1 -I./include -I./lib -L./lib -ldl -lm -lc -o expect
```

Sitten ajetaan testit. Tämä vaihe ei ole pakollinen. Testin virheet tässä vaiheessa eivät ole yllättäviä, eikä niistä tarvitse huolestua.

```
make test
```

```
stty.test
all.tcl:          Total    26      Passed   15      Skipped   0      Failed   11
Sourced 0 Test Files.
Files with failing tests: logfile.test send.test spawn.test
lfs:/mnt/lfs/sources/expect5.45$
```

Seuraavaksi asennetaan.

```
make SCRIPTS="" install
```

```
lfs:/mnt/lfs/sources/expect5.45$ make SCRIPTS="" install
(echo 'if {[package vsatisfies [package provide Tcl] 8.6]} {return}' ; \
echo 'package ifneeded Expect 5.45 \
[list load [file join $dir libexpect5.45.so]]'\
) > pkgIndex.tcl
```

SCRIPTS komennon keskellä estää asentamasta täydentäviä scriptejä Expectiin, koska emme tarvitse niitä.

Lopuksi poistetaan sources-hakemistosta purettu lähdehakemisto.

```
cd ..  
rm -rf expect5.45
```

```
lfs:/mnt/lfs/sources/expect5.45$ cd ..  
lfs:/mnt/lfs/sources$ rm -rf expect5.45  
lfs:/mnt/lfs/sources$
```

2.8.10 DejaGNU-1.6

Arvioitu käännösaika: alle 0,1 SBU

Asennettava ohjelma: runtest

Ensimmäiseksi puretaan dejagnu-paketti ja siirrytään luotuun hakemistoon.

```
tar xf dejagnu-1.6.tar.gz
cd dejagnu-1.6
```

```
lfs:/mnt/lfs/sources$ tar xf dejagnu-1.6.tar.gz
lfs:/mnt/lfs/sources$ cd dejagnu-1.6
lfs:/mnt/lfs/sources/dejagnu-1.6$
```

Seuraavaksi ajetaan configure.

```
./configure --prefix=/tools
```

```
lfs:/mnt/lfs/sources/dejagnu-1.6$ ./configure --prefix=/tools
checking for a BSD-compatible install... /bin/ginstall -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
```

Configuren jälkeen asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/dejagnu-1.6$ make install
Done. Now run 'make install'.
make[1]: Entering directory `/mnt/lfs/sources/dejagnu-1.6'
/bin/mkdir -p '/tools/bin'
/bin/ginstall -c runtest '/tools/bin'
/bin/mkdir -p '/tools/share/dejagnu/libexec'
/bin/ginstall -c config.guess '/tools/share/dejagnu/libexec'
```

Ajetaan testit. Tämä ei edelleenkään ole pakollinen vaihe.

```
make check
```

```
=== Summary ===  
# of expected passes          78  
make[1]: Leaving directory `/mnt/lfs/sources/dejagnu-1.6'  
lfs:/mnt/lfs/sources/dejagnu-1.6$ █
```

Lopuksi poistetaan sources-hakemistosta purettu lähdehakemisto.

```
cd ..  
rm -rf dejagnu-1.6
```

```
lfs:/mnt/lfs/sources/dejagnu-1.6$ cd ..  
lfs:/mnt/lfs/sources$ rm -rf dejagnu-1.6  
lfs:/mnt/lfs/sources$ █
```

2.8.11 Check-0.11.0

Arvioitu käännösaika: 0,1 SBU

Asennettava ohjelma: checkmk

Asennettava kirjasto: libcheck.{a,so}

Ensimmäiseksi puretaan check-paketti ja siirrytään luotuun hakemistoon.

```
tar xf check-0.11.0.tar.gz
cd check-0.11.0
```

```
lfs:/mnt/lfs/sources$ tar xf check-0.11.0.tar.gz
lfs:/mnt/lfs/sources$ cd check-0.11.0
lfs:/mnt/lfs/sources/check-0.11.0$
```

Seuraavaksi ajetaan configure

```
PKG_CONFIG= ./configure --prefix=/tools
```

PKG_CONFIG= komennon alussa kertoo configure-skriptille olla huomioimatta yhtäkään PKG_CONFIG -asetusta, jotka voivat aiheuttaa sen, että järjestelmä yrittää linkata sellaisten hakemistojen kanssa, jotka eivät ole /tools -hakemistossa.

```
lfs:/mnt/lfs/sources/check-0.11.0$ PKG_CONFIG= ./configure --prefix=/tools
checking for a BSD-compatible install... /bin/ginstall -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
```

```
=====
Summary of Check 0.11.0 options:

fork mode ..... yes
high resolution timer replacement .... no
snprintf replacement ..... no
subunit support..... no
timeout unit tests ..... yes
=====
lfs:/mnt/lfs/sources/check-0.11.0$
```

Seuraavaksi ajetaan make.

```
make
```

```
lfs:/mnt/lfs/sources/check-0.11.0$ make
make all-recursive
make[1]: Entering directory `/mnt/lfs/sources/check-0.11.0'
Making all in lib
make[2]: Entering directory `/mnt/lfs/sources/check-0.11.0/lib'
```

Sitten testataan. Tässä kestää jonkin aikaa (n. 4 SBU). Tämä ei ole pakollinen vaihe.

```
make check
```

```
=====
Testsuite summary for Check 0.11.0
=====
```

```
# TOTAL: 8
# PASS: 8
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/check-0.11.0$ make install
Making install in lib
make[1]: Entering directory `/mnt/lfs/sources/check-0.11.0/lib'
make[2]: Entering directory `/mnt/lfs/sources/check-0.11.0/lib'
make[2]: Nothing to be done for `install-exec-am'.
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/mnt/lfs/sources/check-0.11.0/lib'
```

Lopuksi poistetaan sources-hakemistosta purettu lähdehakemisto.

```
cd ..
rm -rf check-0.11.0
```

```
lfs:/mnt/lfs/sources/check-0.11.0$ cd ..
lfs:/mnt/lfs/sources$ rm -rf check-0.11.0
lfs:/mnt/lfs/sources$
```

2.8.12 Ncurses-6.0

Arvioitu käännösaika: 0,5 SBU

Ensin puretaan ncurses-paketti ja siirrytään luotuun hakemistoon.

```
tar xf ncurses-6.0.tar.gz
cd ncurses-6.0
```

```
lfs:/mnt/lfs/sources$ tar xf ncurses-6.0.tar.gz
lfs:/mnt/lfs/sources$ cd ncurses-6.0
lfs:/mnt/lfs/sources/ncurses-6.0$
```

Seuraavaksi varmistetaan, että gawk löytyy configuren aikana.

```
sed -i s/mawk// configure
```

```
lfs:/mnt/lfs/sources/ncurses-6.0$ sed -i s/mawk// configure
lfs:/mnt/lfs/sources/ncurses-6.0$
```

Sitten ajetaan configure.

```
./configure --prefix=/tools \
            --with-shared \
            --without-debug \
            --without-ada \
            --enable-widec \
            --enable-overwrite
```

```
lfs:/mnt/lfs/sources/ncurses-6.0$ ./configure --prefix=/tools \
> --with-shared \
> --without-debug \
> --without-ada \
> --enable-widec \
> --enable-overwrite
checking for egrep... grep -E
Configuring NCURSES 6.0 ABI 6 (Sat Mar 18 16:48:29 EET 2017)
checking for package version... 6.0
checking for package patch date... 20150808
```

```
** Configuration summary for NCURSES 6.0 20150808:

    extended funcs: yes
    xterm terminfo: xterm-new

    bin directory: /tools/bin
    lib directory: /tools/lib
    include directory: /tools/include
    man directory: /tools/share/man
    terminfo directory: /tools/share/terminfo

lfs:/mnt/lfs/sources/ncurses-6.0$
```

Puretaan auki:

- `--without-ada` → Varmistaa, että Ncurses ei sisällytä tukea Ada-kääntäjälle, joka saattaa olla isäntäkoneella, mutta ei ole saatavilla enää sen jälkeen, kun menemme [chroot](#)-ympäristöön.
- `--enable-overwrite` → Käskee Ncurses:n asentaa sen header-tiedostot `/tools/include` -hakemistoon `/tools/include/ncurses` -hakemiston sijaan. Tämä varmistaa, että muut paketit löytävät Ncurses:n header-tiedostot.
- `--enable-widec` → Aiheuttaa sen, että wide-character -kirjastot (esim. `libncursesw.so.6.0`) asennetaan normaalien (esim. `libncurses.so.6.0`) sijaan. Nämä wide-character -kirjastot ovat käytettävissä sekä multibittisissä että perinteisissä 8-bittisissä lokaaleissa, kun normaalit kirjastot toimivat ainoastaan 8-bittisissä lokaaleissa.

Seuraavaksi kootaan paketti.

```
make
```

Asennetaan.

```
make install
```

Ncurses:lle on olemassa testit, jotka voi halutessaan ajaa asennuksen jälkeen. Testit sijaitsevat `test/` -hakemistossa. Samassa hakemistossa on `README`-tiedosto, josta saa tarkempia ohjeita testien tekemiseen.

Lopuksi poistetaan sources-hakemistosta purettu lähdehakemisto.

```
cd ..  
rm -rf ncurses-6.0
```

```
lfs:/mnt/lfs/sources/ncurses-6.0$ cd ..  
lfs:/mnt/lfs/sources$ rm -rf ncurses-6.0  
lfs:/mnt/lfs/sources$
```

2.8.13 Bash-4.4

Arvioitu käännösaika: 0,4 SBU

Ensin puretaan bash-paketti ja siirrytään luotuun hakemistoon.

```
tar xf bash-4.4.tar.gz
cd bash-4.4
```

```
lfs:/mnt/lfs/sources$ tar xf bash-4.4.tar.gz
lfs:/mnt/lfs/sources$ cd bash-4.4
lfs:/mnt/lfs/sources/bash-4.4$
```

Seuraavaksi ajetaan configure.

```
./configure --prefix=/tools --without-bash-malloc
```

Puretaan auki:

- `--without-bash-malloc` → Poistaa käytöstä Bash:n memory allocation (malloc)-toiminnon, jonka tiedetään aiheuttavan segmentaatiovirheitä. Sen sijaan Bash käyttää Glibc:n malloc-toimintoja, jotka ovat paljon vakaampia.

```
lfs:/mnt/lfs/sources/bash-4.4$ ./configure --prefix=/tools --without-bash-malloc
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu

Beginning configuration for bash-4.4-release for x86_64-unknown-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
```

Seuraavaksi ajetaan make.

```
make
```

```
lfs:/mnt/lfs/sources/bash-4.4$ make
rm -f mksyntax
gcc -DPROGRAM="bash" -DCONF_HOSTTYPE="x86_64" -DCONF_OSTYPE="linux-gnu" -DCONF_MACHTYPE="x86_64-unknown-linux-gnu" -DCONF_VENDOR="unknown" -DLOCALEDIR="/tools/share/locale" -DPACKAGE="bash" -DSHELL -DHAVE_CONFIG_H -I. -I. -I./include -I./lib -g -O2 -Wno-parentheses -Wno-format-security -L /lib/termcap -rdynamic -g -O2 -
```

Sitten ajetaan testit. Tämä vaihe ei ole pakollinen.

```
make tests
```

```
lfs:/mnt/lfs/sources/bash-4.4$ make tests

*****
*
* GNU bash, version 4.4.12(1)-release (x86_64-unknown-linux-gnu)
*
*****

Testing /mnt/lfs/sources/bash-4.4/bash
version: 4.4.12(1)-release
versinfo: 4 4 12 1 release x86_64-unknown-linux-gnu
```

Asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/bash-4.4$ make install

*****
*
* GNU bash, version 4.4.12(1)-release (x86_64-unknown-linux-gnu)
*
*****

mkdir -p -- /tools/share/doc/bash
( cd ../po/ ; make DESTDIR= installdirs )
```

Lopuksi luodaan vielä linkki sellaisia ohjelmia varten, jotka käyttävät sh:ta shellinä.

```
ln -sv bash /tools/bin/sh
```

```
lfs:/mnt/lfs/sources/bash-4.4$ ln -sv bash /tools/bin/sh
'/tools/bin/sh' -> 'bash'
lfs:/mnt/lfs/sources/bash-4.4$ █
```

Lopuksi poistetaan vielä purettu hakemisto sources-hakemistosta.

```
cd ..
rm -rf bash-4.4
```

```
lfs:/mnt/lfs/sources/bash-4.4$ cd ..
lfs:/mnt/lfs/sources$ rm -rf bash-4.4
lfs:/mnt/lfs/sources$ █
```

2.8.14 Bison-3.0.4

Arvioitu käännösaika: 0,3 SBU

Ensin puretaan Bison-paketti ja siirrytään luotuun hakemistoon.

```
tar xf bison-3.0.4.tar.xz
cd bison-3.0.4
```

```
lfs:/mnt/lfs/sources$ tar xf bison-3.0.4.tar.xz
lfs:/mnt/lfs/sources$ cd bison-3.0.4
lfs:/mnt/lfs/sources/bison-3.0.4$
```

Ajetaan configure.

```
./configure --prefix=/tools
```

```
lfs:/mnt/lfs/sources/bison-3.0.4$ ./configure --prefix=/tools
checking for a BSD-compatible install... /bin/ginstall -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether make supports nested variables... (cached) yes
```

Käännetään paketti.

```
make
```

```
lfs:/mnt/lfs/sources/bison-3.0.4$ make
GEN      examples/extracted.stamp
extexi:  GEN examples/rpcalc/rpcalc.y
extexi:  GEN examples/mfcalc/mfcalc.y
extexi:  GEN examples/mfcalc/calc.h
extexi:  GEN examples/calc++/calc++-driver.hh
```

Ajetaan testit. Tämä kestää jonkin aikaa, eikä ole pakollinen vaihe.

```
make check
```

```
## ----- ##
## Test results. ##
## ----- ##

486 tests were successful.
7 tests were skipped.
make[3]: Leaving directory `/mnt/lfs/sources/bison-3.0.4'
make[2]: Leaving directory `/mnt/lfs/sources/bison-3.0.4'
make[1]: Leaving directory `/mnt/lfs/sources/bison-3.0.4'
lfs:/mnt/lfs/sources/bison-3.0.4$
```

Asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/bison-3.0.4$ make install
make install-recursive
make[1]: Entering directory `/mnt/lfs/sources/bison-3.0.4'
Making install in po
make[2]: Entering directory `/mnt/lfs/sources/bison-3.0.4/po'
installing da.gmo as /tools/share/locale/da/LC_MESSAGES/bison.mo
installing de.gmo as /tools/share/locale/de/LC_MESSAGES/bison.mo
```

Lopuksi poistetaan sources-hakemistosta purettu lähdehakemisto.

```
cd ..
rm -rf bison-3.0.4
```

```
lfs:/mnt/lfs/sources/bison-3.0.4$ cd ..
lfs:/mnt/lfs/sources$ rm -rf bison-3.0.4
lfs:/mnt/lfs/sources$
```

2.8.15 Bzip2-1.0.6

Arvioitu käännösaika: alle 0,1 SBU

Ensin puretaan bzip-paketti.

```
tar xf bzip2-1.0.6.tar.gz
```

```
lfs:/mnt/lfs/sources$ tar xf bzip2-1.0.6.tar.gz
lfs:/mnt/lfs/sources$
```

Siirrytään bzip-hakemistoon. Koska bzip-paketti ei sisällä configure-skriptiä, ajetaan make.

```
cd bzip2-1.0.6
make
```

```
lfs:/mnt/lfs/sources$ cd bzip2-1.0.6
lfs:/mnt/lfs/sources/bzip2-1.0.6$ make

If compilation produces errors, or a large number of warnings,
please read README.COMPIRATION.PROBLEMS -- you might be able to
adjust the flags in this Makefile to improve matters.
```

Asennetaan.

```
make PREFIX=/tools install
```

```
lfs:/mnt/lfs/sources/bzip2-1.0.6$ make PREFIX=/tools install
if ( test ! -d /tools/bin ) ; then mkdir -p /tools/bin ; fi
if ( test ! -d /tools/lib ) ; then mkdir -p /tools/lib ; fi
if ( test ! -d /tools/man ) ; then mkdir -p /tools/man ; fi
if ( test ! -d /tools/man/man1 ) ; then mkdir -p /tools/man/man1 ; fi
```

Lopuksi poistetaan purettu lähdehakemisto sources-hakemistosta.

```
cd ..  
rm -rf bzip2-1.0.6
```

```
lfs:/mnt/lfs/sources/bzip2-1.0.6$ cd ..  
lfs:/mnt/lfs/sources$ rm -rf bzip2-1.0.6  
lfs:/mnt/lfs/sources$
```

2.8.16 Coreutils-8.26

Arvioitu käännösaika: 0,6 SBU

Ensin puretaan coreutils-paketti.

```
tar xf coreutils-8.26.tar.xz
```

```
lfs:/mnt/lfs/sources$ tar xf coreutils-8.26.tar.xz
lfs:/mnt/lfs/sources$
```

Ajetaan configure.

```
./configure --prefix=/tools --enable-install-program=hostname
```

Puretaan auki:

- `--enable-install-program=hostname` → Asentaa `hostname`-binääriin, jonka Perl:n testit vaativat.

```
lfs:/mnt/lfs/sources/coreutils-8.26$ ./configure --prefix=/tools --enable-install-program=hostname
checking for a BSD-compatible install... /bin/ginstall -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir... /bin/mkdir -p
```

Seuraavaksi ajetaan make.

```
make
```

On mahdollista, että tulee seuraava varoitus:

```
WARNING: 'aclocal-1.15' is missing on your system.
```

Ongelman voi korjata muutamalla komennolla:

```
touch aclocal.m4 configure
touch Makefile.am
touch Makefile.in
```

Ongelman voi myös kiertää asentamalla automake-paketin. Tämän jälkeen aja make uudelleen.

Seuraavaksi ajetaan testit. Tämä vaihe ei ole pakollinen.

```
make RUN_EXPENSIVE_TESTS=yes check
```

Puretaan auki:

- `RUN_EXPENSIVE_TESTS=yes` → Tämä parametri käskee testikokoelman ajaa useita lisätestejä, jotka luokitellaan joillakin alustoilla melko kustannussyöpöiksi CPU:ta ja muistin käyttöä ajatellen, mutta jotka eivät kuitenkaan ole tavallisesti ongelma Linuxissa.

```
=====
testsuite summary for GNU coreutils 8.26
=====
TOTAL: 311
PASS: 277
SKIP: 34
XFAIL: 0
FAIL: 0
XPASS: 0
ERROR: 0
=====
```

Asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/coreutils-8.26$ make install
make install-recursive
make[1]: Entering directory `/mnt/lfs/sources/coreutils-8.26'
Making install in po
make[2]: Entering directory `/mnt/lfs/sources/coreutils-8.26/po'
```

Lopuksi poistetaan purettu lähdehakemisto sources-hakemistosta.

```
cd ..
rm -rf coreutils-8.26
```

```
lfs:/mnt/lfs/sources/coreutils-8.26$ cd ..
lfs:/mnt/lfs/sources$ rm -rf coreutils-8.26
lfs:/mnt/lfs/sources$
```

2.8.17 2.8.16 Diffutils-3.5

Arvioitu käännösaika: 0,2 SBU

Ensimmäiseksi puretaan diffutils-paketti.

```
tar xf diffutils-3.5.tar.xz
```

```
lfs:/mnt/lfs/sources$ tar xf diffutils-3.5.tar.xz
lfs:/mnt/lfs/sources$
```

Sitten siirytään purettuun diffutils-hakemistoon ja ajetaan configure.

```
cd diffutils-3.5
./configure --prefix=/tools
```

```
lfs:/mnt/lfs/sources$ cd diffutils-3.5
lfs:/mnt/lfs/sources/diffutils-3.5$ ./configure --prefix=/tools
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread safe mkdir... /tools/bin/mkdir -p
```

Seuraavaksi ajetaan make.

```
make
```

```
lfs:/mnt/lfs/sources/diffutils-3.5$ make
Making all in lib
make[1]: Entering directory `/mnt/lfs/sources/diffutils-3.5/lib'
Makefile:1777: target `.deps/alloca.Po' given more than once in the same rule.
GEN      alloca.h
GEN      configmake.h
GEN      c++defs.h
GEN      warn-on-use.h
```

Seuraavaksi ajetaan testit. Tämä vaihe ei ole pakollinen.

```
make check
```

```
=====
Testsuite summary for GNU diffutils 3.5
=====
```

```
# TOTAL: 167
# PASS: 141
# SKIP: 26
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/diffutils-3.5$ make install
Making install in lib
make[1]: Entering directory `/mnt/lfs/sources/diffutils-3.5/lib'
Makefile:1777: target `.deps/alloca.Po' given more than once in the same rule.
make install-am
make[2]: Entering directory `/mnt/lfs/sources/diffutils-3.5/lib'
Makefile:1777: target `.deps/alloca.Po' given more than once in the same rule
```

Lopuksi poistetaan purettu hakemisto sources-hakemistosta.

```
cd ..
rm -rf diffutils-3.5
```

```
lfs:/mnt/lfs/sources/diffutils-3.5$ cd ..
lfs:/mnt/lfs/sources$ rm -rf diffutils-3.5
lfs:/mnt/lfs/sources$ █
```

2.8.18 File-5.30

Arvioitu käännösaika: 0,1 SBU

Ensimmäiseksi puretaan file-paketti.

```
tar xf file-5.30.tar.gz
```

```
lfs:/mnt/lfs/sources$ tar xf file-5.30.tar.gz
lfs:/mnt/lfs/sources$
```

Seuraavaksi siirytään purettuun hakemistoon ja ajetaan configure.

```
cd file-5.30
./configure --prefix=/tools
```

```
lfs:/mnt/lfs/sources$ cd file-5.30
lfs:/mnt/lfs/sources/file-5.30$ ./configure --prefix=/tools
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
checking for gawk... gawk
```

Ajetaan make.

```
make
```

```
lfs:/mnt/lfs/sources/file-5.30$ make
make all-recursive
make[1]: Entering directory `/mnt/lfs/sources/file-5.30'
Making all in src
make[2]: Entering directory `/mnt/lfs/sources/file-5.30/src'
sed -e "s/X YY/$(echo 5.30 | tr -d ' ')/" < /src/magic.h.in > magic.h
```

Sitten testit. Tämä vaihe ei ole pakollinen.

```
make check
```

```
lfs:/mnt/lfs/sources/file-5.30$ make check
Making check in src
make[1]: Entering directory `/mnt/lfs/sources/file-5.30/src'
make check-am
make[2]: Entering directory `/mnt/lfs/sources/file-5.30/src'
```

Asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/file-5.30$ make install
Making install in src
make[1]: Entering directory `/mnt/lfs/sources/file-5.30/src'
make install-am
make[2]: Entering directory `/mnt/lfs/sources/file-5.30/src'
make[3]: Entering directory `/mnt/lfs/sources/file-5.30/src'
```

Lopuksi poistetaan sources-hakemistosta purettu lähdehakemisto.

```
cd ..
rm -rf file-5.30
```

```
lfs:/mnt/lfs/sources/file-5.30$ cd ..
lfs:/mnt/lfs/sources$ rm -rf file-5.30
lfs:/mnt/lfs/sources$
```

2.8.19 Findutils-4.6.0

Arvioitu käännösaika: 0,3 SBU

Ensin puretaan Findutils-paketti ja siirrytään luotuun hakemistoon.

```
tar xf findutils-4.6.0.tar.gz
cd findutils-4.6.0
```

```
lfs:/mnt/lfs/sources$ tar xf findutils-4.6.0.tar.gz
lfs:/mnt/lfs/sources$ cd findutils-4.6.0
lfs:/mnt/lfs/sources/findutils-4.6.0$
```

Sitten valmistellaan paketti kääntämistä varten.

```
./configure --prefix=/tools
```

Käännetään paketti.

```
make
```

Ajetaan testit. Tämä vaihe ei ole pakollinen.

```
make check
```

Asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/findutils-4.6.0$ make install
Making install in gl
make[1]: Entering directory `/mnt/lfs/sources/findutils-4.6.0/gl'
Making install in lib
make[2]: Entering directory `/mnt/lfs/sources/findutils-4.6.0/gl/lib'
```

Lopuksi poistetaan purettu lähdehakemisto sources-hakemistosta.

```
cd ..  
rm -rf findutils-4.6.0
```

```
lfs:/mnt/lfs/sources/findutils-4.6.0$ cd ..  
lfs:/mnt/lfs/sources$ rm -rf findutils-4.6.0  
lfs:/mnt/lfs/sources$ █
```

2.8.20 Gawk-4.1.4

Arvioitu käännösaika: 0,2 SBU

Ensin puretaan Gawk-paketti ja siirrytään purettuun hakemistoon.

```
tar xf gawk-4.1.4.tar.xz
cd gawk-4.1.4
```

```
lfs:/mnt/lfs/sources$ tar xf gawk-4.1.4.tar.xz
lfs:/mnt/lfs/sources$ cd gawk-4.1.4
lfs:/mnt/lfs/sources/gawk-4.1.4$
```

Valmistellaan paketti kääntämistä varten.

```
./configure --prefix=/tools
```

```
lfs:/mnt/lfs/sources/gawk-4.1.4$ ./configure --prefix=/tools
checking for a BSD-compatible install... ./install-sh -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
```

Käännetään paketti.

```
make
```

```
lfs:/mnt/lfs/sources/gawk-4.1.4$ make
make 'CFLAGS=-g -O2 -DNDEBUG' 'LDFLAGS=-Wl,-export-dynamic' all-recursive
make[1]: Entering directory `/mnt/lfs/sources/gawk-4.1.4'
Making all in .
make[2]: Entering directory `/mnt/lfs/sources/gawk-4.1.4'
gcc -DDEFPATH='\"./tools/share/awk\"' -DDEFLIBPATH='\"/tools/lib/gawk\"' -DSHLIBEXT='\"'
```

Ajetaan testit. Tämä vaihe ei ole pakollinen.

```
make check
```

```
lfs:/mnt/lfs/sources/gawk-4.1.4$ make check
Making check in .
make[1]: Entering directory `/mnt/lfs/sources/gawk-4.1.4'
make 'CFLAGS=-g -O2 -DNDEBUG' 'LDFLAGS=-Wl,-export-dynamic' check-local
make[2]: Entering directory `/mnt/lfs/sources/gawk-4.1.4'
```

Asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/gawk-4.1.4$ make install
Making install in .
make[1]: Entering directory `/mnt/lfs/sources/gawk-4.1.4'
make[2]: Entering directory `/mnt/lfs/sources/gawk-4.1.4'
  /tools/bin/mkdir -p '/tools/bin'
```

Lopuksi poistetaan purettu lähdehakemisto sources-hakemistosta.

```
cd ..
rm -rf gawk-4.1.4
```

```
lfs:/mnt/lfs/sources/gawk-4.1.4$ cd ..
lfs:/mnt/lfs/sources$ rm -rf gawk-4.1.4
lfs:/mnt/lfs/sources$ █
```

2.8.21 Gettext-0.19.8.1

Arvioitu käännösaika: 0,9 SBU

Tilapäistä työkalusettiämme varten meidän tarvitsee kääntää ja asentaa ainoastaan kolme ohjelmaa Gettext-paketista.

Ensin puretaan Gettext-paketti ja siirrytään purettuun hakemistoon.

```
tar xf gettext-0.19.8.1.tar.xz
cd gettext-0.19.8.1
```

```
lfs:/mnt/lfs/sources$ tar xf gettext-0.19.8.1.tar.xz
lfs:/mnt/lfs/sources$ cd gettext-0.19.8.1
lfs:/mnt/lfs/sources/gettext-0.19.8.1$
```

Valmistellaan Gettext kääntämistä varten.

```
cd gettext-tools
EMACS="no" ./configure --prefix=/tools --disable-shared
```

Puretaan auki:

- `EMACS="no"` → Estää configure-skriptiä määrittämästä, mihin Emacs Lisp -tiedostot asennetaan.
- `--disable-shared` → Meidän ei tarvitse asentaa yhtään Gettextin jaeuista kirjastoista tällä kertaa, joten ei ole mitään tarvetta kääntää niitä.

Käännetään paketti seuraavilla komennoilla:

```
make -C gnulib-lib
make -C intl pluralx.c
make -C src msgfmt
make -C src msgmerge
make -C src xgettext
```

```
lfs:/mnt/lfs/sources/gettext-0.19.8.1/gettext-tools$ make -C intl pluralx.c
make: Entering directory `/mnt/lfs/sources/gettext-0.19.8.1/gettext-tools/intl'
sed -e 's|^#line \([0-9]*\) "\(\plural\.[cy]\)"|#line \1 "../../gettext-runtime/intl/\2"|' < ../../gettext-runtime/intl/plural.c > pluralx.c
make: Leaving directory `/mnt/lfs/sources/gettext-0.19.8.1/gettext-tools/intl'
lfs:/mnt/lfs/sources/gettext-0.19.8.1/gettext-tools$
```

```
lfs:/mnt/lfs/sources/gettext-0.19.8.1/gettext-tools$ make -C src msgmerge
make: Entering directory `/mnt/lfs/sources/gettext-0.19.8.1/gettext-tools/src'
gcc -DLOCALEDIR="/tools/share/locale/" -DBISON_LOCALEDIR="/tools/share/locale/" -DLOCALE_ALIAS_PATH="/tools/share/locale/" -DUSEJAVA=1 -DUSEJEXE=1 -DGETTEXTJEXEDIR="/tools/lib/gettext/" -DGETTEXTJAR="/tools/share/gettext/gettext.jar/" -DLIBDIR="/tools/lib/"
```

```
lfs:/mnt/lfs/sources/gettext-0.19.8.1/gettext-tools$ make -C src xgettext
make: Entering directory `/mnt/lfs/sources/gettext-0.19.8.1/gettext-tools/src'
gcc -DLOCALEDIR="/tools/share/locale/" -DBISON_LOCALEDIR="/tools/share/locale/" -DLOCALE_ALIAS_PATH="/tools/share/locale/" -DUSEJAVA=1 -DUSEJEXE=1 -DGETTEXTJEXEDIR="/tools/lib/gettext/" -DGETTEXTJAR="/tools/share/gettext/gettext.jar/" -DLIBDIR="/tools/lib/" -DGETTEXTDATADIR="/tools/share/gettext/" -DPROJECTSDIR="/tools/share/gettext/p
```

Testejä ei voida ajaa, koska vain kolme ohjelmaa on käännetty.

Asennetaan seuraavat ohjelmat: msgfmt, msgmerge ja xgettext.

```
cp -v src/{msgfmt,msgmerge,xgettext} /tools/bin
```

```
lfs:/mnt/lfs/sources/gettext-0.19.8.1/gettext-tools$ cp -v src/{msgfmt,msgmerge,xgettext} /tools/bin
'src/msgfmt' -> '/tools/bin/msgfmt'
'src/msgmerge' -> '/tools/bin/msgmerge'
'src/xgettext' -> '/tools/bin/xgettext'
lfs:/mnt/lfs/sources/gettext-0.19.8.1/gettext-tools$
```

Lopuksi poistetaan purettu lähdehakemisto sources-hakemistosta.

```
cd $LFS/sources
rm -rf gettext-0.19.8.1
```

```
lfs:/mnt/lfs/sources/gettext-0.19.8.1/gettext-tools$ cd $LFS/sources
lfs:/mnt/lfs/sources$ rm -rf gettext-0.19.8.1
lfs:/mnt/lfs/sources$
```

2.8.22 Grep-3.0

Arvioitu käännösaika: 0,2 SBU

Ensin puretaan grep-paketti ja siirrytään purettuun hakemistoon.

```
tar xf grep-3.0.tar.xz
cd grep-3.0
```

```
lfs:/mnt/lfs/sources$ tar xf grep-3.0.tar.xz
lfs:/mnt/lfs/sources$ cd grep-3.0
lfs:/mnt/lfs/sources/grep-3.0$
```

Ajetaan configure.

```
./configure --prefix=/tools
```

```
lfs:/mnt/lfs/sources/grep-3.0$ ./configure --prefix=/tools
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
```

Käännetään paketti.

```
make
```

```
lfs:/mnt/lfs/sources/grep-3.0$ make
make all-recursive
make[1]: Entering directory `/mnt/lfs/sources/grep-3.0'
Making all in po
make[2]: Entering directory `/mnt/lfs/sources/grep-3.0/po'
make[2]: Nothing to be done for `all'.
make[2]: Leaving directory `/mnt/lfs/sources/grep-3.0/po'
```

Ajetaan testit. Tämä ei ole pakollinen vaihe.

```
make check
```

```
=====
Testsuite summary for GNU grep 3.0
=====
```

```
# TOTAL: 143
# PASS: 127
# SKIP: 16
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/grep-3.0$ make install
Making install in po
make[1]: Entering directory `/mnt/lfs/sources/grep-3.0/po'
installing af.gmo as /tools/share/locale/af/LC_MESSAGES/grep.mo
installing be.gmo as /tools/share/locale/be/LC_MESSAGES/grep.mo
installing bg.gmo as /tools/share/locale/bg/LC_MESSAGES/grep.mo
```

Lopuksi poistetaan purettu hakemisto sources-hakemistosta.

```
cd ..
```

```
rm -rf grep-3.0
```

```
lfs:/mnt/lfs/sources/grep-3.0$ cd ..
lfs:/mnt/lfs/sources$ rm -rf grep-3.0
lfs:/mnt/lfs/sources$
```

2.8.23 Gzip-1.8

Arvioitu käännösaika: 0,1 SBU

Ensimmäiseksi puretaan Gzip-paketti ja siirrytään purettuun hakemistoon.

```
tar xf gzip-1.8.tar.xz
cd gzip-1.8
```

```
lfs:/mnt/lfs/sources$ tar xf gzip-1.8.tar.xz
lfs:/mnt/lfs/sources$ cd gzip-1.8
lfs:/mnt/lfs/sources/gzip-1.8$
```

Valmistellaan paketti kääntämistä varten.

```
./configure --prefix=/tools
```

```
lfs:/mnt/lfs/sources/gzip-1.8$ ./configure --prefix=/tools
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
```

Käännetään paketti.

```
make
```

```
lfs:/mnt/lfs/sources/gzip-1.8$ make
GEN      version.c
GEN      version.h
make all-recursive
make[1]: Entering directory `/mnt/lfs/sources/gzip-1.8'
Making all in lib
make[2]: Entering directory `/mnt/lfs/sources/gzip-1.8/lib'
```

Ajetaan testit. Tämä vaihe ei ole pakollinen.

```
make check
```

```
=====
Testsuite summary for gzip 1.8
=====
# TOTAL: 18
# PASS: 18
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/gzip-1.8$ make install
make install-recursive
make[1]: Entering directory `/mnt/lfs/sources/gzip-1.8'
Making install in lib
make[2]: Entering directory `/mnt/lfs/sources/gzip-1.8/lib'
make install-am
make[3]: Entering directory `/mnt/lfs/sources/gzip-1.8/lib'
make[4]: Entering directory `/mnt/lfs/sources/gzip-1.8/lib'
```

Lopuksi poistetaan purettu lähdehakemisto sources-hakemistosta.

```
cd ..
rm -rf gzip-1.8
```

```
lfs:/mnt/lfs/sources/gzip-1.8$ cd ..
lfs:/mnt/lfs/sources$ rm -rf gzip-1.8
lfs:/mnt/lfs/sources$ █
```

2.8.24 M4-1.4.18

Arvioitu käännösaika: 0,2 SBU

Ensin puretaan M4-paketti ja siirrytään purettuun hakemistoon.

```
tar xf m4-1.4.18.tar.xz
cd m4-1.4.18
```

```
lfs:/mnt/lfs/sources$ tar xf m4-1.4.18.tar.xz
lfs:/mnt/lfs/sources$ cd m4-1.4.18
lfs:/mnt/lfs/sources/m4-1.4.18$
```

Valmistellaan paketti kääntämistä varten.

```
./configure --prefix=/tools
```

```
lfs:/mnt/lfs/sources/m4-1.4.18$ ./configure --prefix=/tools
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
checking for gawk... gawk
```

Käännetään paketti.

```
make
```

```
lfs:/mnt/lfs/sources/m4-1.4.18$ make
make all-recursive
make[1]: Entering directory `/mnt/lfs/sources/m4-1.4.18'
Making all in .
make[2]: Entering directory `/mnt/lfs/sources/m4-1.4.18'
make[2]: Nothing to be done for `all-am'.
make[2]: Leaving directory `/mnt/lfs/sources/m4-1.4.18'
```

Ajetaan testit. Tämä vaihe ei ole pakollinen.

```
make check
```

```
=====
Testsuite summary for GNU M4 1.4.18
=====
# TOTAL: 170
# PASS: 151
# SKIP: 19
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/m4-1.4.18$ make install
make install-recursive
make[1]: Entering directory `/mnt/lfs/sources/m4-1.4.18'
Making install in .
make[2]: Entering directory `/mnt/lfs/sources/m4-1.4.18'
make[3]: Entering directory `/mnt/lfs/sources/m4-1.4.18'
```

Lopuksi poistetaan purettu lähdehakemisto sources-hakemistosta.

```
cd ..
rm -rf m4-1.4.18
```

```
lfs:/mnt/lfs/sources/m4-1.4.18$ cd ..
lfs:/mnt/lfs/sources$ rm -rf m4-1.4.18
lfs:/mnt/lfs/sources$
```

2.8.25 Make-4.2.1

Arvioitu käännösaika: 0,1 SBU

Ensimmäiseksi puretaan make-paketti ja siirrytään purettuun hakemistoon.

```
tar xf make-4.2.1.tar.bz2
cd make-4.2.1
```

```
lfs:/mnt/lfs/sources$ tar xf make-4.2.1.tar.bz2
lfs:/mnt/lfs/sources$ cd make-4.2.1
lfs:/mnt/lfs/sources/make-4.2.1$
```

Valmistellaan paketti kääntämistä varten.

```
./configure --prefix=/tools --without-guile
```

Puretaan auki:

- `--without-guile` → Tämä asetus varmistaa, että Make ei linkkaa Guilen kirjastoja vastaan, jotka saattavat olla läsnä isäntäjärjestelmässä, mutta eivät ole saatavilla chroot-ympäristössä seuraavassa kappaleessa.

```
lfs:/mnt/lfs/sources/make-4.2.1$ ./configure --prefix=/tools --without-guile
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
checking for gawk... gawk
```

Käännetään paketti.

```
make
```

```
lfs:/mnt/lfs/sources/make-4.2.1$ make
make all-recursive
make[1]: Entering directory `/mnt/lfs/sources/make-4.2.1'
Making all in glob
make[2]: Entering directory `/mnt/lfs/sources/make-4.2.1/glob'
```

Ajetaan testit. Tämä vaihe ei ole pakollinen.

```
make check
```

```
585 Tests in 120 Categories Complete ... No Failures :-)
```

Asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/make-4.2.1$ make install
Making install in glob
make[1]: Entering directory `/mnt/lfs/sources/make-4.2.1/glob'
make[2]: Entering directory `/mnt/lfs/sources/make-4.2.1/glob'
make[2]: Nothing to be done for `install-exec-am'.
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/mnt/lfs/sources/make-4.2.1/glob'
```

Lopuksi poistetaan purettu lähdehakemisto sources-hakemistosta.

```
cd ..
rm -rf make-4.2.1
```

```
lfs:/mnt/lfs/sources/make-4.2.1$ cd ..
lfs:/mnt/lfs/sources$ rm -rf make-4.2.1
lfs:/mnt/lfs/sources$
```

2.8.26 Patch-2.7.5

Arvioitu käännösaika: 0,2 SBU

Ensin puretaan patch-paketti ja siirrytään purettuun hakemistoon.

```
tar xf patch-2.7.5.tar.xz
cd patch-2.7.5
```

```
lfs:/mnt/lfs/sources$ tar xf patch-2.7.5.tar.xz
lfs:/mnt/lfs/sources$ cd patch-2.7.5
lfs:/mnt/lfs/sources/patch-2.7.5$
```

Valmistellaan paketti kääntämistä varten.

```
./configure --prefix=/tools
```

```
lfs:/mnt/lfs/sources/patch-2.7.5$ ./configure --prefix=/tools
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
checking for gawk... gawk
```

Käännetään.

```
make
```

```
lfs:/mnt/lfs/sources/patch-2.7.5$ make
make all-recursive
make[1]: Entering directory '/mnt/lfs/sources/patch-2.7.5'
Making all in lib
make[2]: Entering directory '/mnt/lfs/sources/patch-2.7.5/lib'
GEN      alloca.h
GEN      configmake.h
```

Ajetaan testit. Tämä vaihe ei ole pakollinen.

```
make check
```

```
=====
Testsuite summary for GNU patch 2.7.5
=====
# TOTAL: 39
# PASS: 38
# SKIP: 0
# XFAIL: 1
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/patch-2.7.5$ make install
make install-recursive
make[1]: Entering directory '/mnt/lfs/sources/patch-2.7.5'
Making install in lib
make[2]: Entering directory '/mnt/lfs/sources/patch-2.7.5/lib'
make install-am
```

Lopuksi poistetaan purettu lähdehakemisto sources-hakemistosta.

```
cd ..
rm -rf patch-2.7.5
```

```
lfs:/mnt/lfs/sources/patch-2.7.5$ cd ..
lfs:/mnt/lfs/sources$ rm -rf patch-2.7.5
lfs:/mnt/lfs/sources$ █
```

2.8.27 Perl-5.24.1

Arvioitu käännösaika: 1,3 SBU

Ensin puretaan Perl ja siirrytään purettuun hakemistoon.

```
tar xf perl-5.24.1.tar.bz2
cd perl-5.24.1
```

```
lfs:/mnt/lfs/sources$ tar xf perl-5.24.1.tar.bz2
lfs:/mnt/lfs/sources$ cd perl-5.24.1
lfs:/mnt/lfs/sources/perl-5.24.1$
```

Valmistellaan paketti kääntämistä varten.

```
sh Configure -des -Dprefix=/tools -Dlibs=-lm
```

```
lfs:/mnt/lfs/sources/perl-5.24.1$ sh Configure -des -Dprefix=/tools -Dlibs=-lm
First let's make sure your kit is complete.  Checking...
Locating common programs...
Checking compatibility between /tools/bin/echo and builtin echo (if any)...
Symbolic links are supported.
```

Käännetään paketti.

```
make
```

Vain muutama työkalu ja kirjasto asennetaan tässä kappaleessa. Sen takia testejäkään ei voi vielä ajaa.

```
cp -v perl cpan/podlators/scripts/pod2man /tools/bin
mkdir -pv /tools/lib/perl5/5.24.1
cp -Rv lib/* /tools/lib/perl5/5.24.1
```

Lopuksi poistetaan purettu hakemisto.

```
cd ..  
rm -rf perl-5.24.1
```

```
lfs:/mnt/lfs/sources/perl-5.24.1$ cd ..  
lfs:/mnt/lfs/sources$ rm -rf perl-5.24.1  
lfs:/mnt/lfs/sources$ █
```

2.8.28 Sed-4.4

Arvioitu käännösaika: 0,1 SBU

Puretaan sed-paketti ja siirrytään purettuun hakemistoon.

```
tar xf sed-4.4.tar.xz
cd sed-4.4
```

```
lfs:/mnt/lfs/sources$ tar xf sed-4.4.tar.xz
lfs:/mnt/lfs/sources$ cd sed-4.4
lfs:/mnt/lfs/sources/sed-4.4$
```

Ajetaan configure.

```
./configure --prefix=/tools
```

```
lfs:/mnt/lfs/sources/sed-4.4$ ./configure --prefix=/tools
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
```

Ajetaan make.

```
make
```

```
lfs:/mnt/lfs/sources/sed-4.4$ make
GEN      lib/alloca.h
GEN      lib/configmake.h
GEN      lib/c++defs.h
GEN      lib/warn-on-use.h
```

Ajetaan testit. Tämä vaihe ei ole pakollinen.

```
make check
```

```

=====
Testsuite summary for GNU sed 4.4
=====
# TOTAL: 130
# PASS: 103
# SKIP: 27
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====

```

Asennetaan.

```
make install
```

```

lfs:/mnt/lfs/sources/sed-4.4$ make install
make install-recursive
make[1]: Entering directory '/mnt/lfs/sources/sed-4.4'
Making install in po
make[2]: Entering directory '/mnt/lfs/sources/sed-4.4/po'
installing af.gmo as /tools/share/locale/af/LC_MESSAGES/sed.mo
installing ast.gmo as /tools/share/locale/ast/LC_MESSAGES/sed.mo

```

Lopuksi poistetaan purettu hakemisto.

```
cd ..
rm -rf sed-4.4
```

```

lfs:/mnt/lfs/sources/sed-4.4$ cd ..
lfs:/mnt/lfs/sources$ rm -rf sed-4.4
lfs:/mnt/lfs/sources$

```

2.8.29 Tar-1.29

Arvioitu käännösaika: 0,3 SBU

Ensin puretaan tar-paketti ja siirrytään purettuun hakemistoon.

```
tar xf tar-1.29.tar.xz
cd tar-1.29
```

```
lfs:/mnt/lfs/sources$ tar xf tar-1.29.tar.xz
lfs:/mnt/lfs/sources$ cd tar-1.29
lfs:/mnt/lfs/sources/tar-1.29$
```

Ajetaan configure.

```
./configure --prefix=/tools
```

```
lfs:/mnt/lfs/sources/tar-1.29$ ./configure --prefix=/tools
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
```

Ajetaan make.

```
make
```

```
lfs:/mnt/lfs/sources/tar-1.29$ make
make all-recursive
make[1]: Entering directory '/mnt/lfs/sources/tar-1.29'
Making all in doc
make[2]: Entering directory '/mnt/lfs/sources/tar-1.29/doc'
make[2]: Nothing to be done for 'all'
```

Ajetaan testit. Tämä vaihe ei ole pakollinen.

```
make check
```

```
## ----- ##
## Test results. ##
## ----- ##

187 tests were successful.
17 tests were skipped.
make[2]: Leaving directory '/mnt/lfs/sources/tar-1.29/tests'
make[1]: Leaving directory '/mnt/lfs/sources/tar-1.29/tests'
make[1]: Entering directory '/mnt/lfs/sources/tar-1.29'
make[1]: Leaving directory '/mnt/lfs/sources/tar-1.29'
lfs:/mnt/lfs/sources/tar-1.29$ █
```

Asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/tar-1.29$ make install
Making install in doc
make[1]: Entering directory '/mnt/lfs/sources/tar-1.29/doc'
make[2]: Entering directory '/mnt/lfs/sources/tar-1.29/doc'
make[2]: Nothing to be done for 'install-exec-am'.
/tools/bin/mkdir -p '/tools/share/info'
```

Lopuksi poistetaan purettu hakemisto.

```
cd ..
rm -rf tar-1.29
```

```
lfs:/mnt/lfs/sources/tar-1.29$ cd ..
lfs:/mnt/lfs/sources$ rm -rf tar-1.29
lfs:/mnt/lfs/sources$ █
```

2.8.30 Texinfo-6.3

Arvioitu käännösaika: 0,2 SBU

Ensimmäiseksi puretaan Texinfo-paketti ja siirrytään purettuun hakemistoon.

```
tar xf texinfo-6.3.tar.xz
cd texinfo-6.3
```

```
lfs:/mnt/lfs/sources$ tar xf texinfo-6.3.tar.xz
lfs:/mnt/lfs/sources$ cd texinfo-6.3
lfs:/mnt/lfs/sources/texinfo-6.3$
```

Ajetaan configure.

```
./configure --prefix=/tools
```

Osana configure-prosessia ajetaan testi, josta tulee error liittyen TestXS_la-TestXS.lo. Se ei ole oleellinen LFS:n kannalta ja sen voi turvallisesti ignorata.

```
TestXS.xs:4:20: fatal error: EXTERN.h: No such file or directory
#include "EXTERN.h"
```

Ajetaan make.

```
make
```

```
lfs:/mnt/lfs/sources/texinfo-6.3$ make
make all-recursive
make[1]: Entering directory '/mnt/lfs/sources/texinfo-6.3'
Making all in gnulib/lib
make[2]: Entering directory '/mnt/lfs/sources/texinfo-6.3/gnulib/lib'
```

Ajetaan testit. Tämä vaihe ei ole pakollinen.

```
make check
```

Asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/texinfo-6.3$ make install
Making install in gnulib/lib
make[1]: Entering directory '/mnt/lfs/sources/texinfo-6.3/gnulib/lib'
make install-recursive
make[2]: Entering directory '/mnt/lfs/sources/texinfo-6.3/gnulib/lib'
make[3]: Entering directory '/mnt/lfs/sources/texinfo-6.3/gnulib/lib'
make[4]: Entering directory '/mnt/lfs/sources/texinfo-6.3/gnulib/lib'
```

Lopuksi poistetaan purettu hakemisto.

```
cd ..
rm -rf texinfo-6.3
```

```
lfs:/mnt/lfs/sources/texinfo-6.3$ cd ..
lfs:/mnt/lfs/sources$ rm -rf texinfo-6.3
lfs:/mnt/lfs/sources$
```

2.8.31 Util-linux-2.29.1

Arvioitu käännösaika: 0,9 SBU

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon.

```
tar xf util-linux-2.29.1.tar.xz
cd util-linux-2.29.1
```

```
lfs:/mnt/lfs/sources$ tar xf util-linux-2.29.1.tar.xz
lfs:/mnt/lfs/sources$ cd util-linux-2.29.1
lfs:/mnt/lfs/sources/util-linux-2.29.1$
```

Ajetaan configure.

```
./configure --prefix=/tools \
            --without-python \
            --disable-makeinstall-chown \
            --without-systemdsystemunitdir \
            PKG_CONFIG=""
```

```
lfs:/mnt/lfs/sources/util-linux-2.29.1$ ./configure --prefix=/tools \
> --without-python \
> --disable-makeinstall-chown \
> --without-systemdsystemunitdir \
> PKG_CONFIG=""
checking for gcc... gcc
checking whether the C compiler works... yes
```

Puretaan auki:

- `--without-python` → Estää käyttämästä Pythonia, jos se on asennettu isäntäjärjestelmään.
- `--disable-makeinstall-chown` → Estää käyttämästä `chown`-komentoa asennuksen aikana. Sitä ei tarvita, kun asennetaan `/tools`-hakemistoon ja täten myös vältetään rootilla asentamisen tarve.
- `--without-systemdsystemunitdir` → Järjestelmillä, jotka käyttävät `systemd`, paketti yrittää asentaa `systemd`-tiedoston olemattomaan hakemistoon `/tools`-hakemistossa. Tämä asetus estää sen.

- `PKG_CONFIG=""` → Tämä ympäristömuuttujan asettaminen estää tarpeettomien toimintojen lisäämisen, jotka saattavat olla saatavilla isännällä.

Ajetaan make.

```
make
```

```
lfs:/mnt/lfs/sources/util-linux-2.29.1$ make
make all-recursive
make[1]: Entering directory '/mnt/lfs/sources/util-linux-2.29.1'
Making all in po
make[2]: Entering directory '/mnt/lfs/sources/util-linux-2.29.1/po'
make[2]: Nothing to be done for 'all'
```

Asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/util-linux-2.29.1$ make install
make install-recursive
make[1]: Entering directory '/mnt/lfs/sources/util-linux-2.29.1'
Making install in po
make[2]: Entering directory '/mnt/lfs/sources/util-linux-2.29.1/po'
```

Lopuksi poistetaan purettu lähdehakemisto.

```
cd ..
rm -rf util-linux-2.29.1
```

```
lfs:/mnt/lfs/sources/util-linux-2.29.1$ cd ..
lfs:/mnt/lfs/sources$ rm -rf util-linux-2.29.1
lfs:/mnt/lfs/sources$
```

2.8.32 Xz-5.2.3

Arvioitu käännösaika: 0,2 SBU

Ensimmäiseksi puretaan xz ja siirrytään purettuun hakemistoon.

```
tar xf xz-5.2.3.tar.xz
cd xz-5.2.3
```

```
lfs:/mnt/lfs/sources$ tar xf xz-5.2.3.tar.xz
lfs:/mnt/lfs/sources$ cd xz-5.2.3
lfs:/mnt/lfs/sources/xz-5.2.3$
```

Ajetaan configure.

```
./configure --prefix=/tools
```

```
lfs:/mnt/lfs/sources/xz-5.2.3$ ./configure --prefix=/tools
XZ Utils 5.2.3

System type:
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu

Configure options:
checking if debugging code should be compiled... no
```

Ajetaan make.

```
make
```

Ajetaan testit. Tämä vaihe ei ole pakollinen.

```
make check
```

```
=====
All 9 tests passed
=====
```

Asennetaan.

```
make install
```

```
lfs:/mnt/lfs/sources/xz-5.2.3$ make install
Making install in src
make[1]: Entering directory '/mnt/lfs/sources/xz-5.2.3/src'
Making install in liblzma
make[2]: Entering directory '/mnt/lfs/sources/xz-5.2.3/src/liblzma'
```

Lopuksi poistetaan purettu lähdehakemisto.

```
cd ..
rm -rf xz-5.2.3
```

```
lfs:/mnt/lfs/sources/xz-5.2.3$ cd ..
lfs:/mnt/lfs/sources$ rm -rf xz-5.2.3
lfs:/mnt/lfs/sources$
```

2.8.33 Stripping

Tämän osio ei ole pakollinen, mutta jos LFS-partitio on pieni, on hyödyllistä tietää, että tarpeettomia asioita voi poistaa. Tähän mennessä käännetty suoritettavat tiedostot ja kirjastot sisältävät noin 70 MB tarpeettomia debuggaussymboleita. Poistetaan nämä symbolit:

```
strip --strip-debug /tools/lib/*  
/usr/bin/strip --strip-unneeded /tools/{,s}bin/*
```

```
lfs:/mnt/lfs/sources$ strip --strip-debug /tools/lib/*  
strip: Warning: '/tools/lib/audit' is not an ordinary file  
strip: Warning: '/tools/lib/bash' is not an ordinary file  
strip: Warning: '/tools/lib/expect5.45' is not an ordinary file  
strip: Warning: '/tools/lib/gawk' is not an ordinary file
```

Nämä komennot ohittavat monia tiedostoja raportoiden, että tiedostoformaattia ei tunnisteta. Monet niistä ovat binäärien sijaan skriptejä. Käytetään system strip -komentoa sisällyttääksemme strip binääriin /tools-hakemistossa.

--strip-unneeded -asetusta ei saa käyttää kirjastoille. Staattiset kirjastot tuhoitaisiin ja työkaluketjun paketit pitäisi asentaa uudestaan.

Jos haluaa säästää lisää tilaa, voi poistaa myös dokumentaation:

```
rm -rf /tools/{,share}/{info,man,doc}
```

Tässä vaiheessa \$LFS-partitiossa pitäisi olla vähintään 3 GB vapaata tilaa, jotka voidaan käyttää Glibc:n ja Gcc:n asentamiseen seuraavassa kappaleessa. Jos voit kääntää ja asentaa Glibc:n, voit kääntää ja asentaa myös kaikki loput paketit.

2.8.34 \$LFS/tools -hakemiston omistajuuden siirtäminen

Tästä eteenpäin kaikki kirjan loput komennot pitää ajaa pääkäyttäjänä (`root`). Pääkäyttäjäksi kirjautumisen jälkeen on syytä tarkistaa, että \$LFS on asetettu myös pääkäyttäjän ympäristössä oikein.

Tällä hetkellä \$LFS/tools -hakemiston omistaa lfs-käyttäjä, joka on olemassa ainoastaan isäntäjärjestelmässä. Jos \$LFS/tools -hakemiston pitää ennallaan, tiedostot omistaa user ID ilman ID:tä vastaavaa tiliä. Sitä emme toki halua, koska se johtaisi siihen, että myöhemmin luotava käyttäjätili voisi saada kyseisen user ID:n. Silloin tämä luotu käyttäjä omistaisi \$LFS/tools -hakemiston ja kaikki sen sisältämät tiedostot.

Välttääksemme yllä kuvaillun ongelman, siirretään \$LFS/tools -hakemiston omistajuus rootille:

```
chown -R root:root $LFS/tools
```

```
root:/home/lfs# chown -R root:root $LFS/tools
root:/home/lfs#
```

HUOM! Mikäli tarkoituksenasi on säilyttää tilapäiset työkalut muiden LFS-järjestelmien kääntämistä varten tulevaisuudessa, nyt on aika ottaa ne talteen. Seuraavan kappaleen komennot muuttavat työkaluja tehden niistä hyödyttömät tulevia käännöksiä ajatellen.

2.9 Järjestelmän perusohjelmien asentaminen

Tässä kappaleessa etenemme käännösvaiheeseen ja alamme kunnolla rakentaa LFS-järjestelmää. Ensin siirrymme chroot-komennolla tilapäiseen minimaaliseen Linux-järjestelmään, teemme muutamia viimeisiä valmisteluja ja sen jälkeen alamme asentaa paketteja.

Ohjelmien asentaminen on yksinkertaista ja mutkatonta. Vaikka monissa tapauksissa asennusohjeet voisivat olla lyhyemmässä ja yleisemmässä muodossa, tässä kirjassa tarjotaan täydelliset ohjeet jokaiselle paketille mahdollisten virheiden minimoimiseksi.

Varsinkin LFS-ensikertalaisten kannattaa noudattaa kirjan ohjeita pilkulleen. Esimerkiksi `-march` ja `-mtune` -asetuksilla muita kuin kirjassa olevia arvoja ei ole testattu. Ne voivat aiheuttaa ongelmia työkaluketjupakettien (Binutils, GCC ja Glibc) kanssa.

Tämän kirjan pakettien asennusjärjestystä pitää noudattaa tarkasti. Eri paketteja ei saa kääntää samanaikaisesti rinnan. Se voi säästää aikaa (varsinkin dual-CPU -koneilla), mutta voi johtaa siihen, että ohjelma sisältää kiinteän polun `/tools` -hakemistoon. Siinä tapauksessa ohjelma lakkaa toimimasta, kun kyseinen hakemisto poistetaan.

Ennen asennusohjeita jokaisen paketin kohdalla kerrotaan, mitä ohjelmia ja kirjastoja asennetaan ja kauanko paketin kääntämiseen menee (SBU-arvot). Huomaa, että SBU-arvot kappaleessa [2.15](#) sisältävät testit.

2.10 Muutama sana kirjastoista

Staatlisten kirjastojen kääntämistä ja asentamista ei suositella. Moderneissa Linux-järjestelmissä useimpien staatlisten kirjastojen perimmäinen tarkoitus on vanhentunut. Staatlisten kirjaston linkittäminen ohjelmaan on usein epäkäytännöllistä. Mikäli kirjastoon tarvitaan päivitystä esimerkiksi tietoturvaongelman poistamiseksi, kaikki ohjelmat, jotka käyttävät staatlista kirjastoa pitää kääntää ja linkittää uuteen kirjastoon.

Tässä kappaleessa esitetään useimpien staatlisten kirjastojen asentuminen käyttämällä `--disable-static` -asetusta. Joissakin tapauksissa (erityisesti Glibc ja GCC) staatlisten kirjastojen käyttö kuitenkin on välttämätöntä.

Lisää jaetuista kirjastoista: https://www.linux.fi/wiki/Jaetut_kirjastot

2.11 Kernelin virtuaalisten tiedostojärjestelmien valmisteleminen

Kernelin tiedostojärjestelmiä käytetään kommunikoimaan sekä kerneliin että siitä poispäin. Nämä tiedostojärjestelmät ovat virtuaalisia eikä niitä varten käytetä yhtään levytilaa. Näiden tiedostojärjestelmien sisältö on tietokoneen väliaikaismuistissa.

Aloitetaan luomalla hakemistoja, joihin tiedostojärjestelmät [liitetään](#).

```
mkdir -pv $LFS/{dev,proc,sys,run}
```

```
root:/home/lfs# mkdir -pv $LFS/{dev,proc,sys,run}
mkdir: created directory '/mnt/lfs/dev'
mkdir: created directory '/mnt/lfs/proc'
mkdir: created directory '/mnt/lfs/sys'
mkdir: created directory '/mnt/lfs/run'
root:/home/lfs#
```

2.11.1 Varhaisten laitetiedostojen luominen

Kun kernel käynnistää järjestelmän, se vaatii muutamien [laitetiedostojen](#) olemassa olon (`console` ja `null`). Nämä laitetiedostot pitää luoda kovalevyille, jotta ne ovat saatavilla ennen kuin `udev` on käynnistynyt ja kun Linux käynnistetään `init=/bin/bash:n` kanssa. `/dev/console` ja `/dev/null` ovat siitä poikkeuksellisia laitetiedostoja, että ne ovat `/dev`-hakemiston alla pysyvästi, eli myös silloin, kun `/dev`-hakemisto ei ole liitetty. Jos nämä kaksi tiedostoa puuttuvat, järjestelmä on käynnistyskelvoton. Luodaan laitetiedostot seuraavilla komennoilla:

```
mknod -m 600 $LFS/dev/console c 5 1
mknod -m 666 $LFS/dev/null c 1 3
```

```
root:/home/lfs# mknod -m 600 $LFS/dev/console c 5 1
root:/home/lfs# mknod -m 666 $LFS/dev/null c 1 3
root:/home/lfs#
```

2.11.2 /dev -hakemiston liittäminen ja täyttäminen

Suosittelut metodi /dev -hakemiston täyttämiseen laitteilla on liittää virtuaalinen tiedostojärjestelmä (kuten tmpfs) /dev -hakemistoon ja sallia laitteiden dynaaminen luominen tähän virtuaaliseen tiedostojärjestelmään, kun ne tunnistetaan tai niitä käytetään. Laitteiden luomisen hoitaa yleensä käynnistysprosessin aikana Udev. Koska tässä uudessa järjestelmässä ei vielä ole Udevia eikä sitä vielä ole käynnistetty, on tarpeellista liittää ja täyttää /dev manuaalisesti. Tämä saavutetaan 'bind mounttaamalla' (eli `mount` komento `--bind` parametrilla) isäntäjärjestelmän /dev -hakemisto. Komennolla luodaan hakemiston tai liitospisteen peili-kuva johonkin toiseen paikkaan:

```
mount -v --bind /dev $LFS/dev
```

```
root:/home/lfs# mount -v --bind /dev $LFS/dev
/dev on /mnt/lfs/dev type none (rw,bind)
root:/home/lfs#
```

2.11.3 Kernelin virtuaalisten tiedostojärjestelmien liittäminen

Liitetään jäljellä olevat kernelin virtuaaliset tiedostojärjestelmät:

```
mount -vt devpts devpts $LFS/dev/pts -o gid=5,mode=620
mount -vt proc proc $LFS/proc
mount -vt sysfs sysfs $LFS/sys
mount -vt tmpfs tmpfs $LFS/run
```

```
root:/home/lfs# mount -vt devpts devpts $LFS/dev/pts -o gid=5,mode=620
devpts on /mnt/lfs/dev/pts type devpts (rw,gid=5,mode=620)
root:/home/lfs# mount -vt proc proc $LFS/proc
proc on /mnt/lfs/proc type proc (rw)
root:/home/lfs# mount -vt sysfs sysfs $LFS/sys
sysfs on /mnt/lfs/sys type sysfs (rw)
root:/home/lfs# mount -vt tmpfs tmpfs $LFS/run
tmpfs on /mnt/lfs/run type tmpfs (rw)
root:/home/lfs#
```

Puretaan auki `devpts:n` mount-asetukset:

- `gid=5` → Tämä varmistaa, että kaikki `devpts`-luodut laitetiedostot omistaa ryhmä-ID 5. Tätä ID:tä käytämme myöhemmin `tty`-ryhmää varten. Käytämme ryhmän ID:tä nimen sijaan, koska isäntäjärjestelmä saattaa käyttää eri ID:tä sen `tty`-ryhmälle.
- `mode=0620` → Tämä varmistaa sen, että kaikilla `devpts`-luoduilla laitetiedostoilla on tila 0620 (eli käyttäjä pystyy lukemaan ja kirjoittamaan, ryhmä kirjoittamaan). Yhdessä `gid=5` -asetuksen kanssa tällä varmistetaan, että `devpts` luo laitetiedostot, jotka täyttävät `grantpt()`:n vaatimukset. Käytännössä tämä tarkoittaa, että Glibc `pt_chown` -helper binary (jota ei asenneta oletuksena) ei ole tarpeellinen.

Joissakin isäntäjärjestelmissä `/dev/shm` on symbolinen linkki `/run/shm`. Koska `/run` liitettiin äsken, ainoastaan hakemisto täytyy luoda.

```
if [ -h $LFS/dev/shm ]; then
    mkdir -pv $LFS/$(readlink $LFS/dev/shm)
fi
```

```
root:/home/lfs# if [ -h $LFS/dev/shm ]; then
>   mkdir -pv $LFS/$(readlink $LFS/dev/shm)
> fi
root:/home/lfs#
```

2.12 Chroot-ympäristöön siirtyminen

Seuraavaksi siirrymme [chroot](#)-ympäristöön, jotta voimme aloittaa lopullisen LFS-järjestelmän rakentamisen ja asentamisen. Ajetaan seuraava komento pääkäyttäjänä:

```
chroot "$LFS" /tools/bin/env -i \  
    HOME=/root \\  
    TERM="$TERM" \\  
    PS1='\u:\w\$ ' \\  
    PATH=/bin:/usr/bin:/sbin:/usr/sbin:/tools/bin \  
    /tools/bin/bash --login +h
```

`env`-komennon `-i` -asetus poistaa kaikki muuttujat `chroot`-ympäristöstä. Tämän jälkeen vain `HOME`, `TERM`, `PS1` ja `PATH` -muuttujat asetetaan uudelleen. `TERM=$TERM` asettaa `TERM`-muuttujan `chrootin` sisällä samaksi arvoksi kuin `chrootin` ulkopuolella. Jotkin ohjelmat, kuten `vim` ja `less` tarvitsevat tämän muuttujan toimiakseen oikein.

Tästä eteenpäin ei ole enää tarvetta käyttää LFS-muuttujaa, koska kaikki työskentely on rajattu LFS-tiedostojärjestelmään. Tämä johtuu siitä, että Bash:lle kerrotaan, että `$LFS` on itseasiassa `/` (root) -hakemisto.

Huomaa, että `/tools/bin` tulee viimeiseksi `PATH`-muuttujassa. Tämä tarkoittaa sitä, että tilapäisiä työkaluja ei enää käytetä sen jälkeen, kun lopulliset versiot on asennettu. Koska shell ei muista suoritettujen binäärien paikkoja, [hashing](#) poistetaan päältä `+h` -asetuksella.

```
root:/home/lfs# chroot "$LFS" /tools/bin/env -i \  
>     HOME=/root \\  
>     TERM="$TERM" \\  
>     PS1='\u:\w\$ ' \\  
>     PATH=/bin:/usr/bin:/sbin:/usr/sbin:/tools/bin \  
>     /tools/bin/bash --login +h  
I have no name!:/#
```

Bash sanoo nyt "I have no name!". Se on normaalia, koska `/etc/passwd` -tiedostoa ei ole vielä luotu.

HUOM! On tärkeää, että kaikki tämän kappaleen komennot tästä lähtien ajetaan chroot-ympäristön sisällä. Jos poistut tästä chroot-ympäristöstä jostakin syystä (esim. uudelleenkäynnistys), varmista, että kernelin virtuaaliset tiedostojärjestelmät on liitetty, kuten kappaleessa [2.11](#) kerrottiin. Siirry myös takaisin chroot-ympäristöön ennen kuin jatkat asennusta.

2.13 Hakemistojen luominen

Luodaan standardin mukainen hakemistopuu seuraavilla komennoilla:

```
mkdir -pv /{bin,boot,etc}/{opt,syscon-  
fig},home,lib/firmware,mnt,opt}  
mkdir -pv /{media/{floppy,cdrom},sbin,srv,var}  
install -dv -m 0750 /root  
install -dv -m 1777 /tmp /var/tmp  
mkdir -pv /usr/{,local/}{bin,include,lib,sbin,src}  
mkdir -pv /usr/{,local/}share/{color,dict,doc,info,lo-  
cale,man}  
mkdir -v /usr/{,local/}share/{misc,terminfo,zoneinfo}  
mkdir -v /usr/libexec  
mkdir -pv /usr/{,local/}share/man/man{1..8}  
  
case $(uname -m) in  
  x86_64) mkdir -v /lib64 ;;  
esac  
  
mkdir -v /var/{log,mail,spool}  
ln -sv /run /var/run  
ln -sv /run/lock /var/lock  
mkdir -pv /var/{opt,cache,lib/{color,misc,locate},local}
```

Hakemistot luodaan oletuksena käyttöoikeuksilla 755, mutta näin ei haluta kaikkien hakemistojen kohdalla. Yllä olevissa komennoissa on tehty kaksi muutosta. Ensimmäinen muutos varmistaa, että kuka tahansa ei pysty siirtymään `/root` -hakemistoon. Toinen muutos varmistaa, että kuka tahansa käyttäjä pystyy kirjoittamaan `/tmp` ja `/var/tmp` -hakemistoihin, mutta ei pysty poistamaan toisen käyttäjän tiedostoja niistä.

```
<bin,boot,etc/{opt,sysconfig},home,lib/firmware,mnt,opt}
mkdir: created directory '/bin'
mkdir: created directory '/etc'
mkdir: created directory '/etc/opt'
mkdir: created directory '/etc/sysconfig'
mkdir: created directory '/lib'
mkdir: created directory '/lib/firmware'
mkdir: created directory '/mnt'
I have no name!:/# mkdir -pv /{media/{floppy,cdrom},sbin,svr,var}
mkdir: created directory '/media'
mkdir: created directory '/media/floppy'
mkdir: created directory '/media/cdrom'
mkdir: created directory '/sbin'
```

```
mkdir: created directory '/usr/local/share/man/man4'
mkdir: created directory '/usr/local/share/man/man5'
mkdir: created directory '/usr/local/share/man/man6'
mkdir: created directory '/usr/local/share/man/man7'
mkdir: created directory '/usr/local/share/man/man8'
I have no name!:/#
I have no name!:/# case $(uname -m) in
> x86_64) mkdir -v /lib64 ;;
> esac
mkdir: created directory '/lib64'
I have no name!:/#
I have no name!:/# mkdir -v /var/{log,mail,spool}
mkdir: created directory '/var/log'
mkdir: created directory '/var/mail'
mkdir: created directory '/var/spool'
I have no name!:/# ln -sv /run /var/run
'/var/run' -> '/run'
I have no name!:/# ln -sv /run/lock /var/lock
'/var/lock' -> '/run/lock'
```

```
I have no name!:/# mkdir -pv /var/{opt,cache,lib/{color,misc,locate},local}
mkdir: created directory '/var/opt'
mkdir: created directory '/var/cache'
mkdir: created directory '/var/lib'
mkdir: created directory '/var/lib/color'
mkdir: created directory '/var/lib/misc'
mkdir: created directory '/var/lib/locate'
mkdir: created directory '/var/local'
```

LFS:n hakemistorakenne tarkempine selityksineen on kuvattu [liitteessä 5](#).

2.13.1 FHS

Hakemistopuu perustuu [Filesystem Hierarchy Standardiin](https://wiki.linuxfoundation.org/en/FHS) (FHS): <https://wiki.linuxfoundation.org/en/FHS>

FHS-standardin mukaista on myös luoda vapaaehtoisia hakemistoja, kuten `/usr/local/games` ja `/usr/share/games`. Tässä kirjassa luomme vain tarvittavat hakemistot, mutta kukin lukija tekee mitä lystää.

2.14 Tarpeellisten tiedostojen ja symbolisten linkkien luominen

Jotkin ohjelmat käyttävät kiinteitä polkuja toisiin ohjelmiin, joita ei ole vielä olemassa. Tyydyttääksemme näiden ohjelmien tarpeet, luodaan symbolisia linkkejä, jotka oikeat tiedostot korvaavat sitä mukaa, kun ohjelmia asennetaan.

```
ln -sv /tools/bin/{bash,cat,echo,pwd,stty} /bin
ln -sv /tools/bin/perl /usr/bin
ln -sv /tools/lib/libgcc_s.so{,.1} /usr/lib
ln -sv /tools/lib/libstdc++.so{,.6} /usr/lib
sed 's/tools/usr/' /tools/lib/libstdc++.la >
/usr/lib/libstdc++.la
ln -sv bash /bin/sh
```

Puretaan auki:

- `/bin/bash` → Monet bash-skriptit määrittävät `/bin/bash:n`.
- `/bin/cat` → Tämä polku on [kovakoodattu](#) Glibc:n configure-skriptiin.
- `/bin/echo` → Yksi Glibc:n testeistä vaatii tämän.
- `/bin/pwd` → Joissakin Glibc:n configure-skripteissä on tämä polku kovakoodattu.
- `/bin/sty` → Tämä polku on kovakoodattu Expect-pakettiin. Sitä tarvitaan Binutilsin ja GCC:n testien suorittamiseen.
- `/usr/bin/perl` → Monissa Perlin skripteissä on tämä kovakoodattuna.
- `/usr/lib/libgcc_s.so{,.1}` → Glibc tarvitsee tämän, jotta pthreads-kirjasto toimii.
- `/usr/lib/libstdc++{,.6}` → Monet Glibc:n testit tarvitsevat tämän. Myös GMP:n C++ -tuki tarvitsee tämän.

- `/usr/lib/libstdc++.la` → Tämä estää `/tools` -referenssin, joka muuten olisi `/usr/lib/libstdc++.la` :ssa GCC:n asennuksen jälkeen.
- `/bin/sh` → Monet shell-skriptit kovakoodaavat `/bin/sh`.

```
I have no name!:/# ln -sv /tools/bin/{bash,cat,echo,pwd,stty} /bin
'/bin/bash' -> '/tools/bin/bash'
'/bin/cat' -> '/tools/bin/cat'
'/bin/echo' -> '/tools/bin/echo'
'/bin/pwd' -> '/tools/bin/pwd'
'/bin/stty' -> '/tools/bin/stty'
I have no name!:/# ln -sv /tools/bin/perl /usr/bin
'/usr/bin/perl' -> '/tools/bin/perl'
I have no name!:/# ln -sv /tools/lib/libgcc_s.so{,.1} /usr/lib
'/usr/lib/libgcc_s.so' -> '/tools/lib/libgcc_s.so'
'/usr/lib/libgcc_s.so.1' -> '/tools/lib/libgcc_s.so.1'
I have no name!:/# ln -sv /tools/lib/libstdc++.so{,.6} /usr/lib
'/usr/lib/libstdc++.so' -> '/tools/lib/libstdc++.so'
'/usr/lib/libstdc++.so.6' -> '/tools/lib/libstdc++.so.6'
<s/usr/' /tools/lib/libstdc++.la > /usr/lib/libstdc++.la
I have no name!:/# ln -sv bash /bin/sh
'/bin/sh' -> 'bash'
I have no name!:/#
```

Historiallisesti Linux ylläpitää listaa liitetystä tiedostojärjestelmästä `/etc/mtab` -tiedostossa. Modernit kernelit ovat siirtyneet ylläpitämään tätä listaa sisäisesti ja näyttämään sen käyttäjälle `/proc` -hakemiston avulla. Tyydyttääksemme sellaisten työkalujen tarpeet, jotka odottavat `/etc/mtab` -tiedoston olevan olemassa, luodaan seuraava symbolinen linkki:

```
ln -sv /proc/self/mounts /etc/mtab
```

```
I have no name!:/# ln -sv /proc/self/mounts /etc/mtab
'/etc/mtab' -> '/proc/self/mounts'
I have no name!:/#
```

Jotta käyttäjä `root` pääsisi kirjautumaan ja nimi “root” tunnistettaisiin, täytyy luoda oikea sisältö `/etc/passwd` ja `/etc/group` -tiedostoihin.

Luodaan /etc/passwd -tiedosto ajamalla seuraava komento:

```
cat > /etc/passwd << "EOF"
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/dev/null:/bin/false
daemon:x:6:6:Daemon User:/dev/null:/bin/false
messagebus:x:18:18:D-Bus Message Daemon
User:/var/run/dbus:/bin/false
nobody:x:99:99:Unprivileged User:/dev/null:/bin/false
EOF
```

```
I have no name!:/# cat > /etc/passwd << "EOF"
> root:x:0:0:root:/root:/bin/bash
> bin:x:1:1:bin:/dev/null:/bin/false
> daemon:x:6:6:Daemon User:/dev/null:/bin/false
> messagebus:x:18:18:D-Bus Message Daemon User:/var/run/dbus:/bin/false
> nobody:x:99:99:Unprivileged User:/dev/null:/bin/false
> EOF
I have no name!:/# █
```

Salasana rootille on tässä vaiheessa x, mutta se asetetaan myöhemmin.

Luodaan /etc/group -tiedosto seuraavalla komennolla:

```
cat > /etc/group << "EOF"
root:x:0:
bin:x:1:daemon
sys:x:2:
kmem:x:3:
tape:x:4:
tty:x:5:
daemon:x:6:
floppy:x:7:
disk:x:8:
lp:x:9:
dialout:x:10:
audio:x:11:
video:x:12:
```

```
utmp:x:13:
usb:x:14:
cdrom:x:15:
adm:x:16:
messagebus:x:18:
systemd-journal:x:23:
input:x:24:
mail:x:34:
nogroup:x:99:
users:x:999:
EOF
```

```
I have no name!:/# cat > /etc/group << "EOF"
> root:x:0:
> bin:x:1:daemon
> sys:x:2:
> kmem:x:3:
> tape:x:4:
> tty:x:5:
> daemon:x:6:
> floppy:x:7:
> disk:x:8:
> lp:x:9:
> dialout:x:10:
> audio:x:11:
> video:x:12:
> utmp:x:13:
> usb:x:14:
> cdrom:x:15:
> adm:x:16:
> messagebus:x:18:
> systemd-journal:x:23:
> input:x:24:
> mail:x:34:
> nogroup:x:99:
> users:x:999:
> EOF
I have no name!:/#
```

Luodut ryhmät eivät ole osa mitään standardia. Ne on päätetty osittain Udev:n asetuksiin liittyvien vaatimusten ja osittain nykyisten Linux distrojen käytänteiden pohjalta. Lisäksi jotkin testit vaativat tietyt käyttäjät tai ryhmät. Linux Standard

Base (LSB) vaatii ainoastaan, että on olemassa ryhmä root, jolla on group ID (GID) 0, ja ryhmä bin, jolla on GID 1. Järjestelmänvalvoja voi vapaasti päättää kaikki muut ryhmät ja GID:t.

LSB:stä lisää täällä: <http://www.linuxbase.org>

Seuraavaksi poistetaan "I have no name!" -teksti shellistä. Käynnistetään uusi shell:

```
exec /tools/bin/bash --login +h
```

```
I have no name!:/# exec /tools/bin/bash --login +h
root:/#
```

Huomaa +h -asetuksen käyttö. Se poistaa käytöstä bash:n internal path hashingin. Ilman tätä asetusta bash säilyttäisi muistissa viittaukset binääreihin, joita se on suorittanut. Varmistaaksemme, että uusia käännettyjä binäärejä käytetään vanhojen sijasta, +h -asetusta käytetään koko loppukappaleen ajan.

login,agetty ja init -ohjelmat (ja jotkin muut) käyttävät lokitiedostoja tallentaakseen informaatiota, kuten kuka oli kirjautuneena järjestelmään ja milloin. Nämä ohjelmat eivät kirjoita lokitiedostoihin, jos ne eivät ole jo valmiiksi olemassa. Alustetaan lokitiedostot ja annetaan niille oikeat käyttöoikeudet:

```
touch /var/log/{btmp,lastlog,faillog,wtmp}
chgrp -v utmp /var/log/lastlog
chmod -v 664 /var/log/lastlog
chmod -v 600 /var/log/btmp
```

```
root:/# touch /var/log/{btmp,lastlog,faillog,wtmp}
root:/# chgrp -v utmp /var/log/lastlog
changed group of '/var/log/lastlog' from root to utmp
root:/# chmod -v 664 /var/log/lastlog
mode of '/var/log/lastlog' changed from 0644 (rw-r--r--) to 0664 (rw-rw-r--)
root:/# chmod -v 600 /var/log/btmp
mode of '/var/log/btmp' changed from 0644 (rw-r--r--) to 0600 (rw-----)
root:/#
```

Puretaan auki:

- `/var/log/wtmp` -tiedosto tallentaa kaikki sisään- ja uloskirjautumiset.
- `/var/log/lastlog` -tiedosto tallentaa tiedon siitä, milloin kukakin käyttäjä viimeksi kirjautui sisään.
- `/var/log/faillog` -tiedosto tallentaa käyttäjän epäonnistuneet kirjautumisyritykset.
- `/var/log/btmp` -tiedosto tallentaa epäonnistuneet kirjautumisyritykset.

HUOM! `/run/utmp` -tiedosto tallentaa tiedon käyttäjistä, jotka ovat käyttöhetkellä kirjautuneet sisään. Tämä tiedosto luodaan dynaamisesti käynnistyskripteissä.

2.15 Pakettien asentaminen

2.15.1 Linux-4.9.9 API Headers

Käännösaika: alle 0,1 SBU

Asennettavat header-tiedostot: `/usr/include/asm/*.h, /usr/include/asm-generic/*.h, /usr/include/drm/*.h, /usr/include/linux/*.h, /usr/include/misc/*.h, /usr/include/mtd/*.h, /usr/include/rdma/*.h, /usr/include/scsi/*.h, /usr/include/sound/*.h, /usr/include/video/*.h` ja `/usr/include/xen/*.h`

Asennettavat hakemistot: `/usr/include/asm, /usr/include/asm-generic, /usr/include/drm, /usr/include/linux, /usr/include/misc, /usr/include/mtd, /usr/include/rdma, /usr/include/scsi, /usr/include/sound, /usr/include/video` ja `/usr/include/xen`

Ensimmäiseksi puretaan paketti ja siirrytään luotuun hakemistoon.

```
cd /sources
tar xf linux-4.9.9.tar.xz
cd linux-4.9.9
```

```
root:/# cd /sources/
root:/sources# tar xf linux-4.9.9.tar.xz
root:/sources# cd linux-4.9.9
root:/sources/linux-4.9.9#
```

Seuraavaksi varmistetaan, ettei paketissa ole väljähtyneitä tiedostoja.

```
make mrproper
```

```
root:/sources/linux-4.9.9# make mrproper
root:/sources/linux-4.9.9#
```

Puretaan käyttäjälle näkyvät kernelin header-tiedostot lähdehakemistosta. Ne si-
joitetaan paikalliseen välittäjähakemistoon (intermediate directory) ja kopioidaan
tarvittavaan paikkaan, koska extraction-prosessi poistaa kaikki olemassa olevat
tiedostot kohdehakemistosta. On myös olemassa kernelin kehittäjien käyttämiä

piilotettuja tiedostoja, joita emme tarvitse LFS:ssä, joten poistetaan ne välittäjähakemistosta.

```
make INSTALL_HDR_PATH=dest headers_install
find dest/include \( -name .install -o -name ..install.cmd
\) -delete
cp -rv dest/include/* /usr/include
```

```
root:/sources/linux-4.9.9# make INSTALL_HDR_PATH=dest headers_install
CHK      include/generated/uapi/linux/version.h
UPD      include/generated/uapi/linux/version.h
HOSTCC   scripts/basic/fixdep
WRAP     arch/x86/include/generated/asm/clkdev.h
```

```
<ude \( -name .install -o -name ..install.cmd \) -delete
root:/sources/linux-4.9.9# cp -rv dest/include/* /usr/include
'dest/include/asm' -> '/usr/include/asm'
'dest/include/asm/a.out.h' -> '/usr/include/asm/a.out.h'
'dest/include/asm/auxvec.h' -> '/usr/include/asm/auxvec.h'
```

Lopuksi poistetaan purettu hakemisto.

```
cd ..
rm -rf linux-4.9.9
```

```
root:/sources/linux-4.9.9# cd ..
root:/sources# rm -rf linux-4.9.9
root:/sources#
```

2.15.2 Man-pages-4.09

Käännösaika: alle 0,1 SBU

Asennettavat tiedostot: useita man-sivuja

Ensimmäiseksi puretaan man-pages ja siirrytään purettuun hakemistoon.

```
tar xf man-pages-4.09.tar.xz
cd man-pages-4.09
```

```
root:/sources# tar xf man-pages-4.09.tar.xz
root:/sources# cd man-pages-4.09
root:/sources/man-pages-4.09#
```

Asennetaan man-pages.

```
make install
```

```
root:/sources/man-pages-4.09# make install
for i in man?; do \
    install -d -m 755 /usr/share/man/"$i" || exit $?; \
    install -m 644 "$i"/* /usr/share/man/"$i" || exit $?; \
done
root:/sources/man-pages-4.09#
```

Lopuksi poistetaan purettu hakemisto.

```
cd ..
rm -rf man-pages-4.09
```

```
root:/sources/man-pages-4.09# cd ..
root:/sources# rm -rf man-pages-4.09
root:/sources#
```

2.15.3 Glibc-2.25

Käännösaika: 20 SBU

Asennettavat ohjelmat: *catchsegv, gencat, getconf, getent, iconv, iconvconfig, ldconfig, ldd, lddlibc4, locale, localedef, makedb, mtrace, nscd, pldd, rpcgen, sln, sotruss, sprof, tzselect, xtrace, zdump ja zic*

Asennettavat kirjastot: *ld-2.25.so, libBrokenLocale.{a,so}, libSegFault.so, libanl.{a,so}, libc.{a,so}, libc_nonshared.a, libcidn.so, libcrypt.{a,so}, libdl.{a,so}, libg.a, libieee.a, libm.{a,so}, libmcheck.a, libmemusage.so, libnsl.{a,so}, libnss_compat.so, libnss_dns.so, libnss_files.so, libnss_hesiod.so, libnss_nis.so, libnss_nisplus.so, libpthread.{a,so}, libpthread_nonshared.a, libresolv.{a,so}, librpcsvc.a, librt.{a,so}, libthread_db.so, ja libutil.{a,so}*

Asennettavat hakemistot: */usr/include/arpa, /usr/include/bits, /usr/include/gnu, /usr/include/net, /usr/include/netash, /usr/include/netatalk, /usr/include/netax25, /usr/include/neteconet, /usr/include/netinet, /usr/include/netipx, /usr/include/netiucv, /usr/include/netpacket, /usr/include/netrom, /usr/include/netrose, /usr/include/nfs, /usr/include/protocols, /usr/include/rpc, /usr/include/rpcsvc, /usr/include/sys, /usr/lib/audit, /usr/lib/gconv, /usr/lib/locale, /usr/libexec/getconf, /usr/share/i18n, /usr/share/zoneinfo, /var/cache/nscd ja /var/lib/nss_db*

HUOM! Glibc:n käännösjärjestelmä on omavarainen ja asentuu kokonaisuutena, vaikka kääntäjän specs -tiedosto ja linkkeri osoittavat `/tools` -hakemistoon. Specsia ja linkkeriä ei voi muokata ennen kuin Glibc on asentunut, koska Glibc:n autoconf-testit antaisivat vääriä tuloksia, eikä saavutettaisi 'puhdasta' käännöstä.

Ensimmäiseksi puretaan Glibc ja siirrytään purettuun hakemistoon.

```
tar xf glibc-2.25.tar.xz
cd glibc-2.25
```

```
root:/sources# tar xf glibc-2.25.tar.xz
root:/sources# cd glibc-2.25
root:/sources/glibc-2.25#
```

Jotkin Glibc-ohjelmat käyttävät FHS-epäyhteensopivaa `/var/db` -hakemistoa varastoidakseen runtime datansa. Lisätään korjaustiedosto, jotta sellaiset ohjelmat varastoivat runtime datansa FHS-yhteensopiviin paikkoihin.

```
patch -Np1 -i ../glibc-2.25-fhs-1.patch
```

```
root:/sources/glibc-2.25# patch -Np1 -i ../glibc-2.25-fhs-1.patch
patching file Makeconfig
Hunk #1 succeeded at 260 (offset 10 lines).
patching file nscd/nscd.h
Hunk #1 succeeded at 161 (offset 49 lines).
patching file nss/db-Makefile
patching file sysdeps/generic/paths.h
patching file sysdeps/unix/sysv/linux/paths.h
root:/sources/glibc-2.25#
```

Luodaan LSB:n vaatima symbolinen linkki ja x86_64 -koneille dynaamisen lataa-
jan vaatima symbolinen linkki:

```
case $(uname -m) in
    x86) ln -s ld-linux.so.2 /lib/ld-lsb.so.3
        ;;
    x86_64) ln -s ../lib/ld-linux-x86-64.so.2 /lib64
             ln -s ../lib/ld-linux-x86-64.so.2 /lib64/ld-
lsb-x86-64.so.3
        ;;
esac
```

```
root:/sources/glibc-2.25# case $(uname -m) in
>     x86) ln -s ld-linux.so.2 /lib/ld-lsb.so.3
>         ;;
>     x86_64) ln -s ../lib/ld-linux-x86-64.so.2 /lib64
>              ln -s ../lib/ld-linux-x86-64.so.2 /lib64/ld-lsb-x86-64.so.3
>         ;;
> esac
root:/sources/glibc-2.25#
```

Luodaan build-hakemisto ja siirrytään sinne.

```
mkdir -v build
cd      build
```

```
root:/sources/glibc-2.25# mkdir -v build
mkdir: created directory 'build'
root:/sources/glibc-2.25# cd build/
root:/sources/glibc-2.25/build#
```

Ajetaan configure.

```
../configure --prefix=/usr --enable-kernel=2.6.32 --enable-  
obsolete-rpc --enable-stack-protector=strong libc_cv_slib-  
dir=/lib
```

```
root:/sources/glibc-2.25/build# ../configure --prefix=/usr  
> --enable-kernel=2.6.32 \  
> --enable-obsolete-rpc \  
> --enable-stack-protector=strong \  
> libc_cv_slibdir=/lib  
checking build system type... x86_64-pc-linux-gnu  
checking host system type... x86_64-pc-linux-gnu  
checking for gcc... gcc  
checking for suffix of object files... o  
checking whether we are using the GNU C compiler... yes
```

Ajetaan make.

```
make
```

Seuraavaksi ajetaan testit.

HUOM! Tässä vaiheessa Glibc:n testien ajaminen on erittäin tärkeää. Niitä *ei saa* ohittaa missään olosuhteissa.

```
make check
```

```
Summary of test results:  
  3 FAIL  
2645 PASS  
 21 UNSUPPORTED  
 43 XFAIL  
  2 XPASS  
FAIL: posix/tst-getaddrinfo4  
FAIL: posix/tst-getaddrinfo5  
FAIL: rt/tst-cputimer1
```

Testeissä todennäköisesti tulee joitakin virheitä. Glibc:n testit ovat osittain riippuvaisia isäntäjärjestelmästä. Alla lista yleisimmistä ilmenneistä ongelmista tämän LFS-version kanssa:

- `posix/tst-getaddrinfo4` testi epäonnistuu aina, koska ei ole ole-massa tarvittavia verkotusohjelmia, kun testit ajetaan. `posix/tst-getaddrinfo5` epäonnistuu myös joillakin arkkitehtuureilla.
- `rt/tst-cputimer1` ja `rt/tst-cpuclock2` -testit epäonnistuvat syystä, jota ei täysin selvitetty, mutta arvelemme, että ne johtuvat ajoitus-ongelmista.
- Matematiikkatestit epäonnistuvat joskus järjestelmillä, joissa CPU ei ole suhteellisen uusi Intel tai AMD -prosessori.
- `nptl/tst-thread-affinity-{pthread,pthread2,sched}` -testit saattavat epäonnistua syistä, joita ei ole määritelty.
- `malloc/tst-malloc-usable` ja `nptl/tst-cleanupx4` -testit saattavat myös epäonnistua joillain arkkitehtuureilla.

Listan virheistä ei kuitenkaan tarvitse huolestua.

Glibc valittaa asennusvaiheessa `/etc/ld.so.conf` -tiedoston puuttumista. Esitetään harmiton, mutta ärsyttävä varoitus:

```
touch /etc/ld.so.conf
```

```
root:/sources/glibc-2.25/build# touch /etc/ld.so.conf
root:/sources/glibc-2.25/build#
```

Asennetaan paketti:

```
make install
```

```
Your new glibc installation seems to be ok.
```

Asennetaan `nscd`:tä varten konfiguraatiotiedosto ja runtime-hakemisto.

```
cp -v ../nscd/nscd.conf /etc/nscd.conf
mkdir -pv /var/cache/nscd
```

```
root:/sources/glibc-2.25/build# cp -v ../nscd/nscd.conf /etc/nscd.conf
'../nscd/nscd.conf' -> '/etc/nscd.conf'
root:/sources/glibc-2.25/build# mkdir -pv /var/cache/nscd
mkdir: created directory '/var/cache/nscd'
root:/sources/glibc-2.25/build# █
```

Seuraavaksi asennetaan [lokaalit](#), jotka mahdollistavat järjestelmän käyttämisen eri kielillä. Yhtään lokaalia ei ole aivan välttämätöntä asentaa, mutta jos jotkin puuttuvat, tulevien pakettien testit ohittavat tärkeitä osioita.

Yksittäiset lokaalit voi asentaa käyttämällä `localedef`-ohjelmaa. Esimerkiksi ensimmäinen `localedef`-komento alla yhdistää `/usr/share/i18n/locales/cs_CZ` merkistön lokaalin kuvauksen `/usr/share/i18n/charmaps/UTF-8.gz` merkistökartan kuvauksen kanssa ja lisää tuloksen `/usr/lib/locale/locale-archive` -tiedoston loppuun. Seuraavat komennot asentavat vähimmäismäärän tarvittavia lokaaleja testien suorittamista varten:

```
mkdir -pv /usr/lib/locale
localedef -i cs_CZ -f UTF-8 cs_CZ.UTF-8
localedef -i de_DE -f ISO-8859-1 de_DE
localedef -i de_DE@euro -f ISO-8859-15 de_DE@euro
localedef -i de_DE -f UTF-8 de_DE.UTF-8
localedef -i en_GB -f UTF-8 en_GB.UTF-8
localedef -i en_HK -f ISO-8859-1 en_HK
localedef -i en_PH -f ISO-8859-1 en_PH
localedef -i en_US -f ISO-8859-1 en_US
localedef -i en_US -f UTF-8 en_US.UTF-8
localedef -i es_MX -f ISO-8859-1 es_MX
localedef -i fa_IR -f UTF-8 fa_IR
localedef -i fr_FR -f ISO-8859-1 fr_FR
localedef -i fr_FR@euro -f ISO-8859-15 fr_FR@euro
localedef -i fr_FR -f UTF-8 fr_FR.UTF-8
localedef -i it_IT -f ISO-8859-1 it_IT
localedef -i it_IT -f UTF-8 it_IT.UTF-8
localedef -i ja_JP -f EUC-JP ja_JP
localedef -i ru_RU -f KOI8-R ru_RU.KOI8-R
```

```
localedef -i ru_RU -f UTF-8 ru_RU.UTF-8
localedef -i tr_TR -f UTF-8 tr_TR.UTF-8
localedef -i zh_CN -f GB18030 zh_CN.GB18030
```

Lisäksi asenna lokaalit omaa maatasi, kieltäsi ja merkistöä varten.

Vaihtoehtoisesti voit asentaa kaikki lokaalit, jotka on listattu `glibc-2.25/localedata/SUPPORTED` -tiedostossa. Se sisältää kaikki yllä mainitut lokaalit ja lisäksi monia muita. Halutessasi, aja seuraava komento:

```
make localedata/install-locales
```

```
fi_FI.UTF-8... done
fi_FI.ISO-8859-1... done
fi_FI.ISO-8859-15@euro... done
```

Yllä oleva komento asentaa myös Suomi-lokalisoinnin.

Siinä (epätodennäköisessä) tapauksessa, että tarvitset jotain lokaalia, jota ei ole listattu `glibc-2.25/localedata/SUPPORTED` -tiedostossa, käytä `localedef`-komentoa luodaksesi ja asentaaksesi ne itse.

2.15.3.1 Glibc:n konfiguroiminen

`nsswitch.conf`:n lisääminen:

`/etc/nsswitch.conf` -tiedosto pitää luoda, jotta Glibc tietää, mitä kautta se saa tiettyjä nimipalveluja (name-service) käyttöönsä. Tämä tapahtuu määrittelemällä käytettäväksi tietokannoiksi `files` tyyppi eli paikallinen tiedosto.

Luodaan `/etc/nsswitch.conf` -tiedosto:

```
cat > /etc/nsswitch.conf << "EOF"
# Begin /etc/nsswitch.conf

passwd: files
group: files
shadow: files

hosts: files dns
networks: files

protocols: files
services: files
ethers: files
rpc: files

# End /etc/nsswitch.conf
EOF
```

```
root:/sources/glibc-2.25/build# cat > /etc/nsswitch.conf << "EOF"
> # Begin /etc/nsswitch.conf
>
> passwd: files
> group: files
> shadow: files
>
> hosts: files dns
> networks: files
>
> protocols: files
> services: files
> ethers: files
> rpc: files
>
> # End /etc/nsswitch.conf
> EOF
root:/sources/glibc-2.25/build#
```

Seuraavaksi asennetaan aikavyöhykedata:

```
tar -xf ../../tzdata2016j.tar.gz

ZONEINFO=/usr/share/zoneinfo
mkdir -pv $ZONEINFO/{posix,right}

for tz in etcetera southamerica northamerica europe africa
antarctica \
    asia australasia backward pacificnew systemv; do
    zic -L /dev/null -d $ZONEINFO -y "sh yearis-
type.sh" ${tz}
    zic -L /dev/null -d $ZONEINFO/posix -y "sh yearis-
type.sh" ${tz}
    zic -L leapseconds -d $ZONEINFO/right -y "sh yearis-
type.sh" ${tz}
done

cp -v zone.tab zone1970.tab iso3166.tab $ZONEINFO
zic -d $ZONEINFO -p America/New_York
unset ZONEINFO
```

Puretaan auki:

- `zic -L /dev/null ...` → Tämä luo POSIX -aikavyöhykkeet, ilman karkausseunteja (leap seconds). On tavallista laittaa nämä sekä `zoneinfo:on` että `zoneinfo/posix:iin`. POSIX –aikavyöhykkeiden laittaminen `zoneinfo:on` on tarpeellista, koska muuten monet testit raportoivat virheitä. Sulautetuissa järjestelmissä, joissa tilaa on rajatusti eikä aikavyöhykkeiden päivittämiselle ole ehkä tarvetta, säästäisi 1,9 MB ilman `posix`-hakemiston käyttämistä. Tässä tapauksessa jotkin ohjelmat tai testit saattaisivat kuitenkin tuottaa virheitä.

- `zic -L leapseconds ...` → Luo oikeat aikavyöhykkeet, mukaan lukien karkaussekunnit. Sulautettujen järjestelmien tapauksessa voisi säästää 1,9 MB jättämällä `right`-hakemiston huomioimatta.
- `zic ... -p ...` → Luo `posixrules` -tiedoston. Käytämme New Yorkia, koska POSIX vaatii kesäajan olevan US:n mukainen.

```
root:/sources/glibc-2.25/build# tar -xf ../tzdata2016j.tar.gz
root:/sources/glibc-2.25/build#
root:/sources/glibc-2.25/build# ZONEINFO=/usr/share/zoneinfo
root:/sources/glibc-2.25/build# mkdir -pv $ZONEINFO/{posix,right}
mkdir: created directory '/usr/share/zoneinfo/posix'
mkdir: created directory '/usr/share/zoneinfo/right'
root:/sources/glibc-2.25/build#
<a southamerica northamerica europe africa antarctica \
>   asia australasia backward pacificnew systemv; do
>   zic -L /dev/null -d $ZONEINFO -y "sh yearistype.sh" ${tz}
>   zic -L /dev/null -d $ZONEINFO/posix -y "sh yearistype.sh" ${tz}
>   zic -L leapseconds -d $ZONEINFO/right -y "sh yearistype.sh" ${tz}
> done
root:/sources/glibc-2.25/build#
root:/sources/glibc-2.25/build# cp -v zone.tab zone1970.tab iso3166.tab $ZONEINFO
'zone.tab' -> '/usr/share/zoneinfo/zone.tab'
'zone1970.tab' -> '/usr/share/zoneinfo/zone1970.tab'
'iso3166.tab' -> '/usr/share/zoneinfo/iso3166.tab'
root:/sources/glibc-2.25/build# zic -d $ZONEINFO -p America/New_York
root:/sources/glibc-2.25/build# unset ZONEINFO
root:/sources/glibc-2.25/build#
```

Yksi tapa määrittää paikallinen aikavyöhyke on ajaa seuraava skripti:

```
tzselect
```

Vastattuasi muutamaan kysymykseen, skripti tulostaa aikavyöhykkeen nimen (esim. America/Edmonton). On olemassa myös joitain muitakin mahdollisia aikavyöhykkeitä listattuna `/usr/share/zoneinfo` -tiedostoon, kuten Canada/Eastern tai EST5EDT. Näitä voi myös käyttää, vaikka skripti ei niitä tunnista.

```
root:/sources/glibc-2.25/build# tzselect
Please identify a location so that time zone rules can be set correctly.
Please select a continent, ocean, "coord", or "TZ".
1) Africa
2) Americas
3) Antarctica
4) Asia
5) Atlantic Ocean
6) Australia
7) Europe
8) Indian Ocean
9) Pacific Ocean
10) coord - I want to use geographical coordinates.
11) TZ - I want to specify the time zone using the Posix TZ format.
#?
```

```

Please select a country whose clocks agree with yours.
 1) ?land Islands      18) Greece              35) Norway
 2) Albania            19) Guernsey            36) Poland
 3) Andorra            20) Hungary             37) Portugal
 4) Austria            21) Ireland             38) Romania
 5) Belarus            22) Isle of Man         39) Russia
 6) Belgium            23) Italy                40) San Marino
 7) Bosnia & Herzegovina 24) Jersey              41) Serbia
 8) Britain (UK)       25) Latvia              42) Slovakia
 9) Bulgaria           26) Liechtenstein       43) Slovenia
10) Croatia            27) Lithuania           44) Spain
11) Czech Republic     28) Luxembourg          45) Svalbard & Jan Mayen
12) Denmark            29) Macedonia           46) Sweden
13) Estonia            30) Malta                47) Switzerland
14) Finland            31) Moldova             48) Turkey
15) France             32) Monaco              49) Ukraine
16) Germany            33) Montenegro          50) Vatican City
17) Gibraltar          34) Netherlands
#? █

```

```

The following information has been given:

      Finland

Therefore TZ='Europe/Helsinki' will be used.
Local time is now:      Tue Mar 21 19:15:06 EET 2017.
Universal Time is now: Tue Mar 21 17:15:06 UTC 2017.
Is the above information OK?
1) Yes
2) No
#? █

```

Kello oli oikeasti 21.38, mutta mitä pienistä. Hyväksytään valinta ja jatketaan eteenpäin.

Luodaan /etc/localtime -tiedosto:

```
cp -v /usr/share/zoneinfo/<xxx> /etc/localtime
```

Korvaa <xxx> valitulla aikavyöhykkeellä (esim. Europe/Helsinki).

```

<d# cp -v /usr/share/zoneinfo/Europe/Helsinki /etc/localtime
'/usr/share/zoneinfo/Europe/Helsinki' -> '/etc/localtime'
root:/sources/glibc-2.25/build# █

```

Seuraavaksi konfiguroidaan dynaaminen lataaja. Oletuksena dynaaminen lataaja (/lib/ld-linux.so.2) kahlaa /lib ja /usr/lib -hakemistot läpi etsiessään dynaamisia kirjastoja, joita ohjelmat suorituksessaan tarvitsevat. Jos kirjastoja on muissa hakemistoissa, ne täytyy lisätä /etc/ld.so.conf -tiedostoon, jotta dynaaminen lataaja löytää ne. /usr/local/lib ja /opt/lib ovat sellaisia hakemistoja, joiden myös tiedetään yleensä sisältävän lisäkirjastoja, joten lisätään ne dynaamisen lataajan hakupolulle.

Luodaan /etc/ld.so.conf -tiedosto:

```
cat > /etc/ld.so.conf << "EOF"
# Begin /etc/ld.so.conf
/usr/local/lib
/opt/lib

EOF
```

```
root:/sources/glibc-2.25/build# cat > /etc/ld.so.conf << "EOF"
> # Begin /etc/ld.so.conf
> /usr/local/lib
> /opt/lib
>
> EOF
root:/sources/glibc-2.25/build#
```

Jos halutaan, dynaaminen lataaja voi myös etsiä hakemiston ja sisällyttää sen tiedostot. Yleensä nämä tiedostot sisältävät yhden rivin, joka määrittää halutun kirjastopolun. Lisätään tämä ominaisuus seuraavilla komennoilla:

```
cat >> /etc/ld.so.conf << "EOF"
# Add an include directory
include /etc/ld.so.conf.d/*.conf

EOF

mkdir -pv /etc/ld.so.conf.d
```

```
root:/sources/glibc-2.25/build# cat >> /etc/ld.so.conf << "EOF"
> # Add an include directory
> include /etc/ld.so.conf.d/*.conf
>
> EOF
root:/sources/glibc-2.25/build# mkdir -pv /etc/ld.so.conf.d
mkdir: created directory '/etc/ld.so.conf.d'
root:/sources/glibc-2.25/build# █
```

Lopuksi poistetaan purettu hakemisto.

```
cd /sources/
rm -rf glibc-2.25
```

```
root:/sources/glibc-2.25/build# cd /sources/
root:/sources# rm -rf glibc-2.25
root:/sources# █
```

2.15.4 Työkaluketjun säätäminen

Nyt kun lopulliset C-kirjastot on asennettu, on aika muokata työkaluketjua, jotta se linkkaa uudet käännettyt ohjelmat uusia kirjastoja vastaan.

Ensin varmuuskopioidaan `/tools` -linkkeri ja korvataan se linkkerillä, jonka teimme edellisessä kappaleessa. Luomme myös linkin sen vastakappaleeseen `/tools/$ (uname -m)-pc-linux-gnu/bin:ssä`:

```
mv -v /tools/bin/{ld,ld-old}
mv -v /tools/$(uname -m)-pc-linux-gnu/bin/{ld,ld-old}
mv -v /tools/bin/{ld-new,ld}
ln -sv /tools/bin/ld /tools/$(uname -m)-pc-linux-gnu/bin/ld
```

```
root:/sources# mv -v /tools/bin/{ld,ld-old}
'/tools/bin/ld' -> '/tools/bin/ld-old'
root:/sources# mv -v /tools/$(uname -m)-pc-linux-gnu/bin/{ld,ld-old}
'/tools/x86_64-pc-linux-gnu/bin/ld' -> '/tools/x86_64-pc-linux-gnu/bin/ld-old'
root:/sources# mv -v /tools/bin/{ld-new,ld}
'/tools/bin/ld-new' -> '/tools/bin/ld'
root:/sources# ln -sv /tools/bin/ld /tools/$(uname -m)-pc-linux-gnu/bin/ld
'/tools/x86_64-pc-linux-gnu/bin/ld' -> '/tools/bin/ld'
root:/sources#
```

Seuraavaksi korjataan GCC:n specs-tiedostoja, jotta ne osoittavat uuteen dynaamiseen linkkeriin. Kun poistamme `/tools`:n kaikki instanssit, meille pitäisi jäädä jäljelle pelkästään oikea polku dynaamiseen linkkeriin. Muokataan myös specs-tiedostoa, jotta GCC tietää mistä löytää oikeat header-tiedostot ja Glibc:n start-tiedostot. Tämä kaikki saavutetaan gcc ja sed-komentojen yhdistelmällä:

```
gcc -dumpspecs | sed -e 's@/tools@@g' \
-e '/\*startfile_prefix_spec:{n;s@.*@/usr/lib/ @}' \
-e '/\*cpp:{n;s@$@ -isystem /usr/include@}' > \
`dirname $(gcc --print-libgcc-file-name)`/specs
```

```
root:/sources# gcc -dumpspecs | sed -e 's@/tools@@g' \
> -e '/\*startfile_prefix_spec:{n;s@.*@/usr/lib/ @}' \
> -e '/\*cpp:{n;s@$@ -isystem /usr/include@}' > \
> `dirname $(gcc --print-libgcc-file-name)`/specs
root:/sources#
```

Ei ole pöljempi idea tutkia specs-tiedostoa itse manuaalisesti varmistaakseen, että halutut muutokset oikeasti tehtiin.

On tärkeää varmistaa tässä kohtaa, että muokatun työkaluketjun perustoiminnot (kääntäminen ja linkkaaminen) toimivat, kuten pitäisi. Tehdään seuraavanlainen välitarkistus työkalujen eheyden varmistamiseksi:

```
echo 'int main(){}' > dummy.c
cc dummy.c -v -Wl,--verbose &> dummy.log
readelf -l a.out | grep ': /lib'
```

```
root:/sources# echo 'int main(){}' > dummy.c
root:/sources# cc dummy.c -v -Wl,--verbose &> dummy.log
root:/sources# readelf -l a.out | grep ': /lib'
      [Requesting program interpreter: /lib64/ld-linux-x86-64.so.2]
root:/sources#
```

Virheitä ei pitäisi tulla. Viimeisen komennon pitäisi tulostaa seuraava:

```
[Requesting program interpreter: /lib64/ld-linux-x86-64.so.2]
```

Tai 32-bittisillä järjestelmillä:

```
/lib/ld-linux.so.2
```

Seuraavaksi varmistetaan, että käytämme oikeita start-tiedostoja:

```
grep -o '/usr/lib.*/crt[lin].*succeeded' dummy.log
```

Tulosteen pitäisi olla seuraava:

```
/usr/lib/../../lib/crt1.o succeeded
/usr/lib/../../lib/crti.o succeeded
/usr/lib/../../lib/crtn.o succeeded
```

```
root:/sources# grep -o '/usr/lib.*/crt[lin].*succeeded' dummy.log
/usr/lib/../../lib/crt1.o succeeded
/usr/lib/../../lib/crti.o succeeded
/usr/lib/../../lib/crtn.o succeeded
root:/sources#
```

Varmistetaan, että kääntäjä etsii oikeita header-tiedostoja:

```
grep -B1 '^ /usr/include' dummy.log
```

Tulosteen pitäisi olla seuraava:

```
#include <...> search starts here:  
/usr/include
```

```
root:/sources# grep -B1 '^ /usr/include' dummy.log  
#include <...> search starts here:  
/usr/include  
root:/sources#
```

Seuraavaksi varmistetaan, että uutta linkkeriä käytetään oikeiden hakupolkujen kanssa:

```
grep 'SEARCH.*/usr/lib' dummy.log | sed 's|; |\n|g'
```

Viittaukset polkuihin, joihin sisältyy “-linux-gnu” jätetään huomioimatta, mutta muuten tulosteen pitäisi olla seuraava:

```
SEARCH_DIR("/usr/lib")  
SEARCH_DIR("/lib")
```

```
root:/sources# grep 'SEARCH.*/usr/lib' dummy.log | sed 's|; |\n|g'  
SEARCH_DIR(="/tools/x86_64-pc-linux-gnu/lib64")  
SEARCH_DIR("/usr/lib")  
SEARCH_DIR("/lib")  
SEARCH_DIR(="/tools/x86_64-pc-linux-gnu/lib");  
root:/sources#
```

Varmistetaan, että käytämme oikeaa `libc`:tä.

```
grep "/lib.*/libc.so.6 " dummy.log
```

Tulosteen pitäisi olla:

```
attempt to open /lib/libc.so.6 succeeded
```

```
root:/sources# grep "/lib.*libc.so.6 " dummy.log
attempt to open /lib/libc.so.6 succeeded
root:/sources#
```

Lopuksi varmistetaan, että GCC käyttää oikeaa dynaamista linkkeriä:

```
grep found dummy.log
```

Tulosten pitäisi olla seuraava (riippuen toki hieman mikä alusta on käytössä, 32-bit vai 64-bit):

```
found ld-linux-x86-64.so.2 at /lib/ld-linux-x86-64.so.2
```

```
root:/sources# grep found dummy.log
found ld-linux-x86-64.so.2 at /lib/ld-linux-x86-64.so.2
root:/sources#
```

Jos tulostetta ei ilmesty ollenkaan tai se poikkeaa yllä olevasta, jotakin on pahasti pielessä. Siinä tapauksessa ongelma täytyy paikantaa ja korjata ennen kuin jatkaa eteenpäin. Todennäköisin syy on, että jotakin meni pieleen specs-tiedoston säätämisessä.

Kun kaikki toimii, kuten pitääkin, siivotaan testitiedostot pois:

```
rm -v dummy.c a.out dummy.log
```

```
root:/sources# rm -v dummy.c a.out dummy.log
removed 'dummy.c'
removed 'a.out'
removed 'dummy.log'
root:/sources#
```

2.15.5 Zlib-1.2.11

Käännösaika: alle 0,1 SBU

Asennetut kirjastot: libz.{a,so}

Ensimmäiseksi puretaan zlib-paketti ja siirrytään luotuun hakemistoon:

```
tar xf zlib-1.2.11.tar.xz
cd zlib-1.2.11
```

```
root:/sources# tar xf zlib-1.2.11.tar.xz
```

```
root:/sources# cd zlib-1.2.11
root:/sources/zlib-1.2.11#
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/tools
```

```
root:/sources/zlib-1.2.11# ./configure --prefix=/tools
Checking for gcc...
Checking for shared library support...
Building shared library libz.so.1.2.11 with gcc.
Checking for size_t... Yes.
Checking for off64_t... Yes.
Checking for fseeko... Yes.
Checking for strerror... Yes.
Checking for unistd.h... Yes.
Checking for stdarg.h... Yes.
Checking whether to use vs[n]printf() or s[n]printf()... using vs[n]printf().
Checking for vsnprintf() in stdio.h... Yes.
Checking for return value of vsnprintf()... Yes.
Checking for attribute(visibility) support... Yes.
root:/sources/zlib-1.2.11#
```

Ajetaan make:

```
make
```

```
root:/sources/zlib-1.2.11# make
gcc -O3 -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN -I. -c -o example.o test/example.c
gcc -O3 -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN -c -o adler32.o adler32.c
gcc -O3 -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN -c -o crc32.o crc32.c
gcc -O3 -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN -c -o deflate.o deflate.c
gcc -O3 -D_LARGEFILE64_SOURCE=1 -DHAVE_HIDDEN -c -o infback.o infback.c
```

Testataan:

```
make check
```

```
large_inflate(): OK
after_inflateSync(): hello, hello!
inflate with dictionary: hello, hello!
*** zlib test OK ***
hello world
zlib version 1.2.11 = 0x12b0, compile flags = 0xa9
uncompress(): hello, hello!
gzread(): hello, hello!
gzgets() after gzseek: hello!
inflate(): hello, hello!
large_inflate(): OK
after_inflateSync(): hello, hello!
inflate with dictionary: hello, hello!
*** zlib shared test OK ***
hello world
zlib version 1.2.11 = 0x12b0, compile flags = 0xa9
uncompress(): hello, hello!
gzread(): hello, hello!
gzgets() after gzseek: hello!
inflate(): hello, hello!
large_inflate(): OK
after_inflateSync(): hello, hello!
inflate with dictionary: hello, hello!
*** zlib 64-bit test OK ***
root:/sources/zlib-1.2.11#
```

Asennetaan:

```
make install
```

```

root:/sources/zlib-1.2.11# make install
rm -f /tools/lib/libz.a
cp libz.a /tools/lib
chmod 644 /tools/lib/libz.a
cp libz.so.1.2.11 /tools/lib
chmod 755 /tools/lib/libz.so.1.2.11
rm -f /tools/share/man/man3/zlib.3
cp zlib.3 /tools/share/man/man3
chmod 644 /tools/share/man/man3/zlib.3
rm -f /tools/lib/pkgconfig/zlib.pc
cp zlib.pc /tools/lib/pkgconfig
chmod 644 /tools/lib/pkgconfig/zlib.pc
rm -f /tools/include/zlib.h /tools/include/zconf.h
cp zlib.h zconf.h /tools/include
chmod 644 /tools/include/zlib.h /tools/include/zconf.h
root:/sources/zlib-1.2.11# █

```

Jaettu kirjasto pitää siirtää /lib -hakemistoon ja sen takia .so -tiedosto /usr/lib -hakemistossa pitää luoda uudelleen:

```

mv -v /usr/lib/libz.so.* /lib
ln -sfv ../../lib/${readlink /usr/lib/libz.so}
/usr/lib/libz.so

```

```

root:/sources/zlib-1.2.11# mv -v /usr/lib/libz.so.* /lib
mv: cannot stat '/usr/lib/libz.so.*': No such file or directory
<../../lib/${readlink /usr/lib/libz.so} /usr/lib/libz.so
'/usr/lib/libz.so' -> '../../lib/'
root:/sources/zlib-1.2.11# █

```

Kuten kuvasta näkyy, ylemmässä komennossa on jotain häikkää. libz.so.* -tiedostot löytyvät todennäköisesti /tools/lib -hakemistosta. Ei takerruta pikkuseikkoihin, jatketaan eteenpäin.

Lopuksi poistetaan purettu hakemisto:

```

cd ..
rm -rf zlib-1.2.11

```

```

root:/sources/zlib-1.2.11# cd ..
root:/sources# rm -rf zlib-1.2.11
root:/sources# █

```

2.15.6 File-5.30

Käännösaika: 0,1 SBU

Asennettavat ohjelmat: file

Asennettavat kirjastot: libmagic.so

Ensimmäiseksi puretaan file-paketti ja siirrytään purettuun hakemistoon:

```
tar xf file-5.30.tar.gz
cd file-5.30
```

```
root:/sources# tar xf file-5.30.tar.gz
root:/sources# cd file-5.30
root:/sources/file-5.30#
```

Ajetaan configure:

```
./configure --prefix=/tools
```

```
root:/sources/file-5.30# ./configure --prefix=/tools
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
```

Ajetaan make:

```
make
```

```
root:/sources/file-5.30# make
make all-recursive
make[1]: Entering directory '/sources/file-5.30'
Making all in src
make[2]: Entering directory '/sources/file-5.30/src'
sed -e "s/XY/$(echo 5.30 | tr -d '\n')/" < /src/magic.h.in > magic.h
```

Ajetaan testit:

```
make check
```

```
root:/sources/file-5.30# make check
Making check in src
make[1]: Entering directory '/sources/file-5.30/src'
make check-am
make[2]: Entering directory '/sources/file-5.30/src'
make[2]: Nothing to be done for 'check-am'.
make[2]: Leaving directory '/sources/file-5.30/src'
make[1]: Leaving directory '/sources/file-5.30/src'
```

Asennetaan:

```
make install
```

```
root:/sources/file-5.30# make install
Making install in src
make[1]: Entering directory '/sources/file-5.30/src'
make install-am
make[2]: Entering directory '/sources/file-5.30/src'
make[3]: Entering directory '/sources/file-5.30/src'
./tools/bin/mkdir -p '/tools/lib'
./bin/sh ../libtool --mode=install /tools/bin/install -c libmagic.
libtool: install: /tools/bin/install -c .libs/libmagic.so.1.0.0 /tools
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf file-5.30
```

```
root:/sources/file-5.30# cd ..
root:/sources# rm -rf file-5.30
root:/sources#
```

2.15.7 Binutils-2.27

Käännösaika: 5,7 SBU

Asennettavat ohjelmat: *addr2line, ar, as, c++filt, elfedit, gprof, ld, ld.bfd, nm, objcopy, objdump, ranlib, readelf, size, strings ja strip*

Asennettavat kirjastot: *libbfd.{a,so} ja libopcodes.{a,so}*

Asennettavat hakemistot: */usr/lib/ldscripts*

Testataan, että PTY:t toimivat oikein chroot-ympäristössä:

```
expect -c "spawn ls"
```

Tulosteen pitäisi olla seuraava:

```
spawn ls
```

```
root:/sources# expect -c "spawn ls"
spawn ls
root:/sources# █
```

Mikäli tuloste sen sijaan on alla oleva, ympäristö ei toimi oikein PTY-operaatiota varten. Tämä ongelma pitää ratkaista ennen kuin ajetaan Binutilsin ja GCC:n testejä.

```
The system has no more ptys.
```

```
Ask your system administrator to create more.
```

Puretaan binutils-paketti ja siirrytään purettuun hakemistoon.

```
tar xf binutils-2.27.tar.bz2
cd binutils-2.27
```

```
root:/sources# tar xf binutils-2.27.tar.bz2
root:/sources# cd binutils-2.27
root:/sources/binutils-2.27# █
```

Luodaan build-hakemisto ja siirrytään siihen.

```
mkdir -v build
cd build
```

```
root:/sources/binutils-2.27# mkdir -v build
mkdir: created directory 'build'
root:/sources/binutils-2.27# cd build/
root:/sources/binutils-2.27/build#
```

Valmistellaan Binutils kääntämistä varten:

```
../configure --prefix=/usr \
              --enable-gold \
              --enable-ld=default \
              --enable-plugins \
              --enable-shared \
              --disable-werror \
              --with-system-zlib
```

```
root:/sources/binutils-2.27/build# ../configure --prefix=/usr \
>          --enable-gold \
>          --enable-ld=default \
>          --enable-plugins \
>          --enable-shared \
>          --disable-werror \
>          --with-system-zlib
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking target system type... x86_64-pc-linux-gnu
checking for a BSD-compatible install... /tools/bin/install -c
```

Puretaan auki:

- `--enable-gold` → Kääntää gold-linkkerin ja asentaa sen nimellä `ld.gold`.
- `--enable-ld=default` → Kääntää alkuperäisen bdf-linkkerin ja asentaa sen nimillä `ld` (oletuslinkkeri) ja `ld.bfd`.
- `--enable-plugins` → Sallitaan plugin-tuki linkkerille.

- `--with-system-zlib` → Käytetään jo asennettua zlib-kirjastoa sen sijaan, että käännettäisiin sisällytetty versio.

Ajetaan make:

```
make tooldir=/usr
```

```
/usr/lib/../../lib/libz.so: file not recognized: Is a directory
collect2: error: ld returned 1 exit status
make[4]: *** [Makefile:1289: libbfd.la] Error 1
make[4]: Leaving directory '/sources/binutils-2.27/build/bfd'
make[3]: *** [Makefile:1711: all-recursive] Error 1
make[3]: Leaving directory '/sources/binutils-2.27/build/bfd'
make[2]: *** [Makefile:1187: all] Error 2
make[2]: Leaving directory '/sources/binutils-2.27/build/bfd'
make[1]: *** [Makefile:2726: all-bfd] Error 2
make[1]: Leaving directory '/sources/binutils-2.27/build'
make: *** [Makefile:851: all] Error 2
root:/sources/binutils-2.27/build#
```

Mikäli tulee yllä näkyviä virheitä, ongelma todennäköisesti liittyy zlibiin. Silloin kannattaa ajaa build-hakemistossa `make clean`, ja sen jälkeen ajaa `configure` uudestaan ilman `--with-system-zlib` -asetusta.

```
make[4]: Leaving directory '/sources/binutils-2.27/build/ld'
make[3]: Leaving directory '/sources/binutils-2.27/build/ld'
make[2]: Leaving directory '/sources/binutils-2.27/build/ld'
make[1]: Nothing to be done for 'all-target'.
make[1]: Leaving directory '/sources/binutils-2.27/build'
root:/sources/binutils-2.27/build#
```

Jos taas tuloste on yllä olevan kaltainen, eli lopussa ei tule mitään virheitä, voi huoletta jatkaa eteenpäin.

Ajetaan testit.

HUOM! Binutilsien testien suorittaminen tässä vaiheessa on erittäin tärkeää. Älä ohita niitä missään olosuhteissa.

```
make -k check
```

```
=== ld Summary ===  
# of expected passes          1742  
# of expected failures        57  
# of untested testcases       1  
# of unsupported tests         12  
/sources/binutils-2.27/build/ld/ld-new 2.27
```

```
make[3]: Leaving directory '/sources/binutils-2.27/build/libiberty/testsuite'  
make[2]: Leaving directory '/sources/binutils-2.27/build/libiberty'  
make[1]: Target 'check-host' not remade because of errors.  
make[1]: Nothing to be done for 'check-target'.  
make[1]: Leaving directory '/sources/binutils-2.27/build'  
make: *** [Makefile:2206: do-check] Error 2  
make: Target 'check' not remade because of errors.  
root:/sources/binutils-2.27/build#
```

Kunhan odottamattomia virheitä ei tule, voi jatkaa eteenpäin.

Asennetaan paketti:

```
make tooldir=/usr install
```

```
make[3]: Entering directory '/sources/binutils-2.27/build/libiberty/testsuite'  
make[3]: Nothing to be done for 'install'.  
make[3]: Leaving directory '/sources/binutils-2.27/build/libiberty/testsuite'  
make[2]: Leaving directory '/sources/binutils-2.27/build/libiberty'  
make[1]: Nothing to be done for 'install-target'.  
make[1]: Leaving directory '/sources/binutils-2.27/build'  
root:/sources/binutils-2.27/build#
```

Lopuksi poistetaan purettu hakemisto.

```
cd /sources/  
rm -rf binutils-2.27
```

```
root:/sources/binutils-2.27/build# cd /sources/  
root:/sources# rm -rf binutils-2.27  
root:/sources#
```

2.15.8 GMP-6.1.2

Käännösaika: 1,3 SBU

Asennettavat kirjastot: *libgmp.so* ja *libgmpxx.so*

Asennettava hakemisto: */usr/share/doc/gmp-6.1.2*

HUOM! Jos käännät 32-bit x86 -koneella, mutta sinulla on CPU, joka kykenee ajamaan 64-bittistä koodia ja olet määrittänyt CFLAGS:n ympäristössä, configure-skripti yrittää luoda asetukset 64-bittiselle ympäristölle ja epäonnistuu siinä. Välttääksesi tämän aja configure seuraavasti:

```
ABI=32 ./configure ...
```

Ensimmäiseksi puretaan gmp-paketti:

```
tar xf gmp-6.1.2.tar.xz
cd gmp-6.1.2
```

```
root:/sources# tar xf gmp-6.1.2.tar.xz
root:/sources# cd gmp-6.1.2
root:/sources/gmp-6.1.2#
```

Valmistellaan GMP kääntämistä varten:

```
./configure --prefix=/usr \
            --enable-cxx \
            --disable-static \
            --docdir=/usr/share/doc/gmp-6.1.2
```

```
configure: summary of build options:

Version:          GNU MP 6.1.2
Host type:        sandybridge-pc-linux-gnu
ABI:              64
Install prefix:   /usr
Compiler:         gcc
Static libraries: no
Shared libraries: yes

root:/sources/gmp-6.1.2#
```

Puretaan auki:

- `--enable-cxx` → Ottaa käyttöön C++ -tuen.
- `--docdir=/usr/share/doc/gmp-6.1.2` → Määrittää oikean paikan dokumentaatiolle.

Käännetään paketti ja luodaan HTML-dokumentaatio:

```
make
make html
```

```
libtool: link: ( cd ".libs" && rm -f "libgmpxx.la" && ln -s "../libgmpxx.la" )
make[2]: Leaving directory '/sources/gmp-6.1.2'
make[1]: Leaving directory '/sources/gmp-6.1.2'
root:/sources/gmp-6.1.2# clear

root:/sources/gmp-6.1.2# make html
Making html in tests
make[1]: Entering directory '/sources/gmp-6.1.2/tests'
Making html in
```

Ajetaan testit.

HUOM! GMP:n testien ajaminen tässä vaiheessa on erittäin tärkeää. *Älä ohita* niitä missään olosuhteissa.

```
make check 2>&1 | tee gmp-check-log
```

GMP:n koodi on optimoitu sitä prosessoria varten, jota käytetään kääntämiseen. Toisinaan koodi, joka tunnistaa prosessorin, ymmärtää väärin järjestelmän kapasiteetin ja testeissä tulee virheviesti: "Illegal instruction". Tässä tapauksessa, GMP pitää konfiguroida uudelleen `--build=x86_64-unknown-linux-gnu -asetuk-` sen kanssa ja kääntää sen jälkeen uudelleen.

Varmista, että kaikki 190 testiä menivät läpi. Tarkastaaksesi tulokset aja seuraava komento:

```
awk '/# PASS:/{total+=$3} ; END{print total}' gmp-check-log
```

```
<'/# PASS:/{total+=$3} ; END{print total}' gmp-check-log
190
root:/sources/gmp-6.1.2#
```

Asennetaan paketti ja sen dokumentaatio:

```
make install
make install-html
```

```
root:/sources/gmp-6.1.2# make install
make install-recursive
make[1]: Entering directory '/sources/gmp-6.1.2'
Making install in tests
make[2]: Entering directory '/sources/gmp-6.1.2/tests'
Making install in .
make[3]: Entering directory '/sources/gmp-6.1.2/tests'
```

```
root:/sources/gmp-6.1.2# make install-html
Making install-html in tests
make[1]: Entering directory '/sources/gmp-6.1.2/tests'
Making install-html in .
make[2]: Entering directory '/sources/gmp-6.1.2/tests'
make[2]: Nothing to be done for 'install-html-am'
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf gmp-6.1.2
```

```
root:/sources/gmp-6.1.2# cd ..  
root:/sources# rm -rf gmp-6.1.2  
root:/sources# █
```

2.15.9 MPFR-3.1.5

Käännösaika: 0,8 SBU

Asennettavat kirjastot: libmpfr.so

Asennettava hakemisto: /usr/share/doc/mpfr-3.1.5

Ensimmäiseksi puretaan mpfr-paketti:

```
tar xf mpfr-3.1.5.tar.xz
cd mpfr-3.1.5
```

```
root:/sources# tar xf mpfr-3.1.5.tar.xz
root:/sources# cd mpfr-3.1.5
root:/sources/mpfr-3.1.5#
```

Ajetaan configure:

```
./configure --prefix=/usr \
            --disable-static \
            --enable-thread-safe \
            --docdir=/usr/share/doc/mpfr-3.1.5
```

```
root:/sources/mpfr-3.1.5# ./configure --prefix=/usr \
> --disable-static \
> --enable-thread-safe \
> --docdir=/usr/share/doc/mpfr-3.1.5
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
```

Käännetään paketti ja luodaan HTML-dokumentaatio:

```
make
make html
```

```
root:/sources/mpfr-3.1.5# make html
Making html in doc
make[1]: Entering directory '/sources/mpfr-3.1.5/doc'
rm -rf mpfr.htm
if /bin/sh /sources/mpfr-3.1.5/missing makeinfo --html --enable-encoding -I \
```

Ajetaan testit.

HUOM! MPFR:n testien ajaminen tässä vaiheessa on erittäin tärkeää. *Älä ohita* niitä missään olosuhteissa.

```
make check
```

```
=====
Testsuite summary for MPFR 3.1.5
=====
# TOTAL: 160
# PASS: 159
# SKIP: 1
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan paketti ja sen dokumentaatio:

```
make install
make install-html
```

```
root:/sources/mpfr-3.1.5# make install
Making install in doc
make[1]: Entering directory '/sources/mpfr-3.1.5/doc'
make[2]: Entering directory '/sources/mpfr-3.1.5/doc'
make[2]: Nothing to be done for 'install-exec-am'.
../tools/bin/mkdir -p '/usr/share/doc/mpfr-3.1.5'
```

```
root:/sources/mpfr-3.1.5# make install-html
Making install-html in doc
make[1]: Entering directory '/sources/mpfr-3.1.5/doc'
../tools/bin/mkdir -p '/usr/share/doc/mpfr-3.1.5'
../tools/bin/mkdir -p '/usr/share/doc/mpfr-3.1.5/mpfr.html'
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf mpfr-3.1.5
```

```
root:/sources/mpfr-3.1.5# cd ..  
root:/sources# rm -rf mpfr-3.1.5  
root:/sources# █
```

2.15.10 MPC-1.0.3

Käännösaika: 0,3 SBU

Asennettavat kirjastot: libmpc.so

Asennettava hakemisto: /usr/share/doc/mpc-1.0.3

Ensimmäiseksi puretaan mpc:

```
tar xf mpc-1.0.3.tar.gz
cd mpc-1.0.3
```

```
root:/sources# tar xf mpc-1.0.3.tar.gz
root:/sources# cd mpc-1.0.3
root:/sources/mpc-1.0.3#
```

Ajetaan configure:

```
./configure --prefix=/usr \
            --disable-static \
            --docdir=/usr/share/doc/mpc-1.0.3
```

```
root:/sources/mpc-1.0.3# ./configure --prefix=/usr \
> --disable-static \
> --docdir=/usr/share/doc/mpc-1.0.3
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
```

Käännetään paketti ja luodaan HTML-dokumentaatio:

```
make
make html
```

```
root:/sources/mpc-1.0.3# make
make all-recursive
make[1]: Entering directory '/sources/mpc-1.0.3'
Making all in src
make[2]: Entering directory '/sources/mpc-1.0.3/src'
/bin/sh ../libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I
```

```
root:/sources/mpc-1.0.3# make html
Making html in src
make[1]: Entering directory '/sources/mpc-1.0.3/src'
make[1]: Nothing to be done for 'html'.
make[1]: Leaving directory '/sources/mpc-1.0.3/src'
```

Ajetaan testit:

```
make check
```

```
=====
Testsuite summary for mpc 1.0.3
=====
```

```
# TOTAL: 64
# PASS: 64
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan paketti ja sen dokumentaatio:

```
make install
make install-html
```

```
root:/sources/mpc-1.0.3# make install
Making install in src
make[1]: Entering directory '/sources/mpc-1.0.3/src'
make[2]: Entering directory '/sources/mpc-1.0.3/src'
  /tools/bin/mkdir -p '/usr/lib'
  /bin/sh ../libtool --mode=install /tools/bin/install -c libmpc
```

```
root:/sources/mpc-1.0.3# make install-html
Making install-html in src
make[1]: Entering directory '/sources/mpc-1.0.3/src'
make[1]: Nothing to be done for 'install-html'.
make[1]: Leaving directory '/sources/mpc-1.0.3/src'
Making install-html in tests
make[1]: Entering directory '/sources/mpc-1.0.3/tests'
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf mpc-1.0.3
```

```
root:/sources/mpc-1.0.3# cd ..  
root:/sources# rm -rf mpc-1.0.3  
root:/sources# █
```

2.15.11 GCC-6.3.0

Arvioitu käännösaika: 82 SBU

Asennettavat ohjelmat: *c++, cc (linkki gcc:hen), cpp, g++, gcc, gcc-ar, gcc-nm, gcc-ranlib ja gcov*

Asennettavat kirjastot: *libasan.{a,so}, libatomic.{a,so}, libgcc.a, libgcc_eh.a, libgcc_s.so, libgcov.a, libgomp.{a,so}, libiberty.a, libitm.{a,so}, liblto_plugin.so, libquadmath.{a,so}, libssp.{a,so}, libssp_nonshared.a, libstdc++.a, libsupc++.a ja libtsan.{a,so}*

Asennettavat hakemistot: */usr/include/c++, /usr/lib/gcc, /usr/libexec/gcc ja /usr/share/gcc-6.3.0*

Ensimmäiseksi puretaan GCC ja siirrytään purettuun hakemistoon:

```
tar xf gcc-6.3.0.tar.bz2
cd gcc-6.3.0
```

```
root:/sources# tar xf gcc-6.3.0.tar.bz2
root:/sources# cd gcc
root:/sources# cd gcc-6.3.0
root:/sources/gcc-6.3.0#
```

Mikäli käännetään x86_64 -koneella, muutetaan oletushakemiston nimi 64-bittisille kirjastoille "lib":ksi.

```
case $(uname -m) in
  x86_64)
    sed -e '/m64=/s/lib64/lib/' \
        -i.orig gcc/config/i386/t-linux64
    ;;
esac
```

```
root:/sources/gcc-6.3.0# case $(uname -m) in
>   x86_64)
>     sed -e '/m64=/s/lib64/lib/' \
>         -i.orig gcc/config/i386/t-linux64
>     ;;
> esac
root:/sources/gcc-6.3.0#
```

Luodaan build-hakemisto ja siirrytään siihen.

```
mkdir -v build
cd build
```

```
root:/sources/gcc-6.3.0# mkdir -v build
mkdir: created directory 'build'
root:/sources/gcc-6.3.0# cd build/
root:/sources/gcc-6.3.0/build#
```

Ajetaan configure.

HUOM! Jos sinulla oli viimeksi Binutilsin ja zlibin kanssa vaikeuksia, ota viimeinen `--with-system-zlib` -asetus pois configuresta.

```
SED=sed \
../configure --prefix=/usr \
              --enable-languages=c,c++ \
              --disable-multilib \
              --disable-bootstrap \
              --with-system-zlib
```

```
root:/sources/gcc-6.3.0/build# SED=sed \
> ../configure --prefix=/usr \
>               --enable-languages=c,c++ \
>               --disable-multilib \
>               --disable-bootstrap \
>               --with-system-zlib \
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking target system type... x86_64-pc-linux-gnu
checking for a BSD-compatible install... /tools/bin/install -c
```

Puretaan auki:

- `SED=sed` → Estää kovakoodatun polun `/tools/bin/sed:iin`.
- `--with-system-zlib` → Käskee GCC:tä linkkaamaan järjestelmään asennettuun kopioon zlib-kirjastosta sen sijaan, että linkkaisi omaan sisäiseen kopioonsa.

Ajetaan make.

```
make
```

```
true D0=all multi-do # make
make[4]: Leaving directory '/sources/gcc-6.3.0/build/x86_64-pc-linux-gnu/libatomic'
make[3]: Leaving directory '/sources/gcc-6.3.0/build/x86_64-pc-linux-gnu/libatomic'
make[2]: Leaving directory '/sources/gcc-6.3.0/build/x86_64-pc-linux-gnu/libatomic'
make[1]: Leaving directory '/sources/gcc-6.3.0/build'
root:/sources/gcc-6.3.0/build#
```

Yhden testisetin tiedetään uuvuttavan stackin, joten nostetaan stackin kokoa ennen kuin ajetaan testit:

```
root:/sources/gcc-6.3.0/build# ulimit -s 32768
root:/sources/gcc-6.3.0/build#
```

Ajetaan testit.

```
make -k check
```

```
make[1]: Leaving directory '/sources/gcc-6.3.0/build'
make: *** [Makefile:2243: do-check] Error 2
make: Target 'check' not remade because of errors.
root:/sources/gcc-6.3.0/build#
```

Saadaksesi yhteenvedon testien tuloksista, aja seuraava:

```
../contrib/test_summary
```

```
==== libstdc++ tests ====

Running target unix
FAIL: experimental/filesystem/iterators/directory_iterator.cc execution test
FAIL: experimental/filesystem/iterators/recursive_directory_iterator.cc execution test
FAIL: experimental/filesystem/operations/exists.cc execution test
FAIL: experimental/filesystem/operations/is_empty.cc execution test
FAIL: experimental/filesystem/operations/temp_directory_path.cc execution test

==== libstdc++ Summary ====

# of expected passes          11108
# of unexpected failures       5
# of expected failures        66
# of unsupported tests        235
```

```
=== gcc tests ===

Running target unix
FAIL: gcc.dg/cpp/trad/include.c (test for excess errors)

=== gcc Summary ===
# of expected passes      117920
# of unexpected failures   1
# of expected failures    316
# of unsupported tests     1703
/sources/gcc-6.3.0/build/gcc/xgcc version 6.3.0 (GCC)
```

Tuloksia voi verrata seuraaviin:

<http://www.linuxfromscratch.org/lfs/build-logs/8.0/>

<http://gcc.gnu.org/ml/gcc-testresults/>

Muutamia odottamattomia virheitä ei aina voi välttää. GCC:n kehittäjät ovat yleensä tietoisia näistä virheistä, mutta eivät ole vielä ratkaisseet niitä. Esimerkiksi viiden libstdc++ -testin tiedetään epäonnistuvan, kun ne ajetaan pääkäyttäjänä, kuten me teemme. Mikäli testitulokset eivät suunnattoman paljon eroa yllä olevista linkeistä, on turvallista jatkaa.

Asennetaan paketti:

```
make install
```

Luodaan [FHS](#):n historiallisista syistä vaatima symbolinen linkki:

```
ln -sv ../usr/bin/cpp /lib
```

```
root:/sources/gcc-6.3.0/build# ln -sv ../usr/bin/cpp /lib
'/lib/cpp' -> '../usr/bin/cpp'
root:/sources/gcc-6.3.0/build#
```

Monet paketit käyttävät C -kääntäjän nimenä `cc`:tä. Tyydyttääksemme nämä paketit, luodaan symbolinen linkki:

```
ln -sv gcc /usr/bin/cc
```

```
root:/sources/gcc-6.3.0/build# ln -sv gcc /usr/bin/cc
'/usr/bin/cc' -> 'gcc'
root:/sources/gcc-6.3.0/build#
```

Luodaan symbolinen linkki salliaksemme ohjelmien kääntämisen käyttäen LTO:ta (Link Time Optimization).

```
install -v -dm755 /usr/lib/bfd-plugins
ln -sfv ../../libexec/gcc/$(gcc -dump-
machine)/6.3.0/liblto_plugin.so \
    /usr/lib/bfd-plugins/
```

```
root:/sources/gcc-6.3.0/build# install -v -dm755 /usr/lib/bfd-plugins
install: creating directory '/usr/lib/bfd-plugins'
<libexec/gcc/$(gcc -dumpmachine)/6.3.0/liblto_plugin.so \
>    /usr/lib/bfd-plugins/
'/usr/lib/bfd-plugins/liblto_plugin.so' -> ' ../../libexec/gcc/x86_64-pc-linux-gnu/6.3.
0/liblto_plugin.so'
root:/sources/gcc-6.3.0/build#
```

Nyt, kun viimeinen työkaluketjumme on valmis, on tärkeää varmistaa, että kääntäminen ja linkkaaminen toimivat kuten pitääkin. Tehdään samat eheystestit kuin aiemmin tässä kappaleessa:

```
echo 'int main(){}' > dummy.c
cc dummy.c -v -Wl,--verbose &> dummy.log
readelf -l a.out | grep ': /lib'
```

Virheitä ei pitäisi tulla ja viimeisen komennon tulosteen pitäisi olla seuraava (riipuen hieman alustasta):

```
[Requesting program interpreter: /lib64/ld-linux-x86-64.so.2]
```

```
root:/sources/gcc-6.3.0/build# echo 'int main(){}' > dummy.c
root:/sources/gcc-6.3.0/build# cc dummy.c -v -Wl,--verbose &> dummy.log
root:/sources/gcc-6.3.0/build# readelf -l a.out | grep ': /lib'
[Requesting program interpreter: /lib64/ld-linux-x86-64.so.2]
root:/sources/gcc-6.3.0/build#
```

Varmistetaan, että käytämme oikeita start-tiedostoja:

```
grep -o '/usr/lib.*/crt[lin].*succeeded' dummy.log
```

Viimeisen komennon tulosteen pitäisi olla:

```
/usr/lib/gcc/x86_64-pc-linux-gnu/6.3.0/../../../../lib/crt1.o  
succeeded  
  
/usr/lib/gcc/x86_64-pc-linux-gnu/6.3.0/../../../../lib/crti.o  
succeeded  
  
/usr/lib/gcc/x86_64-pc-linux-gnu/6.3.0/../../../../lib/crtn.o  
succeeded
```

Riippuen koneen arkkitehtuurista, yllä oleva tuloste voi vaihdella hieman. Tärkeää on kuitenkin varmistaa, että gcc on löytänyt kaikki kolme `crt*.o` -tiedostoa `/usr/lib` -hakemistosta.

```
root:/sources/gcc-6.3.0/build# grep -o '/usr/lib.*/crt[lin].*succeeded' dummy.log  
/usr/lib/gcc/x86_64-pc-linux-gnu/6.3.0/../../../../lib/crt1.o succeeded  
/usr/lib/gcc/x86_64-pc-linux-gnu/6.3.0/../../../../lib/crti.o succeeded  
/usr/lib/gcc/x86_64-pc-linux-gnu/6.3.0/../../../../lib/crtn.o succeeded  
root:/sources/gcc-6.3.0/build#
```

Varmistetaan, että kääntäjä etsii oikeita header-tiedostoja:

```
grep -B4 '^ /usr/include' dummy.log
```

Tulosteen pitäisi olla seuraava:

```
#include <...> search starts here:  
  
/usr/lib/gcc/x86_64-pc-linux-gnu/6.3.0/include  
  
/usr/local/include  
  
/usr/lib/gcc/x86_64-pc-linux-gnu/6.3.0/include-fixed  
  
/usr/include
```

Jälleen tuloste voi hieman vaihdella arkkitehtuurin mukaan.

```
root:/sources/gcc-6.3.0/build# grep -B4 '^ /usr/include' dummy.log
#include <...> search starts here:
 /usr/lib/gcc/x86_64-pc-linux-gnu/6.3.0/include
 /usr/local/include
 /usr/lib/gcc/x86_64-pc-linux-gnu/6.3.0/include-fixed
 /usr/include
root:/sources/gcc-6.3.0/build#
```

Varmistetaan, että uutta linkkeriä käytetään oikeiden hakupolkujen kanssa:

```
grep 'SEARCH.*/usr/lib' dummy.log |sed 's|; |\n|g'
```

Komennon pitäisi tulostaa seuraavaa:

```
SEARCH_DIR("/usr/x86_64-pc-linux-gnu/lib64")
SEARCH_DIR("/usr/local/lib64")
SEARCH_DIR("/lib64")
SEARCH_DIR("/usr/lib64")
SEARCH_DIR("/usr/x86_64-pc-linux-gnu/lib")
SEARCH_DIR("/usr/local/lib")
SEARCH_DIR("/lib")
SEARCH_DIR("/usr/lib");
```

32-bittinen järjestelmä saattaa näyttää muutamia eri hakemistoja. Alla esimerkki i686-koneen tulosteesta:

```
SEARCH_DIR("/usr/i686-pc-linux-gnu/lib32")
SEARCH_DIR("/usr/local/lib32")
SEARCH_DIR("/lib32")
SEARCH_DIR("/usr/lib32")
SEARCH_DIR("/usr/i686-pc-linux-gnu/lib")
SEARCH_DIR("/usr/local/lib")
SEARCH_DIR("/lib")
SEARCH_DIR("/usr/lib");
```

```
root:/sources/gcc-6.3.0/build# grep 'SEARCH.*/usr/lib' dummy.log |sed 's|; |\n|g'
SEARCH_DIR("/usr/x86_64-pc-linux-gnu/lib64")
SEARCH_DIR("/usr/local/lib64")
SEARCH_DIR("/lib64")
SEARCH_DIR("/usr/lib64")
SEARCH_DIR("/usr/x86_64-pc-linux-gnu/lib")
SEARCH_DIR("/usr/local/lib")
SEARCH_DIR("/lib")
SEARCH_DIR("/usr/lib");
root:/sources/gcc-6.3.0/build#
```

Varmistetaan, että käytämme oikeaa libc:tä:

```
grep "/lib.*/libc.so.6 " dummy.log
```

Viimeisen komennon pitäisi tulostaa seuraavaa:

```
attempt to open /lib/libc.so.6 succeeded
```

```
root:/sources/gcc-6.3.0/build# grep "/lib.*/libc.so.6 " dummy.log
attempt to open /lib/libc.so.6 succeeded
root:/sources/gcc-6.3.0/build#
```

Varmistetaan, että GCC käyttää oikeaa dynaamista linkkeriä:

```
grep found dummy.log
```

Tulosteen pitäisi olla kutakuinkin seuraavanlaista (riippun hieman alustasta):

```
found ld-linux-x86-64.so.2 at /lib/ld-linux-x86-64.so.2
```

Mikäli tuloste ei näy ollenkaan tai poikkeaa yllä olevasta, jotakin on pahasti pielessä. Ongelma pitää jäljittää ja ratkaista ennen kuin voi jatkaa. Todennäköisin syy on, että jotakin meni pieleen specs-tiedoston säätämisessä.

```
root:/sources/gcc-6.3.0/build# grep found dummy.log
found ld-linux-x86-64.so.2 at /lib/ld-linux-x86-64.so.2
root:/sources/gcc-6.3.0/build#
```

Jos kaikki toimii niin kuin pitääkin, siivotaan pois testitiedostot:

```
rm -v dummy.c a.out dummy.log
```

```
root:/sources/gcc-6.3.0/build# rm -v dummy.c a.out dummy.log
removed 'dummy.c'
removed 'a.out'
removed 'dummy.log'
root:/sources/gcc-6.3.0/build#
```

Siirretään väärään paikkaan asetettu tiedosto:

```
mkdir -pv /usr/share/gdb/auto-load/usr/lib
mv -v /usr/lib/*gdb.py /usr/share/gdb/auto-load/usr/lib
```

```
root:/sources/gcc-6.3.0/build# mkdir -pv /usr/share/gdb/auto-load/usr/lib
mkdir: created directory '/usr/share/gdb'
mkdir: created directory '/usr/share/gdb/auto-load'
mkdir: created directory '/usr/share/gdb/auto-load/usr'
mkdir: created directory '/usr/share/gdb/auto-load/usr/lib'
<mv -v /usr/lib/*gdb.py /usr/share/gdb/auto-load/usr/lib
'/usr/lib/libstdc++.so.6.0.22-gdb.py' -> '/usr/share/gdb/auto-load/usr/lib/libstdc++.s
o.6.0.22-gdb.py'
root:/sources/gcc-6.3.0/build#
```

Lopuksi poistetaan purettu hakemisto:

```
cd /sources/
rm -rf gcc-6.3.0
```

```
root:/sources/gcc-6.3.0/build# cd /sources/
root:/sources# rm -rf gcc-6.3.0
root:/sources#
```

2.15.12 Bzip2-1.0.6

Arvioitu käännösaika: alle 0,1 SBU

Asennettavat ohjelmat: *bunzip2* (linkki *bzip2:een*), *bzcat* (linkki *bzip2:een*), *bzcmp* (linkki *bzdiff:iin*), *bzdiff*, *bzegrep* (linkki *bzgrep:iin*), *bzfgrep* (linkki *bzgrep:iin*), *bzgrep*, *bzip2*, *bzip2recover*, *bzless* (linkki *bzmore:en*) ja *bzmore*

Asennettavat kirjastot: *libbz2.{a,so}*

Asennettava hakemisto: */usr/share/doc/bzip2-1.0.6*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf bzip2-1.0.6.tar.gz
cd bzip2-1.0.6
```

```
root:/sources# tar xf bzip2-1.0.6.tar.gz
root:/sources# cd bzip2-1.0.6
root:/sources/bzip2-1.0.6#
```

Seuraavalla komennolla varmistetaan, että symbolisten linkkien asennukset ovat relative:

```
sed -i 's@(\ln -s -f \)$(PREFIX)/bin/@\1@' Makefile
```

```
root:/sources/bzip2-1.0.6# sed -i 's@(\ln -s -f \)$(PREFIX)/bin/@\1@' Makefile
root:/sources/bzip2-1.0.6#
```

Varmistetaan, että man-sivut asennetaan oikeaan paikkaan:

```
sed -i "s@(PREFIX)/man@(PREFIX)/share/man@g" Makefile
```

```
root:/sources/bzip2-1.0.6# sed -i "s@(PREFIX)/man@(PREFIX)/share/man@g" Makefile
root:/sources/bzip2-1.0.6#
```

Valmistellaan Bzip2 kääntämistä varten:

```
make -f Makefile-libbz2_so
make clean
```

```

root:/sources/bzip2-1.0.6# make -f Makefile-libbz2_so
gcc -fpic -fPIC -Wall -Winline -O2 -g -D_FILE_OFFSET_BITS=64 -c blocksort.c
gcc -fpic -fPIC -Wall -Winline -O2 -g -D_FILE_OFFSET_BITS=64 -c huffman.c
gcc -fpic -fPIC -Wall -Winline -O2 -g -D_FILE_OFFSET_BITS=64 -c crctable.c
gcc -fpic -fPIC -Wall -Winline -O2 -g -D_FILE_OFFSET_BITS=64 -c randtable.c
gcc -fpic -fPIC -Wall -Winline -O2 -g -D_FILE_OFFSET_BITS=64 -c compress.c
gcc -fpic -fPIC -Wall -Winline -O2 -g -D_FILE_OFFSET_BITS=64 -c decompress.c
gcc -fpic -fPIC -Wall -Winline -O2 -g -D_FILE_OFFSET_BITS=64 -c bzlib.c
gcc -shared -Wl,-soname -Wl,libbz2.so.1.0 -o libbz2.so.1.0.6 blocksort.o huffman.o crctable.o randtable.o compress.o decompress.o bzlib.o
gcc -fpic -fPIC -Wall -Winline -O2 -g -D_FILE_OFFSET_BITS=64 -o bzip2-shared bzip2.c libbz2.so.1.0.6
bzip2.c: In function 'testStream':
bzip2.c:557:37: warning: variable 'nread' set but not used [-Wunused-but-set-variable]
    Int32  bzerr, bzerr_dummy, ret, nread, streamNo, i;
                                   ^~~~~
rm -f libbz2.so.1.0
ln -s libbz2.so.1.0.6 libbz2.so.1.0
root:/sources/bzip2-1.0.6#

```

```

root:/sources/bzip2-1.0.6# make clean
rm -f *.o libbz2.a bzip2 bzip2recover \
sample1.rb2 sample2.rb2 sample3.rb2 \
sample1.tst sample2.tst sample3.tst
root:/sources/bzip2-1.0.6#

```

Puretaan auki:

- `-f Makefile-libbz2_so` → Aiheuttaa sen, että Bzip2 käännetään käyttäen eri Makefile -tiedostoa, tässä tapauksessa `Makefile-libbz2_so` -tiedostoa, joka luo dynaamisen `libbz2.so` -kirjaston ja linkkaa Bzip2:n työkalut sitä vasten.

Käännetään ja testataan paketti:

```
make
```

```

root:/sources/bzip2-1.0.6# make

If compilation produces errors, or a large number of warnings,
please read README.COMPIRATION.PROBLEMS -- you might be able to
adjust the flags in this Makefile to improve matters.

Also in README.COMPIRATION.PROBLEMS are some hints that may help
if your build produces an executable which is unable to correctly

```

Asennetaan ohjelma:

```
make PREFIX=/usr install
```

```
root:/sources/bzip2-1.0.6# make PREFIX=/usr install
```

If compilation produces errors, or a large number of warnings, please read README.COMPIRATION.PROBLEMS -- you might be able to adjust the flags in this Makefile to improve matters.

Also in README.COMPIRATION.PROBLEMS are some hints that may help if your build produces an executable which is unable to correctly

Asennetaan jaettu bzip2-binääri /bin -hakemistoon, tehdään joitakin tarpeellisia symbolisia linkkejä ja siivotetaan:

```
cp -v bzip2-shared /bin/bzip2
cp -av libbz2.so* /lib
ln -sv ../../lib/libbz2.so.1.0 /usr/lib/libbz2.so
rm -v /usr/bin/{bunzip2,bzcat,bzip2}
ln -sv bzip2 /bin/bunzip2
ln -sv bzip2 /bin/bzcat
```

```
root:/sources/bzip2-1.0.6# cp -v bzip2-shared /bin/bzip2
'bzip2-shared' -> '/bin/bzip2'
root:/sources/bzip2-1.0.6# cp -av libbz2.so* /lib
'libbz2.so.1.0' -> '/lib/libbz2.so.1.0'
root:/sources/bzip2-1.0.6# ln -sv ../../lib/libbz2.so.1.0 /usr/lib/libbz2.so
'/usr/lib/libbz2.so' -> '../../lib/libbz2.so.1.0'
root:/sources/bzip2-1.0.6# rm -v /usr/bin/{bunzip2,bzcat,bzip2}
removed '/usr/bin/bunzip2'
removed '/usr/bin/bzcat'
removed '/usr/bin/bzip2'
root:/sources/bzip2-1.0.6# ln -sv bzip2 /bin/bunzip2
'/bin/bunzip2' -> 'bzip2'
root:/sources/bzip2-1.0.6# ln -sv bzip2 /bin/bzcat
'/bin/bzcat' -> 'bzip2'
root:/sources/bzip2-1.0.6# █
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf bzip2-1.0.6
```

```
root:/sources/bzip2-1.0.6# cd ..
root:/sources# rm -rf bzip2-1.0.6
root:/sources# █
```

2.15.13 Pkg-config-0.29.1

Arvioitu käännösaika: 0,4 SBU

Asennettava ohjelma: pkg-config

Asennettava hakemisto: /usr/share/doc/pkg-config-0.29.1

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf pkg-config-0.29.1.tar.gz
cd pkg-config-0.29.1
```

```
root:/sources# tar xf pkg-config-0.29.1.tar.gz
root:/sources# cd pkg-config-0.29.1
root:/sources/pkg-config-0.29.1#
```

Ajetaan configure:

```
./configure --prefix=/usr \
            --with-internal-glib \
            --disable-compile-warnings \
            --disable-host-tool \
            --docdir=/usr/share/doc/pkg-config-0.29.1
```

Puretaan auki:

- `--with-internal-glib` → Tämä sallii pkg-configin käyttää sen omaa sisäistä versiota Glib:stä, koska ulkoinen versio ei ole saatavilla LFS:ssä.
- `--disable-compile-warnings` → Estää käännösjärjestelmää käyttämästä käännöslippuja, koska se aiheuttaisi käännösvirheen käännettäessä GCC kutosta.
- `--disable-host-tool` → Estää luomasta ei-toivottua kovaa linkkiä pkg-config -ohjelmaan.

```
root:/sources/pkg-config-0.29.1# ./configure --prefix=/usr \
> --with-internal-glib \
> --disable-compile-warnings \
> --disable-host-tool \
> --docdir=/usr/share/doc/pkg-config-0.29.1
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
```

Käännetään paketti.

```
make
```

```
root:/sources/pkg-config-0.29.1# make
make all-recursive
make[1]: Entering directory '/sources/pkg-config-0.29.1'
Making all in glib
make[2]: Entering directory '/sources/pkg-config-0.29.1/glib'
make all-recursive
make[3]: Entering directory '/sources/pkg-config-0.29.1/glib'
```

Ajetaan testit:

```
make check
```

```
=====
All 28 tests passed
=====
```

Asennetaan paketti:

```
make install
```

```
root:/sources/pkg-config-0.29.1# make install
Making install in glib
make[1]: Entering directory '/sources/pkg-config-0.29.1/glib'
make install-recursive
make[2]: Entering directory '/sources/pkg-config-0.29.1/glib'
Making install in .
make[3]: Entering directory '/sources/pkg-config-0.29.1/glib'
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf pkg-config-0.29.1
```

```
root:/sources/pkg-config-0.29.1# cd ..  
root:/sources# rm -rf pkg-config-0.29.1  
root:/sources# █
```

2.15.14 Ncurses-6.0

Käännösaika: 0,4 SBU

Asennettavat ohjelmat: *captainfo* (linkki tic:iin), *clear*, *infocmp*, *infotocap* (link tic:iin), *ncursesw6-config*, *reset* (linkki tset:iin), *tabs*, *tic*, *toe*, *tput* ja *tset*

Asennettavat kirjastot: *libcursesw.so* (symbolinen linkki ja linkkeriskripti *libncursesw.so:hon*), *libformw.so*, *libmenuw.so*, *libncursesw.so*, *libncurses++w.a*, *libpanelw.so* ja niiden non-wide-character vastineet ilman "w"-kirjainta kirjastonimissä.

Asennettavat hakemistot: */usr/share/tabset*, */usr/share/terminfo* ja */usr/share/doc/ncurses-6.0*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf ncurses-6.0.tar.gz
cd ncurses-6.0
```

```
root:/sources# tar xf ncurses-6.0.tar.gz
root:/sources# cd ncurses-6.0
root:/sources/ncurses-6.0#
```

Ei asenneta staattista kirjastoa, jota configure ei käsittele:

```
sed -i '/LIBTOOL_INSTALL/d' c++/Makefile.in
```

```
root:/sources/ncurses-6.0# sed -i '/LIBTOOL_INSTALL/d' c++/Makefile.in
root:/sources/ncurses-6.0#
```

Ajetaan configure:

```
./configure --prefix=/usr \
            --mandir=/usr/share/man \
            --with-shared \
            --without-debug \
            --without-normal \
            --enable-pc-files \
            --enable-widec
```

Puretaan auki:

- `--enable-widex` → Aiheuttaa sen, että wide-character -kirjastot (esim. `libncursesw.so.6.0`) käännetään tavallisten sijaan (esim. `libncurses.so.6.0`). Nämä wide-character -kirjastot ovat käytettävissä sekä multibittisissä että perinteisissä 8-bittisissä lokaaleissa. Tavalliset kirjastot sen sijaan toimivat oikein ainoastaan 8-bittisissä lokaaleissa. Wide-character ja tavalliset kirjastot ovat source-yhteensopivia, mutta eivät binääriyhteensopivia.
- `--enable-pc-files` → Luo ja asentaa .pc-tiedostot.
- `--without-normal` → Estää kääntämästä ja asentamasta useimpia staattisia kirjastoja.

```
** Configuration summary for NCURSES 6.0 20150808:

    extended funcs: yes
    xterm terminfo: xterm-new

    bin directory: /usr/bin
    lib directory: /usr/lib
    include directory: /usr/include
    man directory: /usr/share/man
    terminfo directory: /usr/share/terminfo
    pkg-config directory: yes

root:/sources/ncurses-6.0#
```

Ajetaan make:

```
make
```

Tällä paketilla on testit olemassa, mutta ne voi ajaa vasta sen jälkeen, kun paketti on asennettu. Testit sijaitsevat `test/` -hakemistossa. Saman hakemiston README-tiedostossa on tarkemmat ohjeet.

Asennetaan paketti:

```
make install
```

Siirretään jaetut kirjastot `/lib` -hakemistoon, jotta ne sijaitsevat siellä, missä niiden oletetaan sijaitsevan:

```
mv -v /usr/lib/libncursesw.so.6* /lib
```

```
root:/sources/ncurses-6.0# mv -v /usr/lib/libncursesw.so.6* /lib
'/usr/lib/libncursesw.so.6' -> '/lib/libncursesw.so.6'
'/usr/lib/libncursesw.so.6.0' -> '/lib/libncursesw.so.6.0'
root:/sources/ncurses-6.0#
```

Koska kirjastot on siirretty, yksi symbolinen linkki osoittaa olemattomaan tiedostoon. Luodaan se uudestaan:

```
ln -sfv ../../lib/$(readlink /usr/lib/libncursesw.so)
/usr/lib/libncursesw.so
```

```
<adlink /usr/lib/libncursesw.so) /usr/lib/libncursesw.so
'/usr/lib/libncursesw.so' -> '../../lib/libncursesw.so.6'
root:/sources/ncurses-6.0#
```

Monet ohjelmat olettavat linkkerin kykenevän löytämään non-wide-character Ncurses -kirjastoja. Huijataan sellaisia ohjelmia linkkaamaan wide-character -kirjastojen kanssa:

```
for lib in ncurses form panel menu ; do
    rm -vf /usr/lib/lib${lib}.so
    echo "INPUT(-l${lib}w)" > /usr/lib/lib${lib}.so
    ln -sfv ${lib}w.pc /usr/lib/pkgconfig/${lib}.pc
done
```

```
root:/sources/ncurses-6.0# for lib in ncurses form panel menu ; do
> rm -vf /usr/lib/lib${lib}.so
> echo "INPUT(-l${lib}w)" > /usr/lib/lib${lib}.so
> ln -sfv ${lib}w.pc /usr/lib/pkgconfig/${lib}.pc
> done
ln: failed to create symbolic link '/usr/lib/pkgconfig/ncurses.pc': No such file or directory
ln: failed to create symbolic link '/usr/lib/pkgconfig/form.pc': No such file or directory
ln: failed to create symbolic link '/usr/lib/pkgconfig/panel.pc': No such file or directory
ln: failed to create symbolic link '/usr/lib/pkgconfig/menu.pc': No such file or directory
root:/sources/ncurses-6.0#
```

Varmistetaan, että vanhat ohjelmat, jotka etsivät `-lcurses`:a käännösvaiheessa ovat silti käännettävissä:

```
rm -vf /usr/lib/libcursesw.so
echo "INPUT(-lcursesw)" > /usr/lib/libcursesw.so
ln -sfv libcurses.so /usr/lib/libcurses.so
```

```
root:/sources/ncurses-6.0# rm -vf /usr/lib/libcursesw.so
root:/sources/ncurses-6.0# echo "INPUT(-lcursesw)" > /usr/lib/libcursesw.so
root:/sources/ncurses-6.0# ln -sfv libcurses.so /usr/lib/libcurses.so
'/usr/lib/libcurses.so' -> 'libcurses.so'
root:/sources/ncurses-6.0#
```

Halutessasi voit asentaa Ncurses:n dokumentaation:

```
mkdir -v /usr/share/doc/ncurses-6.0
cp -v -R doc/* /usr/share/doc/ncurses-6.0
```

```
root:/sources/ncurses-6.0# mkdir -v /usr/share/doc/ncurses-6.0
mkdir: created directory '/usr/share/doc/ncurses-6.0'
root:/sources/ncurses-6.0# cp -v -R doc/* /usr/share/doc/ncurses-6.0
'doc/hackguide.doc' -> '/usr/share/doc/ncurses-6.0/hackguide.doc'
'doc/html' -> '/usr/share/doc/ncurses-6.0/html'
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf ncurses-6.0
```

```
root:/sources/ncurses-6.0# cd ..
root:/sources# rm -rf ncurses-6.0
root:/sources#
```

2.15.15 Attr-2.4.47

Käännösaika: alle 0,1 SBU

Asennettavat ohjelmat: attr, getfattr, setattr

Asennettava kirjasto: libattr.so

Asennettavat hakemistot: /usr/include/attr ja /usr/share/doc/attr-2.4.47

Ensimmäiseksi puretaan Attr-paketti ja siirrytään purettuun hakemistoon:

```
tar xf attr-2.4.47.src.tar.gz
cd attr-2.4.47
```

```
root:/sources# tar xf attr-2.4.47.src.tar.gz
root:/sources# cd attr-2.4.47
root:/sources/attr-2.4.47#
```

Muokataan dokumentaatiohakemistoa, jotta siitä tulee versioitu hakemisto:

```
sed -i -e 's|/@pkg_name@|&-@pkg_version@|' include/builddefs.in
```

```
<e 's|/@pkg_name@|&-@pkg_version@|' include/builddefs.in
root:/sources/attr-2.4.47#
```

Estetään manuaalisivujen asennus, jotka man-pages jo asensi aiemmin:

```
sed -i -e "/SUBDIRS/s|man[25]||g" man/Makefile
```

```
root:/sources/attr-2.4.47# sed -i -e "/SUBDIRS/s|man[25]||g" man/Makefile
root:/sources/attr-2.4.47#
```

Ajetaan configure:

```
./configure --prefix=/usr \
            --bindir=/bin \
            --disable-static
```

```
root:/sources/attr-2.4.47# ./configure --prefix=/usr \
> --bindir=/bin \
> --disable-static
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
```

Käännetään paketti:

```
make
```

```
root:/sources/attr-2.4.47# make
/tools/bin/make include libmisc libattr attr getfattr setfattr examples test m4 man do
c po debian package
make[1]: Entering directory '/sources/attr-2.4.47'
/tools/bin/make -C include
make[2]: Entering directory '/sources/attr-2.4.47/include'
```

Testit täytyy ajaa tiedostojärjestelmässä, joka tukee laajennettuja attribuutteja, kuten ext2, ext3 tai ext4 -tiedostojärjestelmät. Testien tiedetään epäonnistuvan myös, jos useita testejä ajetaan yhtäaikaaisesti (-j -asetus suurempi kuin 1). Ajetaan testit:

```
make -j1 tests root-tests
```

```
[118] # make -j1 tests root-tests
29 commands (29 passed, 0 failed)
make[1]: Leaving directory '/sources/attr-2.4.47/test'
root:/sources/attr-2.4.47#
```

Asennetaan paketti:

```
make install install-dev install-lib
chmod -v 755 /usr/lib/libattr.so
```

```
root:/sources/attr-2.4.47# make install install-dev install-lib
/tools/bin/make include libmisc libattr attr getfattr setfattr examples test m4 man do
c po debian package
make[1]: Entering directory '/sources/attr-2.4.47'
/tools/bin/make -C include
make[2]: Entering directory '/sources/attr-2.4.47/include'
```

```
root:/sources/attr-2.4.47# chmod -v 755 /usr/lib/libattr.so
mode of '/usr/lib/libattr.so' changed from 0644 (rw-r--r--) to 0755 (rwxr-xr-x)
root:/sources/attr-2.4.47#
```

Jaettu kirjasto pitää siirtää /lib -hakemistoon ja sen seurauksena .so -tiedosto /usr/lib -hakemistossa pitää luoda uudelleen:

```
mv -v /usr/lib/libattr.so.* /lib
ln -sfv ../../lib/$(readlink /usr/lib/libattr.so)
/usr/lib/libattr.so
```

```
root:/sources/attr-2.4.47# mv -v /usr/lib/libattr.so.* /lib
'/usr/lib/libattr.so.1' -> '/lib/libattr.so.1'
'/usr/lib/libattr.so.1.1.0' -> '/lib/libattr.so.1.1.0'
<lib/$(readlink /usr/lib/libattr.so) /usr/lib/libattr.so
'/usr/lib/libattr.so' -> '../../lib/libattr.so.1'
root:/sources/attr-2.4.47#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf attr-2.4.47
```

```
root:/sources/attr-2.4.47# cd ..
root:/sources# rm -rf attr-2.4.47
root:/sources#
```

2.15.16 Acl-2.2.52

Käännösaika: alle 0,1 SBU

Asennettavat ohjelmat: *chacl, getfacl ja setacl*

Asennettava kirjasto: *libacl.so*

Asennettavat hakemistot: */usr/include/acl ja /usr/share/doc/acl-2.2.52*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf acl-2.2.52.src.tar.gz
cd acl-2.2.52
```

```
root:/sources# tar xf acl-2.2.52.src.tar.gz
root:/sources# cd acl-2.2.52
root:/sources/acl-2.2.52#
```

Muokataan dokumentaatiohakemistoa siten, että se on versioitu hakemisto:

```
sed -i -e 's|/@pkg_name@|&-@pkg_version@|' include/builddefs.in
```

```
<e 's|/@pkg_name@|&-@pkg_version@|' include/builddefs.in
root:/sources/acl-2.2.52#
```

Korjataan joitakin rikkoutuneita testejä:

```
sed -i "s:| sed.*::g" test/{sbits-restore,cp,misc}.test
```

```
root:/sources/acl-2.2.52# sed -i "s:| sed.*::g" test/{sbits-restore,cp,misc}.test
root:/sources/acl-2.2.52#
```

Korjataan yksi bugi:

```
sed -i -e "/TABS-1;/a if (x > (TABS-1)) x = (TABS-1);" \
libacl/__acl_to_any_text.c
```

```
root:/sources/acl-2.2.52# sed -i -e "/TABS-1;/a if (x > (TABS-1)) x = (TABS-1);" \
> libacl/_acl_to_any_text.c
root:/sources/acl-2.2.52#
```

Ajetaan configure:

```
./configure --prefix=/usr \
            --bindir=/bin \
            --disable-static \
            --libexecdir=/usr/lib
```

```
root:/sources/acl-2.2.52# ./configure --prefix=/usr \
> --bindir=/bin \
> --disable-static \
> --libexecdir=/usr/lib
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
```

Käännetään paketti:

```
make
```

```
root:/sources/acl-2.2.52# make
/tools/bin/make include libmisc libacl getfacl setfacl chacl m4 man doc po test examples package debian
make[1]: Entering directory '/sources/acl-2.2.52'
/tools/bin/make -C include
```

Acl:n testit täytyy ajaa voi ajaa sen jälkeen, kun Coreutils on käännetty Acl:n kirjastojen kanssa. Halutessasi voit palata tähän pakettiin Coreutilsin kääntämisen jälkeen ja ajaa testit komennolla `make -j1 tests`.

Asennetaan paketti:

```
make install install-dev install-lib
chmod -v 755 /usr/lib/libacl.so
```

```
root:/sources/acl-2.2.52# make install install-dev install-lib
/tools/bin/make include libmisc libacl getfacl setfacl chacl m4 man doc po test examples package debian
make[1]: Entering directory '/sources/acl-2.2.52'
/tools/bin/make -C include
make[2]: Entering directory '/sources/acl-2.2.52/include'
```

```
root:/sources/acl-2.2.52# chmod -v 755 /usr/lib/libacl.so
mode of '/usr/lib/libacl.so' changed from 0644 (rw-r--r--) to 0755 (rwxr-xr-x)
root:/sources/acl-2.2.52#
```

Jaettu kirjasto pitää siirtää `/lib` -hakemistoon, jonka seurauksena `.so` -tiedosto `/usr/lib` -hakemistossa pitää luoda uudelleen:

```
mv -v /usr/lib/libacl.so.* /lib
ln -sfv ../../lib/$(readlink /usr/lib/libacl.so)
/usr/lib/libacl.so
```

```
root:/sources/acl-2.2.52# mv -v /usr/lib/libacl.so.* /lib
'/usr/lib/libacl.so.1' -> '/lib/libacl.so.1'
'/usr/lib/libacl.so.1.1.0' -> '/lib/libacl.so.1.1.0'
<./lib/$(readlink /usr/lib/libacl.so) /usr/lib/libacl.so
'/usr/lib/libacl.so' -> ' ../../lib/libacl.so.1'
root:/sources/acl-2.2.52#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf acl-2.2.52
```

```
root:/sources/acl-2.2.52# cd ..
root:/sources# rm -rf acl-2.2.52
root:/sources#
```

2.15.17 Libcap-2.25

Käännösaika: alle 0,1 SBU

Asennettavat ohjelmat: *capsh, getcap, getpcaps ja setcap*

Asennettava kirjasto: *libcap.so*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf libcap-2.25.tar.xz
cd libcap-2.25
```

```
root:/sources# tar xf libcap-2.25.tar.xz
root:/sources# cd libcap-2.25
root:/sources/libcap-2.25#
```

Estetään staattista kirjastoa asentumasta:

```
sed -i '/install.*STALIBNAME/d' libcap/Makefile
```

```
root:/sources/libcap-2.25# sed -i '/install.*STALIBNAME/d' libcap/Makefile
root:/sources/libcap-2.25#
```

Käännetään paketti:

```
make
```

```
root:/sources/libcap-2.25# make
make -C libcap all
make[1]: Entering directory '/sources/libcap-2.25/libcap'
=> making cap_names.list.h from /sources/libcap-2.25/libcap/./libcap/include/uapi/linux/capability.h
perl -e 'while ($l=<=) { if ($l =~ /^#define[ \t](CAP[ A-Z]+)[ \t]+([0-9]+)\s+$/) { $
```

Libcap:n mukana ei tule testejä.

Asennetaan paketti:

```
make RAISE_SETFCAP=no lib=lib prefix=/usr install
chmod -v 755 /usr/lib/libcap.so
```

```
root:/sources/libcap-2.25# make RAISE_SETFCAP=no lib=lib prefix=/usr install
make -C libcap install
make[1]: Entering directory '/sources/libcap-2.25/libcap'
mkdir -p -m 0755 /usr/include/sys
install -m 0644 include/sys/capability.h /usr/include/sys
```

```
root:/sources/libcap-2.25# chmod -v 755 /usr/lib/libcap.so
mode of '/usr/lib/libcap.so' changed from 0644 (rw-r--r--) to 0755 (rwxr-xr-x)
root:/sources/libcap-2.25#
```

Puretaan auki:

- `RAISE_SETFCAP=no` → Tämä estää asennusvirheen, jos kernel tai tiedostojärjestelmä ei tue laajennettuja kykyjä (extended capabilities).
- `lib=lib` → Asentaa kirjaston `$prefix/lib` -hakemistoon sen sijaan, että asentaisi sen `$prefix/lib64` -hakemistoon `x86_64` -koneella. Asetuksella ei ole vaikutusta `x86` -koneella.

Jaettu kirjasto pitää siirtää `/lib` -hakemistoon ja sen seurauksena `.so` -tiedosto `/usr/lib` -hakemistossa pitää luoda uudelleen.

```
mv -v /usr/lib/libcap.so.* /lib
ln -sfv ../../lib/$(readlink /usr/lib/libcap.so)
/usr/lib/libcap.so
```

```
root:/sources/libcap-2.25# mv -v /usr/lib/libcap.so.* /lib
'/usr/lib/libcap.so.2' -> '/lib/libcap.so.2'
'/usr/lib/libcap.so.2.25' -> '/lib/libcap.so.2.25'
<../../lib/$(readlink /usr/lib/libcap.so) /usr/lib/libcap.so
'/usr/lib/libcap.so' -> '../../lib/libcap.so.2'
root:/sources/libcap-2.25#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf libcap-2.25
```

```
root:/sources/libcap-2.25# cd ..
root:/sources# rm -rf libcap-2.25
root:/sources#
```

2.15.18 Sed-4.4

Käännösaika: 0,3 SBU

Asennettava ohjelma: sed

Asennettava hakemisto: /usr/share/doc/sed-4.4

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf sed-4.4.tar.xz
cd sed-4.4
```

```
root:/sources# tar xf sed-4.4.tar.xz
root:/sources# cd sed-4.4
root:/sources/sed-4.4#
```

Korjataan ongelma LFS-ympäristössä ja poistetaan epäonnistuvat testi:

```
sed -i 's/usr/tools/' build-aux/help2man
sed -i 's/panic-tests.sh//' Makefile.in
```

```
root:/sources/sed-4.4# sed -i 's/usr/tools/' build-aux/help2man
root:/sources/sed-4.4# sed -i 's/panic-tests.sh//' Makefile.in
root:/sources/sed-4.4#
```

Ajetaan configure:

```
./configure --prefix=/usr --bindir=/bin
```

```
root:/sources/sed-4.4# ./configure --prefix=/usr --bindir=/bin
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
```

Käännetään paketti ja luodaan HTML-dokumentaatio:

```
make
make html
```

```
root:/sources/sed-4.4# make
GEN      lib/alloca.h
GEN      lib/configmake.h
GEN      lib/c++defs.h
GEN      lib/warn-on-use.h
```

```
root:/sources/sed-4.4# make html
Making html in po
make[1]: Entering directory '/sources/sed-4.4/po'
make[1]: Nothing to be done for 'html'.
```

Ajetaan testit:

```
make check
```

```
PASS: testsuite/temp-file-cleanup.sh
PASS: testsuite/title-case.sh
PASS: testsuite/unbuffered.sh
PASS: testsuite/follow-symlinks.sh
PASS: testsuite/follow-symlinks-stdin.sh
make[5]: Entering directory '/sources/sed-4.4'
make[5]: Nothing to be done for 'testsuite/.log'.
make[5]: Leaving directory '/sources/sed-4.4'
fatal: making test-suite.log: failed to create testsuite/.trs
fatal: making test-suite.log: failed to create testsuite/.log
make[4]: *** [Makefile:3141: test-suite.log] Error 1
make[4]: Leaving directory '/sources/sed-4.4'
make[3]: *** [Makefile:3250: check-TESTS] Error 2
make[3]: Leaving directory '/sources/sed-4.4'
make[2]: *** [Makefile:4090: check-am] Error 2
make[2]: Leaving directory '/sources/sed-4.4'
make[1]: *** [Makefile:3027: check-recursive] Error 1
make[1]: Leaving directory '/sources/sed-4.4'
make: *** [Makefile:4092: check] Error 2
root:/sources/sed-4.4#
```

Asennetaan paketti ja sen dokumentaatio:

```
make install
```

```
install -d -m755          /usr/share/doc/sed-4.4
install -m644 doc/sed.html /usr/share/doc/sed-4.4
```

```
root:/sources/sed-4.4# make install
make install-recursive
make[1]: Entering directory '/sources/sed-4.4'
Making install in po
make[2]: Entering directory '/sources/sed-4.4/po'
installing af.gmo as /usr/share/locale/af/LC_MESSAGES/sed.mo
```

```
root:/sources/sed-4.4# install -d -m755 /usr/share/doc/sed-4.4
root:/sources/sed-4.4# install -m644 doc/sed.html /usr/share/doc/sed-4.4
root:/sources/sed-4.4# █
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf sed-4.4
```

```
root:/sources/sed-4.4# cd ..
root:/sources# rm -rf sed-4.4
root:/sources# █
```

2.15.19 Shadow-4.4

Käännösaika: 0,2 SBU

Asennettavat ohjelmat: *chage, chfn, chgpasswd, chpasswd, chsh, expiry, faillog, gpasswd, groupadd, groupdel, groupmems, groupmod, grpck, grpconv, grpunconv, lastlog, login, logoutd, newgidmap, newgrp, newuidmap, newusers, nologin, passwd, pwck, pwconv, pwunconv, sg (linkki newgrp:hen), su, useradd, userdel, usermod, vigr (linkki vipw:en) ja vipw*

Asennettava hakemisto: */etc/default*

Jos haluat pakottaa vahvojen salasanojen käytön, lue [BLFS](http://www.linuxfromscratch.org/blfs/view/8.0/postlfs/cracklib.html)-kirjasta CrackLib:n asentamisesta ja asenna se ennen kuin käännät Shadow:n. Lisää `--with-libcrack` -asetus `configure`-komentoon. Linkki BLFS-kirjaan: <http://www.linuxfromscratch.org/blfs/view/8.0/postlfs/cracklib.html>

Ensimmäiseksi puretaan shadow-paketti ja siirrytään purettuun hakemistoon:

```
tar xf shadow-4.4.tar.xz
cd shadow-4.4
```

```
root:/sources# tar xf shadow-4.4.tar.xz
root:/sources# cd shadow-4.4
root:/sources/shadow-4.4#
```

Estetään `groups`-ohjelman ja sen man-sivujen asentuminen, koska Coreutils tarjoaa paremman version. Lisäksi estetään niiden man-sivujen asentuminen, jotka man-pages on jo aiemmin asentanut.

```
sed -i 's/groups$(EXEEXT) //' src/Makefile.in
find man -name Makefile.in -exec sed -i 's/groups\.1 / /'
{} \;
find man -name Makefile.in -exec sed -i 's/getspnam\.3 / /'
{} \;
find man -name Makefile.in -exec sed -i 's/passwd\.5 / /'
{} \;
```

```
root:/sources/shadow-4.4# sed -i 's/groups$(EXEEXT) //' src/Makefile.in
<name Makefile.in -exec sed -i 's/groups\.1 / ' {} \;
<name Makefile.in -exec sed -i 's/getspram\.3 / ' {} \;
<name Makefile.in -exec sed -i 's/passwd\.5 / ' {} \;
root:/sources/shadow-4.4#
```

Sen sijaan, että käytettäisiin salasanojen kryptaukseen oletuksena olevaa crypt-metodia, käytetään turvallisempaa SHA-512 -metodia. Se sallii myös pidemmät kuin 8-merkkiset salasanat. Sitä tarvitaan myös vaihtaaksemme vanhentunut /var/spool/mail, jota Shadow käyttää oletuksena, /var/mail -hakemistoon.

```
sed -i -e 's@#ENCRYPT_METHOD DES@ENCRYPT_METHOD SHA512@' \
      -e 's@/var/spool/mail@/var/mail@' etc/login.defs
```

```
< -i -e 's@#ENCRYPT_METHOD DES@ENCRYPT_METHOD SHA512@' \
>      -e 's@/var/spool/mail@/var/mail@' etc/login.defs
root:/sources/shadow-4.4#
```

Korjataan bugi, joka estää useradd:ia asettamasta shell entryä /etc/passwd -tiedostoon.

```
echo '--- src/useradd.c      (old)
+++ src/useradd.c      (new)
@@ -2027,6 +2027,8 @@
        is_shadow_grp = sgr_file_present ();
    #endif

+       get_defaults ();
+
        process_flags (argc, argv);

    #ifdef ENABLE_SUBIDS
@@ -2036,8 +2038,6 @@
        (!user_id || (user_id <= uid_max && user_id >=
uid_min));
    #endif
                                     /* ENABLE_SUBIDS */
```

```
-      get_defaults ();
-
#ifdef ACCT_TOOLS_SETUID
#ifdef USE_PAM
    {' | patch -p0 -l
```

```
root:/sources/shadow-4.4# echo '--- src/useradd.c    (old)
> +++ src/useradd.c    (new)
> @@ -2027,6 +2027,8 @@
>      is_shadow_grp = sgr_file_present ();
>  #endif
>
> +      get_defaults ();
> +
>      process_flags (argc, argv);
>
>  #ifdef ENABLE_SUBIDS
> @@ -2036,8 +2038,6 @@
>      (!user_id || (user_id <= uid_max && user_id >= uid_min));
>  #endif
>                                     /* ENABLE_SUBIDS */
>
> -      get_defaults ();
> -
>  #ifdef ACCT_TOOLS_SETUID
>  #ifdef USE_PAM
>      {' | patch -p0 -l
patching file src/useradd.c
root:/sources/shadow-4.4#
```

Jos päätit kääntää Shadow:n Cracklibin kanssa, aja seuraava:

```
sed -i 's@DICTPATH.*@DICTPATH\t/lib/cracklib/pw_dict@'
etc/login.defs
```

```
<PATH.*@DICTPATH\t/lib/cracklib/pw_dict@' etc/login.defs
root:/sources/shadow-4.4#
```

Tehdään oletususeradd yhteneväiseksi LFS groups -tiedoston kanssa:

```
sed -i 's/1000/999/' etc/useradd
```

```
root:/sources/shadow-4.4# sed -i 's/1000/999/' etc/useradd
root:/sources/shadow-4.4#
```

Korjataan yksi tietoturvaongelma:

```
sed -i -e '47 d' -e '60,65 d' libmisc/myname.c
```

```
root:/sources/shadow-4.4# sed -i -e '47 d' -e '60,65 d' libmisc/myname.c  
root:/sources/shadow-4.4#
```

Valmistellaan Shadow kääntämistä varten:

```
./configure --sysconfdir=/etc --with-group-name-max-length=32
```

Puretaan auki:

- `--with-group-name-max-length=32` → Käyttäjänimen maksimipituus on 32 merkkiä. Asetetaan ryhmän nimen maksipituudeksi sama.

```
shadow will be compiled with the following features:
```

auditing support:	no
CrackLib support:	no
PAM support:	no
SELinux support:	no
ACL support:	yes
Extended Attributes support:	yes
tcb support (incomplete):	no
shadow group support:	yes
S/Key support:	no
SHA passwords encryption:	yes
nscd support:	yes
subordinate IDs support:	yes

```
root:/sources/shadow-4.4#
```

Käännetään paketti:

```
make
```

```
root:/sources/shadow-4.4# make
make all-recursive
make[1]: Entering directory '/sources/shadow-4.4'
Making all in po
make[2]: Entering directory '/sources/shadow-4.4/po'
make[2]: Nothing to be done for 'all'.
make[2]: Leaving directory '/sources/shadow-4.4/po'
```

Shadow:n mukana ei tule testejä.

Asennetaan paketti:

```
make install
```

```
root:/sources/shadow-4.4# make install
Making install in po
make[1]: Entering directory '/sources/shadow-4.4/po'
/tools/bin/mkdir -p /usr/share
installing bs.gmo as /usr/share/locale/bs/LC_MESSAGES/shadow.mo
installing ca.gmo as /usr/share/locale/ca/LC_MESSAGES/shadow.mo
installing cs.gmo as /usr/share/locale/cs/LC_MESSAGES/shadow.mo
```

Siirretään väärin sijoitettu ohjelma oikeaan paikkaan:

```
mv -v /usr/bin/passwd /bin
```

```
root:/sources/shadow-4.4# mv -v /usr/bin/passwd /bin
'/usr/bin/passwd' -> '/bin/passwd'
root:/sources/shadow-4.4#
```

2.15.19.1 Shadow:n konfiguroiminen

Shadow sisältää työkaluja käyttäjien ja ryhmien lisäämiseen, muokkaamiseen ja poistamiseen, niiden salasanojen asettamiseen ja vaihtamiseen sekä muihin admin-hommiin. Jos haluat lukea selityksen siitä, mitä *password shadowing* tarkoittaa, se löytyy `doc/HOWTO` -tiedostosta. Jos käytetään Shadow-tukea, täytyy muistaa, että ohjelmat, joiden tarvitsee varmentaa salasanaja (FTP-ohjelmat, pop3-taustaprosessit jne.) täytyy olla Shadow-yhteensopivia. Käytännössä se tarkoittaa, että niiden täytyy pystyä toimimaan shadowed-salasanojen kanssa.

Otetaan shadowed-salasanat käyttöön:

```
pwconv
```

```
root:/sources/shadow-4.4# pwconv  
root:/sources/shadow-4.4# █
```

Otetaan shadowed-ryhmäsalasanat käyttöön:

```
grpconv
```

```
root:/sources/shadow-4.4# grpconv  
root:/sources/shadow-4.4# █
```

Shadow:n stock-konfiguraatio `useradd`-työkalua varten sisältää muutamia huomautuksia, jotka vaativat selitystä. Oletustoiminto `useradd`-työkalulle on luoda käyttäjä ja samanniminen ryhmä. Oletuksena used ID (UID) ja group ID (GID) -numerot alkavat 1000:lla. Tämä tarkoittaa, että mikäli et aseta parametreja `useradd`ille, jokainen käyttäjä on uniikin ryhmän jäsen. Mikäli tätä ei haluta, täytyy asettaa `-g` -parametri `useradd`ille. Oletusparametrit varastoidaan `/etc/default/useradd` -tiedostoon. Sinun täytyy mahdollisesti muokata kahta parametria siinä tiedostossa omia tarpeitasi varten.

`/etc/default/useradd` -parametrien selitykset:

- `GROUP=1000` → Tämä parametri asettaa `/etc/group` -tiedostoon group-numeroiden alun. Voit muokata sitä mihin tahansa haluat. Huomaa, että `useradd` ei ikinä kierrätä UID:a tai GID:a. Jos tässä parametrissa identifioitu numero on käytössä, se käyttää seuraavaa vapaana olevaa numeroa.
- `CREATE_MAIL_SPOOL=yes` → Tämä parametri aiheuttaa sen, että `useradd` luo mailbox-tiedoston uudelle käyttäjälle. `useradd` asettaa tämän tiedoston ryhmäomistajuuden mail-ryhmälle 0660-oikeuksin. Mikäli et halua, että `useradd` luo näitä mailbox-tiedostoja, aja seuraava komento:

```
sed -i 's/yes/no/' /etc/default/useradd
```

2.15.19.2 Rootin salasanan asettaminen

Valitse sanasana rootille ja aseta se ajamalla seuraava komento:

```
passwd root
```

```
root:/sources/shadow-4.4# passwd root
Changing password for root
Enter the new password (minimum of 5 characters)
Please use a combination of upper and lower case letters and numbers.
New password:
Re-enter new password:
passwd: password changed.
root:/sources/shadow-4.4#
```

Lopuksi vielä, kun muistetaan, poistetaan Shadow:n purettu hakemisto:

```
cd ..
rm -rf shadow-4.4
```

```
root:/sources/shadow-4.4# cd ..
root:/sources# rm -rf shadow-4.4
root:/sources#
```

2.15.20 Psmisc-22.21

Käännösaika: alle 0,1 SBU

Asennettavat ohjelmat: *fuser, killall, peekfd, prtstat, pstree ja pstree.x11* (linkki *pstree:hen*)

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf psmisc-22.21.tar.gz
cd psmisc-22.21
```

```
root:/sources# tar xf psmisc-22.21.tar.gz
root:/sources# cd psmisc-22.21
root:/sources/psmisc-22.21#
```

Valmistele Psmisc kääntämistä varten:

```
./configure --prefix=/usr
```

```
root:/sources/psmisc-22.21# ./configure --prefix=/usr
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
```

Käännetään paketti:

```
make
```

```
root:/sources/psmisc-22.21# make
make all-recursive
make[1]: Entering directory '/sources/psmisc-22.21'
Making all in doc
make[2]: Entering directory '/sources/psmisc-22.21/doc'
```

Psmisc:n mukana ei tule testejä.

Asennetaan paketti:

```
make install
```

```
root:/sources/psmisc-22.21# make install
Making install in doc
make[1]: Entering directory '/sources/psmisc-22.21/doc'
make[2]: Entering directory '/sources/psmisc-22.21/doc'
make[2]: Nothing to be done for 'install-exec-am'
```

Siirretään killall ja fuser -ohjelmat [FHS:n](#) määrittelemään paikkaan:

```
mv -v /usr/bin/fuser /bin
mv -v /usr/bin/killall /bin
```

```
root:/sources/psmisc-22.21# mv -v /usr/bin/fuser /bin
'/usr/bin/fuser' -> '/bin/fuser'
root:/sources/psmisc-22.21# mv -v /usr/bin/killall /bin
'/usr/bin/killall' -> '/bin/killall'
root:/sources/psmisc-22.21# █
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf psmisc-22.21
```

```
root:/sources/psmisc-22.21# cd ..
root:/sources# rm -rf psmisc-22.21
root:/sources# █
```

2.15.21 Iana-Etc-2.30

Käännösaika: alle 0,1 SBU

Asennettavat tiedostot: /etc/protocols ja /etc/services

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf iana-etc-2.30.tar.bz2
cd iana-etc-2.30
```

```
root:/sources# tar xf iana-etc-2.30.tar.bz2
root:/sources# cd iana-etc-2.30
root:/sources/iana-etc-2.30#
```

Seuraava komento muuntaa IANA:n tarjoaman raakadatan oikeaan muotoon /etc/protocols ja /etc/services -datatiedostoja varten:

```
make
```

```
root:/sources/iana-etc-2.30# make
ln -f -s protocol-numbers.dist protocol-numbers
gawk --re-interval -f protocols.gawk -v strip=no \
    protocol-numbers > protocols
ln -f -s port-numbers.dist port-numbers
gawk -f services.gawk -v strip=no port-numbers > services
root:/sources/iana-etc-2.30#
```

Tämän paketin mukana ei tule testejä.

Asennetaan:

```
make install
```

```
root:/sources/iana-etc-2.30# make install
install -d /etc
install -m 644 protocols /etc
install -m 644 services /etc
root:/sources/iana-etc-2.30#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf iana-etc-2.30
```

```
root:/sources/iana-etc-2.30# cd ..  
root:/sources# rm -rf iana-etc-2.30  
root:/sources# █
```

2.15.22 M4-1.4.18

Käännösaika: 0,4 SBU

Asennettava ohjelma: M4

Ensimmäiseksi puretaan M4 ja siirrytään purettuun hakemistoon:

```
tar xf m4-1.4.18.tar.xz
cd m4-1.4.18
```

```
root:/sources# tar xf m4-1.4.18.tar.xz
root:/sources# cd m4-1.4.18
root:/sources/m4-1.4.18#
```

Valmistellaan M4 kääntämistä varten:

```
./configure --prefix=/usr
```

```
root:/sources/m4-1.4.18# ./configure --prefix=/usr
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE) ... yes
```

Käännetään paketti:

```
make
```

```
root:/sources/m4-1.4.18# make
make all-recursive
make[1]: Entering directory '/sources/m4-1.4.18'
Making all in .
make[2]: Entering directory '/sources/m4-1.4.18'
make[2]: Nothing to be done for 'all-am'.
make[2]: Leaving directory '/sources/m4-1.4.18'
```

Ajetaan testit:

```
make check
```

```
=====
Testsuite summary for GNU M4 1.4.18
=====
# TOTAL: 170
# PASS: 157
# SKIP: 13
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan:

```
make install
```

```
root:/sources/m4-1.4.18# make install
make install-recursive
make[1]: Entering directory '/sources/m4-1.4.18'
Making install in .
make[2]: Entering directory '/sources/m4-1.4.18'
make[3]: Entering directory '/sources/m4-1.4.18'
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf m4-1.4.18
```

```
root:/sources/m4-1.4.18# cd ..
root:/sources# rm -rf m4-1.4.18
root:/sources#
```

2.15.23 Bison-3.0.4

Käännösaika: 0,3 SBU

Asennettavat ohjelmat: bison ja yacc

Asennettava kirjasto: liby.a

Asennettava hakemisto: /usr/share/bison

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf bison-3.0.4.tar.xz
cd bison-3.0.4
```

```
root:/sources# tar xf bison-3.0.4.tar.xz
```

```
root:/sources# cd bison-3.0.4
root:/sources/bison-3.0.4#
```

Valmistellaan Bison kääntämistä varten:

```
./configure --prefix=/usr --docdir=/usr/share/doc/bison-3.0.4
```

```
./configure --prefix=/usr --docdir=/usr/share/doc/bison-3.0.4
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
checking for gawk... gawk
```

Käännetään paketti:

```
make
```

```
root:/sources/bison-3.0.4# make
GEN      examples/extracted.stamp
extexi:  GEN examples/rpcalc/rpcalc.y
extexi:  GEN examples/mfcalc/mfcalc.y
extexi:  GEN examples/mfcalc/calc.h
extexi:  GEN examples/calcu/calcu_driver.bb
```

Bisonin testit voi ajaa vasta flexin asennuksen jälkeen. Halutessasi voit palata tekemään testit myöhemmin komennolla `make check`.

Asennetaan paketti:

```
make install
```

```
root:/sources/bison-3.0.4# make install
make install-recursive
make[1]: Entering directory '/sources/bison-3.0.4'
Making install in po
make[2]: Entering directory '/sources/bison-3.0.4/po'
installing da.gmo as /usr/share/locale/da/LC_MESSAGES/bison.mo
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf bison-3.0.4
```

```
root:/sources/bison-3.0.4# cd ..
root:/sources# rm -rf bison-3.0.4
root:/sources#
```

2.15.24 Flex-2.6.3

Käännösaika: 0,4 SBU

Asennettavat ohjelmat: *flex*, *flex++* (linkki *flex:iin*) ja *lex* (linkki *flex:iin*)

Asennettavat kirjastot: *libfl.so* ja *libfl_pic.so*

Asennettava hakemisto: */usr/share/doc/flex-2.6.3*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf flex-2.6.3.tar.gz
cd flex-2.6.3
```

```
root:/sources# tar xf flex-2.6.3.tar.gz
root:/sources# cd flex-2.6.3
root:/sources/flex-2.6.3#
```

Käännösprosessi olettaa, että help2man-ohjelma on saatavilla man-sivun luomiseen. Koska näin ei ole, käytetään ympäristömuuttujaa ohittaaksemme luomisprosessin. Valmistellaan flex kääntämistä varten:

```
HELP2MAN=/tools/bin/true \
./configure --prefix=/usr --docdir=/usr/share/doc/flex-
2.6.3
```

```
root:/sources/flex-2.6.3# HELP2MAN=/tools/bin/true \
> ./configure --prefix=/usr --docdir=/usr/share/doc/flex-2.6.3
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
```

Käännetään paketti:

```
make
```

```
root:/sources/flex-2.6.3# make
Making all in lib
make[1]: Entering directory '/sources/flex-2.6.3/lib'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory '/sources/flex-2.6.3/lib'
```

Ajetaan testit. cxx_restart on sellainen testi, jonka tiedetään epäonnistuvan.

```
make check
```

```
=====
Testsuite summary for the fast lexical analyser generator 2.6.3
=====
# TOTAL: 114
# PASS: 113
# SKIP: 0
# XFAIL: 0
# FAIL: 1
# XPASS: 0
# ERROR: 0
=====
See tests/test-suite.log
Please report to flex-help@lists.sourceforge.net
=====
```

Asennetaan paketti:

```
make install
```

```
root:/sources/flex-2.6.3# make install
Making install in lib
make[1]: Entering directory '/sources/flex-2.6.3/lib'
make[2]: Entering directory '/sources/flex-2.6.3/lib'
make[2]: Nothing to be done for 'install-exec-am'.
```

Jotkin ohjelmat eivät vielä tiedä flexistä, vaan yrittävät ajaa sen edeltäjää, lexiä. Tukeaksemme näitä ohjelmia, luodaan symbolinen linkki nimeltä lex, joka ajaa flexin lexin emulaatiotilassa.

```
ln -sv flex /usr/bin/lex
```

```
root:/sources/flex-2.6.3# ln -sv flex /usr/bin/lex
'/usr/bin/lex' -> 'flex'
root:/sources/flex-2.6.3# █
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf flex-2.6.3
```

```
root:/sources/flex-2.6.3# cd ..  
root:/sources# rm -rf flex-2.6.3  
root:/sources# █
```

2.15.25 Grep-3.0

Käännösaika: 0,4 SBU

Asennettavat ohjelmat: *egrep, fgrep ja grep*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf grep-3.0.tar.xz
cd grep-3.0
```

```
root:/sources# tar xf grep-3.0.tar.xz
root:/sources# cd grep-3.0
root:/sources/grep-3.0#
```

Valmistellaan grep kääntämistä varten:

```
./configure --prefix=/usr --bindir=/bin
```

```
root:/sources/grep-3.0# ./configure --prefix=/usr --bindir=/bin
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
```

Käännetään paketti:

```
make
```

```
root:/sources/grep-3.0# make
make all-recursive
make[1]: Entering directory '/sources/grep-3.0'
Making all in po
make[2]: Entering directory '/sources/grep-3.0/po'
make[2]: Nothing to be done for 'all'.
make[2]: Leaving directory '/sources/grep-3.0/po'
```

Ajetaan testit:

```
make check
```

```
=====
Testsuite summary for GNU grep 3.0
=====
# TOTAL: 109
# PASS: 88
# SKIP: 19
# XFAIL: 2
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

```
=====
Testsuite summary for GNU grep 3.0
=====
# TOTAL: 143
# PASS: 136
# SKIP: 7
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan paketti:

```
make install
```

```
root:/sources/grep-3.0# make install
Making install in po
make[1]: Entering directory '/sources/grep-3.0/po'
installing af.gmo as /usr/share/locale/af/LC_MESSAGES/grep.mo
installing be.gmo as /usr/share/locale/be/LC_MESSAGES/grep.mo
installing bg.gmo as /usr/share/locale/bg/LC_MESSAGES/grep.mo
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf grep-3.0
```

```
root:/sources/grep-3.0# cd ..
root:/sources# rm -rf grep-3.0
root:/sources#
```

2.15.26 Readline-7.0

Käännösaika: 0,1 SBU

Asennettavat kirjastot: libhistory.so ja libreadline.so

Asennettavat hakemistot: /usr/include/readline, /usr/share/readline ja /usr/share/doc/readline-7.0

Ensimmäiseksi puretaan readline-paketti ja siirrytään purettuun hakemistoon:

```
tar xf readline-7.0.tar.gz
cd readline-7.0
```

```
root:/sources# tar xf readline-7.0.tar.gz
root:/sources# cd readline-7.0
root:/sources/readline-7.0#
```

Readlinen uudelleenasetaminen voi aiheuttaa vanhojen kirjastojen uudelleen nimeämisen <kirjastonnimi>.old-nimisiksi. Tavallisesti tämä ei ole ongelma, mutta saattaa joissain tapauksissa aiheuttaa linkkausbugin ldconfigissa. Tämä voidaan välttää ajamalla kaksi seuraava sed-komentoa:

```
sed -i '/MV.*old/d' Makefile.in
sed -i '/{OLDSUFF}/c:' support/shlib-install
```

```
root:/sources/readline-7.0# sed -i '/MV.*old/d' Makefile.in
root:/sources/readline-7.0# sed -i '/{OLDSUFF}/c:' support/shlib-install
root:/sources/readline-7.0#
```

Valmistellaan Readline kääntämistä varten:

```
./configure --prefix=/usr \
            --disable-static \
            --docdir=/usr/share/doc/readline-7.0
```

```

root:/sources/readline-7.0# ./configure --prefix=/usr \
> --disable-static \
> --docdir=/usr/share/doc/readline-7.0
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu

Beginning configuration for readline-7.0 for x86_64-unknown-linux-gnu

checking whether make sets $(MAKE)... yes
checking for gcc... gcc

```

Käännetään paketti:

```
make SHLIB_LIBS=-lcurses
```

```

root:/sources/readline-7.0# make SHLIB_LIBS=-lcurses
test -d shlib || mkdir shlib
( cd shlib ; make all )
make[1]: Entering directory '/sources/readline-7.0/shlib'
rm -f readline.so
gcc -c -DHAVE_CONFIG_H -I. -I. -I. -I. -DBUILD_LIBRARY_VERSION='"7.0"' -D_GNU_SOURCE -fPIC -o readline.o readline.c

```

Puretaan auki:

- `SHLIB_LIBS=-lcurses` → Pakottaa Readlinen linkkaamaan libncursesw-kirjastoa vastaan.

Tämän paketin mukana ei tule testejä.

Asennetaan paketti:

```
make SHLIB_LIBS=-lcurses install
```

```

root:/sources/readline-7.0# make SHLIB_LIBS=-lcurses install
/bin/sh ./support/mkinstalldirs /usr/include \
    /usr/include/readline /usr/lib \
    /usr/share/info /usr/share/man/man3 /usr/share/doc/readline-7.0
mkdir -p -- /usr/include/readline /usr/lib /usr/share/info /usr/share/man/man3 /usr/sh

```

Siirretään dynaamiset kirjastot parempaan paikkaan ja korjataan joitakin symbolisia linkkejä:

```

mv -v /usr/lib/lib{readline,history}.so.* /lib
ln -sfv ../../lib/$(readlink /usr/lib/libreadline.so)
    /usr/lib/libreadline.so
ln -sfv ../../lib/$(readlink /usr/lib/libhistory.so )
    /usr/lib/libhistory.so

```

```

root:/sources/readline-7.0# mv -v /usr/lib/lib{readline,history}.so.* /lib
'/usr/lib/libreadline.so.7' -> '/lib/libreadline.so.7'
'/usr/lib/libreadline.so.7.0' -> '/lib/libreadline.so.7.0'
'/usr/lib/libhistory.so.7' -> '/lib/libhistory.so.7'
'/usr/lib/libhistory.so.7.0' -> '/lib/libhistory.so.7.0'
<adlink /usr/lib/libreadline.so) /usr/lib/libreadline.so
'/usr/lib/libreadline.so' -> '../lib/libreadline.so.7'
<adlink /usr/lib/libhistory.so) /usr/lib/libhistory.so
'/usr/lib/libhistory.so' -> '../lib/libhistory.so.7'
root:/sources/readline-7.0# █

```

Halutessasi, asenna dokumentaatio:

```

install -v -m644 doc/*.{ps,pdf,html,dvi}
/usr/share/doc/readline-7.0

```

```

<644 doc/*.{ps,pdf,html,dvi} /usr/share/doc/readline-7.0
'doc/history.ps' -> '/usr/share/doc/readline-7.0/history.ps'
'doc/history_3.ps' -> '/usr/share/doc/readline-7.0/history_3.ps'
'doc/readline.ps' -> '/usr/share/doc/readline-7.0/readline.ps'
'doc/readline_3.ps' -> '/usr/share/doc/readline-7.0/readline_3.ps'
'doc/rluserman.ps' -> '/usr/share/doc/readline-7.0/rluserman.ps'
'doc/history.pdf' -> '/usr/share/doc/readline-7.0/history.pdf'
'doc/readline.pdf' -> '/usr/share/doc/readline-7.0/readline.pdf'
'doc/rluserman.pdf' -> '/usr/share/doc/readline-7.0/rluserman.pdf'
'doc/history.html' -> '/usr/share/doc/readline-7.0/history.html'
'doc/readline.html' -> '/usr/share/doc/readline-7.0/readline.html'
'doc/rluserman.html' -> '/usr/share/doc/readline-7.0/rluserman.html'
'doc/history.dvi' -> '/usr/share/doc/readline-7.0/history.dvi'
'doc/readline.dvi' -> '/usr/share/doc/readline-7.0/readline.dvi'
'doc/rluserman.dvi' -> '/usr/share/doc/readline-7.0/rluserman.dvi'
root:/sources/readline-7.0# █

```

Lopuksi poistetaan purettu hakemisto:

```

cd ..
rm -rf readline-7.0

```

```

root:/sources/readline-7.0# cd ..
root:/sources# rm -rf readline-7.0
root:/sources# █

```

2.15.27 Bash-4.4

Käännösaika: 1,7 SBU

Asennettavat ohjelmat: bash, bashbug ja sh (linkki bash:iin)

Asennettava hakemisto: /usr/share/doc/bash-4.4

Ensimmäiseksi puretaan bash ja siirrytään purettuun hakemistoon:

```
tar xf bash-4.4.tar.gz
cd bash-4.4
```

```
root:/sources# tar xf bash-4.4.tar.gz
```

```
root:/sources# cd bash-4.4
root:/sources/bash-4.4#
```

Valmistellaan Bash kääntämistä varten:

```
./configure --prefix=/usr \
            --docdir=/usr/share/doc/bash-4.4 \
            --without-bash-malloc \
            --with-installed-readline
```

Puretaan auki:

- `--with-installed-readline` → Pakottaa Bashin käyttämään jo asennettua readline-kirjastoa sen sijaan, että se käyttäisi omaa versiotaan.

Käännetään paketti:

```
make
```

```
root:/sources/bash-4.4# ./configure --prefix=/usr \
> --docdir=/usr/share/doc/bash-4.4 \
> --without-bash-malloc \
> --with-installed-readline
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu

Beginning configuration for bash-4.4-release for x86_64-unknown-linux-gnu
checking for gcc... gcc
```

Valmistellaan testejä varten. Varmistetaan, että nobody-käyttäjä voi kirjoittaa sources-puuhun:

```
chown -Rv nobody .
```

Ajetaan testit nobody-käyttäjänä.

```
su nobody -s /bin/bash -c "PATH=$PATH make tests"
```

```
root:/sources/bash-4.4# su nobody -s /bin/bash -c "PATH=$PATH make tests"
rm -f mksyntax
gcc -DPROGRAM='\"bash\"' -DCONF_HOSTTYPE='\"x86_64\"' -DCONF_OSTYPE='\"linux-gnu\"' -DCONF_MACHTYPE='\"x86_64-unknown-linux-gnu\"' -DCONF_VENDOR='\"unknown\"' -DLOCALEDIR='\"/usr/share/locale\"' -DPACKAGE='\"bash\"' -DSHELL -DHAVE_CONFIG_H -I. -I. -I./include -I./lib -g -O2 -Wno-parentheses -Wno-format-security -rdynamic -g -O2 -Wno-parentheses -Wno-format-security
```

Asennetaan paketti ja siirretään suoritettava päätiedosto /bin -hakemistoon.

```
make install
mv -vf /usr/bin/bash /bin
```

```
root:/sources/bash-4.4# make install

*****
*
* GNU bash, version 4.4.0(1)-release (x86_64-unknown-linux-gnu)
*
*****

make[1]: Entering directory '/sources/bash-4.4/support'
rm -f man2html.o
gcc -c -DHAVE_CONFIG_H -DSHELL -I/sources/bash-4.4 -I. -g -O2 -Wno-parentheses -Wno-format-security man2html.c
gcc -DHAVE_CONFIG_H -DSHELL -I/sources/bash-4.4 -I. -g -O2 -Wno-parentheses -Wno-format-security
```

```
root:/sources/bash-4.4# mv -vf /usr/bin/bash /bin
'/usr/bin/bash' -> '/bin/bash'
root:/sources/bash-4.4#
```

Ajetaan vasta käännetty bash-ohjelma, joka korvaa tällä hetkellä suoritettavan:

```
exec /bin/bash --login +h
```

```
root:/sources/bash-4.4# exec /bin/bash --login +h
root:/sources/bash-4.4#
```

Parametrit, joita käytettiin, tekevät bashista interaktiivisen login shellin ja estävät hashing:n, jotta uudet ohjelmat löydetään sitä mukaa, kun ne ovat saatavilla.

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf bash-4.4
```

```
root:/sources/bash-4.4# cd ..  
root:/sources# rm -rf bash-4.4  
root:/sources#
```

2.15.28 Bc-1.06.95

Käännösaika: alle 0,1 SBU

Asennettavat ohjelmat: bc ja dc

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf bc-1.06.95.tar.bz2
cd bc-1.06.95
```

```
root:/sources# tar xf bc-1.06.95.tar.bz2
root:/sources# cd bc-1.06.95
root:/sources/bc-1.06.95#
```

Korjataan joitain pieniä muistivuotoja:

```
patch -Np1 -i ../bc-1.06.95-memory_leak-1.patch
```

```
root:/sources/bc-1.06.95# patch -Np1 -i ../bc-1.06.95-memory_leak-1.patch
patching file bc/bc.y
patching file bc/storage.c
patching file bc/util.c
root:/sources/bc-1.06.95#
```

Ajetaan configure:

```
./configure --prefix=/usr \
            --with-readline \
            --mandir=/usr/share/man \
            --infodir=/usr/share/info
```

```
root:/sources/bc-1.06.95# ./configure --prefix=/usr --with-readline --mandir=/usr/share/man --infodir=/usr/share/info
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for awk... awk
```

Puretaan auki:

- `--with-readline` → Pakottaa Bc:n käyttämään järjestelmään jo asennettua readline-kirjastoa sen sijaan, että se käyttäisi omaa versiotaan.

Käännetään paketti:

make

```
root:/sources/bc-1.06.95# make
make all-recursive
make[1]: Entering directory '/sources/bc-1.06.95'
Making all in lib
make[2]: Entering directory '/sources/bc-1.06.95/lib'
if gcc -DHAVE_CONFIG_H -I. -I -I -I -I -I -I / -b -g -O2 -Wall -funsigned-char
```

Ajetaan testit. Tulostetta tulee melko paljon, joten voit halutessasi ohjata sen tiedostoon.

```
echo "quit" | ./bc/bc -l Test/checklib.b
```

Asennetaan paketti:

```
make install
```

```
root:/sources/bc-1.06.95# make install
Making install in lib
make[1]: Entering directory '/sources/bc-1.06.95/lib'
make[2]: Entering directory '/sources/bc-1.06.95/lib'
make[2]: Nothing to be done for 'install exec am'
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf bc-1.06.95
```

```
root:/sources/bc-1.06.95# cd ..
root:/sources# rm -rf bc-1.06.95
root:/sources#
```

2.15.29 Libtool-2.4.6

Käännösaika: 2,0 SBU

Asennettavat ohjelmat: libtool ja libtoolize

Asennettavat kirjastot: libltdl.so

Asennettavat hakemistot: /usr/include/libltdl ja /usr/share/libtool

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf libtool-2.4.6.tar.xz
cd libtool-2.4.6
```

```
root:/sources# tar xf libtool-2.4.6.tar.xz
root:/sources# cd libtool-2.4.6
root:/sources/libtool-2.4.6#
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/usr
```

```
root:/sources/libtool-2.4.6# ./configure --prefix=/usr
## ----- ##
## Configuring libtool 2.4.6 ##
## ----- ##

checking for GNU M4 that supports accurate traces... /usr/bin/m4
checking whether /usr/bin/m4 accepts --gnu... yes
```

Käännetään paketti:

```
make
```

```
root:/sources/libtool-2.4.6# make
GEN      libtoolize
make all-recursive
make[1]: Entering directory '/sources/libtool-2.4.6'
Making all in .
make[2]: Entering directory '/sources/libtool-2.4.6'
CC      libltdl/loaders/libltdl_libltdl_la-preopen.lo
CC      libltdl/libltdl_libltdl_la-lt__alloc.lo
CC      libltdl/libltdl_libltdl_la-lt__dlloader.lo
CC      libltdl/libltdl_libltdl_la-lt__error.lo
CC      libltdl/libltdl_libltdl_la-ltdl.lo
CC      libltdl/libltdl_libltdl_la-slist.lo
CC      libltdl/loaders/dlopen.lo
CCLD    libltdl/dlopen.la
CC      libltdl/lt__strl.lo
CCLD    libltdl/libltdl.la
make[2]: Leaving directory '/sources/libtool-2.4.6'
make[1]: Leaving directory '/sources/libtool-2.4.6'
root:/sources/libtool-2.4.6#
```

Ajetaan testit (n. 11 SBU)

```
make check
```

Viiden testin tiedetään epäonnistuvan LFS-ympäristössä riippuvuussuhteista johtuen, mutta kaikki testit menevät läpi, jos ne ajetaan uudestaan automaken asentamisen jälkeen.

```
## ----- ##
## Test results. ##
## ----- ##

ERROR: 138 tests were run,
64 failed (59 expected failures).
32 tests were skipped.
## ----- ##
## testsuite.log was created. ##
## ----- ##
```

Asennetaan paketti:

```
make install
```

```
root:/sources/libtool-2.4.6# make install
make install-recursive
make[1]: Entering directory '/sources/libtool-2.4.6'
Making install in .
make[2]: Entering directory '/sources/libtool-2.4.6'
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf libtool-2.4.6
```

```
root:/sources/libtool-2.4.6# cd ..
root:/sources# rm -rf libtool-2.4.6
root:/sources#
```

2.15.30 GDBM-1.12

Käännösaika: 0,1 SBU

Asennettavat ohjelmat: *gdbm_dump, gdbm_load ja gdbmtool*

Asennettavat kirjastot: *libgdbm.so ja libgdbm_compat.so*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf gdbm-1.12.tar.gz
cd gdbm-1.12
```

```
root:/sources# tar xf gdbm-1.12.tar.gz
root:/sources# cd gdbm-1.12
root:/sources/gdbm-1.12#
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/usr \
            --disable-static \
            --enable-libgdbm-compat
```

```
root:/sources/gdbm-1.12# ./configure --prefix=/usr --disable-static --enable-libgdbm-compat
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
checking for gawk... gawk
```

Puretaan auki:

- `--enable-libgdbm-compat` → Sallii libgdm-yhteensopivuuskirjaston kääntämisen, koska jotkut paketit LFS:n ulkopuolella saattavat vaatia vanhempia DBM-rutiineja, joita se tarjoaa.

Käännetään paketti:

```
make
```

```
root:/sources/gdbm-1.12# make
make all-recursive
make[1]: Entering directory '/sources/gdbm-1.12'
Making all in po
make[2]: Entering directory '/sources/gdbm-1.12/po'
```

Ajetaan testit:

```
make check
```

```
## ----- ##
## Test results. ##
## ----- ##

All 22 tests were successful.
```

Asennetaan paketti:

```
make install
```

```
root:/sources/gdbm-1.12# make install
Making install in po
make[1]: Entering directory '/sources/gdbm-1.12/po'
installing de.gmo as /usr/share/locale/de/LC_MESSAGES/gdbm.mo
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf gdbm-1.12
```

```
root:/sources/gdbm-1.12# cd ..
root:/sources# rm -rf gdbm-1.12
root:/sources#
```

2.15.31 Gperf-3.0.4

Käännösaika: alle 0,1 SBU

Asennettavat ohjelmat: gperf

Asennettavat hakemistot: /usr/share/doc/gperf-3.0.4

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf gperf-3.0.4.tar.gz
cd gperf-3.0.4
```

```
root:/sources# tar xf gperf-3.0.4.tar.gz
root:/sources# cd gperf-3.0.4
root:/sources/gperf-3.0.4#
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/usr --docdir=/usr/share/doc/gperf-
3.0.4
```

```
root:/sources/gperf-3.0.4# ./configure --prefix=/usr --docdir=/usr/share/doc/gperf-3.0
.4
checking whether make sets $(MAKE)... yes
configure: creating ./config.status
config.status: creating Makefile
... configuring in lib (/sources/gperf-3.0.4/lib)
```

Käännetään paketti:

```
make
```

```
root:/sources/gperf-3.0.4# make
cd lib; make all
make[1]: Entering directory '/sources/gperf-3.0.4/lib'
gcc -g -O2 -I. -c ./getopt.c
gcc -g -O2 -I. -c ./getopt1.c
```

Testien tiedetään epäonnistuvan, jos yhtä aikaa ajetaan monia testejä (-j -asetus suurempi kuin 1). Ajetaan testit:

```
make -j1 check
```

```
root:/sources/gperf-3.0.4# make -j1 check
cd lib; make check
make[1]: Entering directory '/sources/gperf-3.0.4/lib'
make[1]: Nothing to be done for 'check'.
make[1]: Leaving directory '/sources/gperf-3.0.4/lib'
cd src; make check
make[1]: Entering directory '/sources/gperf-3.0.4/src'
```

Asennetaan paketti:

```
make install
```

```
root:/sources/gperf-3.0.4# make install
cd lib; make install
make[1]: Entering directory '/sources/gperf-3.0.4/lib'
make[1]: Nothing to be done for 'install'.
make[1]: Leaving directory '/sources/gperf-3.0.4/lib'
cd src; make install
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf gperf-3.0.4
```

```
root:/sources/gperf-3.0.4# cd ..
root:/sources# rm -rf gperf-3.0.4
root:/sources#
```

2.15.32 Expat-2.2.0

Käännösaika: alle 0,1 SBU

Asennettavat ohjelmat: xmlwf

Asennettavat kirjastot: libexpat.so

Asennettavat hakemistot: /usr/share/doc/expat-2.2.0

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf expat-2.2.0.tar.bz2
cd expat-2.2.0
```

```
root:/sources# tar xf expat-2.2.0.tar.bz2
root:/sources# cd expat-2.2.0
root:/sources/expat-2.2.0#
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/usr --disable-static
```

```
root:/sources/expat-2.2.0# ./configure --prefix=/usr --disable-static
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking how to print strings... printf
```

Käännetään paketti:

```
make
```

```
root:/sources/expat-2.2.0# make
/bin/sh ./libtool --verbose --mode=compile gcc -I./lib -I. -g -O2 -Wall -Wmissing-prototypes -Wstrict-prototypes -fexceptions -DHAVE_EXPAT_CONFIG_H -o lib/xmlparse.lo -c lib/xmlparse.c
libtool: compile: gcc -I./lib -I. -g -O2 -Wall -Wmissing-prototypes -Wstrict-prototypes
```

Ajetaan testit:

```
make check
```

```
Expat version: expat_2.2.0
100%: Checks: 54, Failed: 0
root:/sources/expat-2.2.0#
```

Asennetaan paketti:

```
make install
```

```
root:/sources/expat-2.2.0# make install
/bin/sh ./conftools/mkinstalldirs /usr/lib /usr/include /usr/lib/pkgconfig
/bin/sh ./libtool --mode=install /tools/bin/install -c libexpat.la /usr/lib/libexpat.l
a
libtool: install: /tools/bin/install -c libs/libexpat.so.1.6.2 /usr/lib/libexpat.so.1
```

Halutessasi voit asentaa dokumentaation:

```
install -v -dm755 /usr/share/doc/expat-2.2.0
install -v -m644 doc/*.{html,png,css} /usr/share/doc/expat-
2.2.0
```

```
root:/sources/expat-2.2.0# install -v -dm755 /usr/share/doc/expat-2.2.0
install: creating directory '/usr/share/doc/expat-2.2.0'
root:/sources/expat-2.2.0# install -v -m644 doc/*.{html,png,css} /usr/share/doc/expat-
2.2.0
'doc/reference.html' -> '/usr/share/doc/expat-2.2.0/reference.html'
'doc/expat.png' -> '/usr/share/doc/expat-2.2.0/expat.png'
'doc/valid-xhtml10.png' -> '/usr/share/doc/expat-2.2.0/valid-xhtml10.png'
'doc/style.css' -> '/usr/share/doc/expat-2.2.0/style.css'
root:/sources/expat-2.2.0#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf expat-2.2.0
```

```
root:/sources/expat-2.2.0# cd ..
root:/sources# rm -rf expat-2.2.0
root:/sources#
```

2.15.33 Inetutils-1.9.4

Käännösaika: 0,4 SBU

Asennettavat ohjelmat: *dnsdomainname, ftp, ifconfig, hostname, ping, ping6, talk, telnet, tftp ja traceroute*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf inetutils-1.9.4.tar.xz
cd inetutils-1.9.4
```

```
root:/sources# tar xf inetutils-1.9.4.tar.xz
root:/sources# cd inetutils-1.9.4
root:/sources/inetutils-1.9.4#
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/usr \
            --localstatedir=/var \
            --disable-logger \
            --disable-whois \
            --disable-rcp \
            --disable-rexec \
            --disable-rlogin \
            --disable-rsh \
            --disable-servers
```

Puretaan auki:

- `--disable-logger` → Estää Inetutilsia asentamasta logger-ohjelmaa, koska Util-linux asentaa tuoreemman version.
- `--disable-whois` → Estää Inetutilsin whois-clientin kääntämisen, koska se on vanhentunut. Ohjeet parempaan whois-clienttiin löytyvät [BLFS](#)-kirjasta.

- `--disable-r*` → Nämä parametrit estävät vanhentuneiden ohjelmien kääntämisen, joita ei pitäisi enää käyttää tietoturvaongelmien vuoksi.
- `--disable-servers` → Estää monien nettiservereiden asentamisen, joiden arvellaan olevan epäsoivia LFS:n kanssa.

Käännetään paketti:

```
make
```

```
root:/sources/inetutils-1.9.4# make
echo 1.9.4 > .version-t && mv .version-t .version
make all-recursive
make[1]: Entering directory '/sources/inetutils-1.9.4'
Making all in lib
```

Ajetaan testit:

```
make check
```

```
=====
Testsuite summary for GNU inetutils 1.9.4
=====
# TOTAL: 10
# PASS: 10
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan paketti:

```
make install
```

```
root:/sources/inetutils-1.9.4# make install
make install-recursive
make[1]: Entering directory '/sources/inetutils-1.9.4'
Making install in lib
make[2]: Entering directory '/sources/inetutils-1.9.4/lib'
make install-recursive
```

Siirretään jotkin ohjelmat, jotta ne ovat saatavilla, jos /usr ei ole saatavissa.

```
mv -v /usr/bin/{hostname,ping,ping6,traceroute} /bin
mv -v /usr/bin/ifconfig /sbin
```

```
root:/sources/inetutils-1.9.4# mv -v /usr/bin/{hostname,ping,ping6,traceroute} /bin
'/usr/bin/hostname' -> '/bin/hostname'
'/usr/bin/ping' -> '/bin/ping'
'/usr/bin/ping6' -> '/bin/ping6'
'/usr/bin/traceroute' -> '/bin/traceroute'
root:/sources/inetutils-1.9.4#
```

```
root:/sources/inetutils-1.9.4# mv -v /usr/bin/ifconfig /sbin
'/usr/bin/ifconfig' -> '/sbin/ifconfig'
root:/sources/inetutils-1.9.4#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf inetutils-1.9.4
```

```
root:/sources/inetutils-1.9.4# cd ..
root:/sources# rm -rf inetutils-1.9.4
root:/sources#
```

2.15.34 Perl-5.24.1

Käännösaika: 5,9 SBU

Asennettavat ohjelmat: *c2ph, corelist, cpan, enc2xs, encguess, h2ph, h2xs, instmodsh, json_pp, libnetcfg, perl, perl5.24.1 (kova linkki perliin), perlbug, perldoc, perlvp, perlthanks (kova linkki perlbugiin), piconv, pl2pm, pod2html, pod2man, pod2text, pod2usage, podchecker, podselect, prove, pstruct (kova linkki c2ph), ptar, ptardiff, ptargrep, shasum, splain, xsubpp ja zipdetails*

Asennettavat kirjastot: *Liikaa listattavaksi tässä*

Asennettavat hakemistot: */usr/lib/perl5*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf perl-5.24.1.tar.bz2
cd perl-5.24.1
```

```
root:/sources# tar xf perl-5.24.1.tar.bz2
root:/sources# cd perl-5.24.1
root:/sources/perl-5.24.1#
```

Luodaan `/etc/hosts` -tiedosto, johon yksi Perlin konfiguraatiotiedosto viittaa.

```
echo "127.0.0.1 localhost $(hostname)" > /etc/hosts
```

```
root:/sources/perl-5.24.1# echo "127.0.0.1 localhost $(hostname)" > /etc/hosts
root:/sources/perl-5.24.1#
```

Perlin tämä versio kääntää `Compress::Raw::Zlib` ja `Compress::Raw::BZip2` -moduulit. Oletuksena Perl käyttää `sources:n` sisäistä kopioita kääntämistä varten. Ajetaan seuraava komento, jotta Perl käyttää järjestelmään asennettuja kirjastoja:

```
export BUILD_ZLIB=False
export BUILD_BZIP2=0
```

```
root:/sources/perl-5.24.1# export BUILD_ZLIB=False
root:/sources/perl-5.24.1# export BUILD_BZIP2=0
root:/sources/perl-5.24.1#
```

HUOM! Jos sinulla oli aiemmin ongelmia zlibin kanssa, aja sen sijaan seuraavat komennot:

```
export BUILD_ZLIB=True
export BUILD_BZIP2=0
```

Halutessasi täyden kontrollin siihen, miten Perl kasataan, voit poistaa “-des” -asetukset alla olevasta komennosta. Vaihtoehtoisesti voit käyttää alla olevaa komentoa tarkalleen siinä muodossa kuin se on käyttäkäsesi oletuksia, jotka Perl automaattisesti tunnistaa:

```
sh Configure -des -Dprefix=/usr \
-Dvendorprefix=/usr \
-Dman1dir=/usr/share/man/man1 \
-Dman3dir=/usr/share/man/man3 \
-Dpager="/usr/bin/less -isR" \
-Duseshrplib
```

```
root:/sources/perl-5.24.1# sh Configure -des -Dprefix=/usr -Dvendorprefix=/usr -Dman1dir=/usr/share/man/man1 -Dman3dir=/usr/share/man/man3 -Dpager="/usr/bin/less -isR" -Duseshrplib
First let's make sure your kit is complete. Checking...
Locating common programs...
```

Puretaan auki:

- -Dvendorprefix=/usr → Tämä varmistaa, että perl tietää miten kertoa paketeille mihin niiden pitäisi asentaa niiden perl-moduulit.
- -Dpager="/usr/bin/less -isR" → Käytetään less:iä more:n sijaan.
- -Dman1dir=/usr/share/man/man1
Dman3dir=/usr/share/man/man3 → Koska Groffia, ei ole asennettu

vielä, Configure luulee, että emme halua Perlin man-sivuja. Nämä asetukset syrjäyttävät sen.

- `-Duseshrplib` → Käännetään jaettu libperl, jota jotkin perlin moduulit tarvitsevat.

Käännetään paketti:

```
make
```

```
root:/sources/perl-5.24.1# make
echo @`sh` cflags "optimize='-O2'" opmini.o` -fPIC -DPERL_IS_MINIPERL -DPERL_EXTERNAL_
GLOB opmini.c
gcc -c -DPERL_CORE -fwrapv -fno-strict-aliasing -pipe -fstack-protector-strong -I/usr/
local/include -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64 -std=c89 -O2 -Wall -Werror=de
```

Ajetaan testit (n. 2,5 SBU):

```
make -k test
```

HUOM! Monet Zlibiin liittyvät testit epäonnistuvat, koska käytetään järjestelmän versiota zlib:stä sen sijaan että käytettäisiin sisäistä versiota.

```
Failed 9 tests out of 2233, 99.60% okay.
../cpan/IO-Socket-IP/t/01local-client-v4.t
../cpan/IO-Socket-IP/t/02local-server-v4.t
../cpan/IO-Socket-IP/t/03local-cross-v4.t
../cpan/IO-Socket-IP/t/11sockopts.t
../cpan/IO-Socket-IP/t/18fdopen.t
../cpan/IO-Socket-IP/t/20subclass.t
../cpan/IO-Socket-IP/t/21as-inet.t
../cpan/IO-Socket-IP/t/22timeout.t
../cpan/IO-Socket-IP/t/30nonblocking-connect.t
```

Asennetaan paketti ja siivotaan:

```
make install
```

```
unset BUILD_ZLIB BUILD_BZIP2
```

```
root:/sources/perl-5.24.1# unset BUILD_ZLIB BUILD_BZIP2
root:/sources/perl-5.24.1#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf perl-5.24.1
```

```
root:/sources/perl-5.24.1# cd ..  
root:/sources# rm -rf perl-5.24.1  
root:/sources# █
```

2.15.35 XML::Parser-2.44

Käännösaika: alle 0,1 SBU

Asennettava moduuli: Expat.so

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf XML-Parser-2.44.tar.gz
cd XML-Parser-2.44
```

```
root:/sources# tar xf XML-Parser-2.44.tar.gz
root:/sources# cd XML-Parser-2.44
root:/sources/XML-Parser-2.44#
```

Valmistellaan XML::Parser kääntämistä varten:

```
perl Makefile.PL
```

```
root:/sources/XML-Parser-2.44# perl Makefile.PL
Checking if your kit is complete...
Looks good
Warning: prerequisite LWP::UserAgent 0 not found.
Generating a Unix-style Makefile
Writing Makefile for XML::Parser::Expat
Writing MYMETA.yml and MYMETA.json
Generating a Unix-style Makefile
Writing Makefile for XML::Parser
Writing MYMETA.yml and MYMETA.json
root:/sources/XML-Parser-2.44#
```

Käännetään paketti:

```
make
```

```
root:/sources/XML-Parser-2.44# make
cp Parser/Encodings/x-euc-jp-jisx0221.enc blib/lib/XML/Parser/Encodings/x-euc-jp-jisx0221.enc
cp Parser/Encodings/Japanese_Encodings.msg blib/lib/XML/Parser/Encodings/Japanese_Encodings.msg
root:/sources/XML-Parser-2.44#
```

Ajetaan testit:

```
make test
```

```
All tests successful.  
Files=15, Tests=141, 1 wallclock secs ( 0.08 usr 0.02 sys + 0.56 cusr 0.05 csys =  
0.71 CPU)  
Result: PASS  
root:/sources/XML-Parser-2.44#
```

Asennetaan paketti:

```
make install
```

```
root:/sources/XML-Parser-2.44# make install  
make[1]: Entering directory '/sources/XML-Parser-2.44/Expat'  
Manifesting 1 pod document  
make[1]: Leaving directory '/sources/XML-Parser-2.44/Expat'  
Manifesting 6 pod documents  
Files found in blib/arch: installing files in blib/lib into architecture dependent lib
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf XML-Parser-2.44
```

```
root:/sources/XML-Parser-2.44# cd ..  
root:/sources# rm -rf XML-Parser-2.44  
root:/sources#
```

2.15.36 Intltool-0.51.0

Käännösaika: alle 0,1 SBU

Asennettavat ohjelmat: intltool-extract, intltool-merge, intltool-prepare, intltool-update ja intltoolize

Asennettavat hakemistot: /usr/share/doc/intltool-0.51.0 ja /usr/share/intltool

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf intltool-0.51.0.tar.gz
cd intltool-0.51.0
```

```
root:/sources# tar xf intltool-0.51.0.tar.gz
root:/sources# cd intltool-0.51.0
root:/sources/intltool-0.51.0#
```

Korjataan varoitus, jonka aiheuttaa perl-5.22 ja myöhemmät versiot:

```
sed -i 's:\\\\${:\\\\$\\{: intltool-update.in
```

```
root:/sources/intltool-0.51.0# sed -i 's:\\\\${:\\\\$\\{: intltool-update.in
root:/sources/intltool-0.51.0#
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/usr
```

```

root:/sources/intltool-0.51.0# ./configure --prefix=/usr
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking for perl... /usr/bin/perl
checking for perl >= 5.8.1... 5.24.1
checking for XML::Parser... ok
checking for bzip2... no
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating Makefile
config.status: creating doc/Makefile
config.status: creating intltoolize
config.status: creating tests/Makefile
config.status: creating tests/cases/Makefile
config.status: creating tests/results/Makefile
config.status: creating tests/selftest.pl
root:/sources/intltool-0.51.0# █

```

Käännetään paketti:

```
make
```

```

root:/sources/intltool-0.51.0# make
Making all in tests
make[1]: Entering directory '/sources/intltool-0.51.0/tests'
Making all in cases
make[2]: Entering directory '/sources/intltool-0.51.0/tests/cases'
make[2]: Nothing to be done for 'all'.
make[2]: Leaving directory '/sources/intltool-0.51.0/tests/cases'

```

Ajetaan testit:

```
make check
```

```

=====
Testsuite summary for intltool 0.51.0
=====
# TOTAL: 1
# PASS: 1
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====

```

Asennetaan paketti:

```
make install  
install -v -Dm644 doc/I18N-HOWTO /usr/share/doc/intltool-  
0.51.0/I18N-HOWTO
```

```
root:/sources/intltool-0.51.0# make install  
Making install in tests  
make[1]: Entering directory '/sources/intltool-0.51.0/tests'  
Making install in cases  
make[2]: Entering directory '/sources/intltool-0.51.0/tests/cases'
```

```
root:/sources/intltool-0.51.0# install -v -Dm644 doc/I18N-HOWTO /usr/share/doc/intltool-  
0.51.0/I18N-HOWTO  
install: creating directory '/usr/share/doc/intltool-0.51.0'  
'doc/I18N-HOWTO' -> '/usr/share/doc/intltool-0.51.0/I18N-HOWTO'  
root:/sources/intltool-0.51.0#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf intltool-0.51.0
```

```
root:/sources/intltool-0.51.0# cd ..  
root:/sources# rm -rf intltool-0.51.0  
root:/sources#
```

2.15.37 Autoconf-2.69

Käännösaika: alle 0.1 SBU (n. 3,5 SBU testien kanssa)

Asennettavat ohjelmat: *autoconf, autoheader, autom4te, autoreconf, autoscan, autoupdate ja ifnames*

Asennettavat hakemistot: */usr/share/autoconf*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf autoconf-2.69.tar.xz
cd autoconf-2.69
```

```
root:/sources# tar xf autoconf-2.69.tar.xz
root:/sources# cd autoconf-2.69
root:/sources/autoconf-2.69#
```

Valmistellaan Autoconf kääntämistä varten:

```
./configure --prefix=/usr
```

```
root:/sources/autoconf-2.69# ./configure --prefix=/usr
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
checking for gawk... gawk
```

Käännetään paketti:

```
make
```

```
root:/sources/autoconf-2.69# make
make all-recursive
make[1]: Entering directory '/sources/autoconf-2.69'
Making all in bin
make[2]: Entering directory '/sources/autoconf-2.69/bin'
rm -f autom4te autom4te.tmp
```

Ajetaan testit (saattavat viedä jonkin aikaa):

```
make check
```

Monet testit, jotka käyttävät Automakea, ohitetaan. Testit voi ajaa uudelleen automaken asentamisen jälkeen, jos haluaa. Kahden testin tiedetään epäonnistuvan johtuen muutoksista libtool-2.4.3 ja myöhemmissä versioissa.

```
## ----- ##
## Test results. ##
## ----- ##

ERROR: 450 tests were run,
6 failed (4 expected failures).
53 tests were skipped.
## ----- ##
## testsuite.log was created. ##
## ----- ##
```

```
501: Libtool                                FAILED (foreign.at:61)
502: shtool                                  ok
Autoscan.
503: autoscan                               FAILED (autoscan.at:44)
```

Asennetaan paketti:

```
make install
```

```
root:/sources/autoconf-2.69# make install
make install-recursive
make[1]: Entering directory '/sources/autoconf-2.69'
Making install in bin
make[2]: Entering directory '/sources/autoconf-2.69/bin'
make[3]: Entering directory '/sources/autoconf-2.69/bin'
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf autoconf-2.69
```

```
root:/sources/autoconf-2.69# cd ..
root:/sources# rm -rf autoconf-2.69
root:/sources#
```

2.15.38 3.6.38 Automake-1.15

Käännösaika: alle 0,1 SBU (n. 7,5 SBU testien kanssa)

Asennettavat ohjelmat: *aclocal*, *aclocal-1.15* (kovalinkattu *aclocal*:in kanssa), *automake* ja *automake-1.15* (kovalinkattu *automake*:n kanssa)

Asennettavat hakemistot: */usr/share/aclocal-1.15*, */usr/share/automake-1.15* ja */usr/share/doc/automake-1.15*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf automake-1.15.tar.xz
cd automake-1.15
```

```
root:/sources# tar xf automake-1.15.tar.xz
root:/sources# cd automake-1.15
root:/sources/automake-1.15#
```

Korjataan varoitus, jonka aiheuttaa perl-5.22 ja myöhemmät versiot.

```
sed -i 's:/\\\${: /\\\$\\{: ' bin/automake.in
```

```
root:/sources/automake-1.15# sed -i 's:/\\\${: /\\\$\\{: ' bin/automake.in
root:/sources/automake-1.15#
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/usr --docdir=/usr/share/doc/automake-1.15
```

```
root:/sources/automake-1.15# ./configure --prefix=/usr --docdir=/usr/share/doc/automake-1.15
checking whether make supports nested variables... yes
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
```

Käännetään paketti:

```
make
```

```

root:/sources/automake-1.15# make
GEN      bin/automake
GEN      bin/aclocal
GEN      t/ax/shell-no-trail-bslash
GEN      t/ax/cc-no-c-o
GEN      runtest
GEN      doc/aclocal.1
GEN      doc/automake.1
GEN      lib/Automake/Config.pm
GEN      doc/aclocal-1.15.1
GEN      doc/automake-1.15.1
GEN      t/ax/test-defs.sh
root:/sources/automake-1.15# █

```

Muutammat testit linkkaavat virheellisesti flex-kirjaston väärään versioon, joten kierämme ongelman tilapäisesti. Lisäksi `-j4` -asetuksen käyttäminen nopeuttaa testejä jopa järjestelmillä, joissa on vain yksi prosessori. Ajetaan testit (kestää jonkin aikaa):

```

sed -i "s:./configure:LEXLIB=/usr/lib/libfl.a &:" t/lex-
{clean,depend}-cxx.sh
make -j4 check

```

Neljän testin tiedetään epäonnistuvan.

```

root:/sources/automake-1.15# sed -i "s:./configure:LEXLIB=/usr/lib/libfl.a &:" t/lex-
clean,depend}-cxx.sh
root:/sources/automake-1.15# █

```

```

=====
Testsuite summary for GNU Automake 1.15
=====
# TOTAL: 2899
# PASS: 2695
# SKIP: 160
# XFAIL: 41
# FAIL: 3
# XPASS: 0
# ERROR: 0
=====
See ./test-suite.log
Please report to bug-automake@gnu.org
=====

```

Asennetaan paketti:

```
make install
```

```
root:/sources/automake-1.15# make install
make[1]: Entering directory '/sources/automake-1.15'
  /tools/bin/mkdir -p '/usr/bin'
  /tools/bin/install -c bin/automake bin/aclocal '/usr/bin'
make install-exec-hook
make[2]: Entering directory '/sources/automake-1.15'
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf automake-1.15
```

```
root:/sources/automake-1.15# cd ..
root:/sources# rm -rf automake-1.15
root:/sources#
```

2.15.39 Xz-5.2.3

Käännösaika: 0,2 SBU

Asennettavat ohjelmat: *lzcat* (linkki [xz:aan](#)), *lzcmp* (linkki [xzdifff:iin](#)), *lzdifff* (linkki [xzdifff:iin](#)), *lzegrep* (linkki [xzgrep:iin](#)), *lzfgrep* (linkki [xzgrep:iin](#)), *lzgrep* (linkki [xzgrep:iin](#)), *lzless* (linkki [xzless:iin](#)), *lzma* (linkki [xz:aan](#)), *lzmdec*, *lzmainfo*, *lzmore* (linkki [xzmore:en](#)), *unlzma* (linkki [xz:aan](#)), *unxz* (linkki [xz:aan](#)), *xz*, *xzcat* (linkki [xz:aan](#)), *xzcmp* (linkki [xzdifff:iin](#)), *xzdec*, *xzdifff*, *xzegrep* (linkki [xzgrep:iin](#)), *xzfgrep* (linkki [xzgrep:iin](#)), *xzgrep*, *xzless* ja *xzmore*

Asennettavat kirjastot: *liblzma.so*

Asennettavat hakemistot: */usr/include/lzma* ja */usr/share/doc/xz-5.2.3*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf xz-5.2.3.tar.xz
cd xz-5.2.3
```

```
root:/sources# tar xf xz-5.2.3.tar.xz
root:/sources# cd xz-5.2.3
root:/sources/xz-5.2.3#
```

Valmistellaan Xz kääntämistä varten:

```
./configure --prefix=/usr \
            --disable-static \
            --docdir=/usr/share/doc/xz-5.2.3
```

```
root:/sources/xz-5.2.3# ./configure --prefix=/usr --disable-static --docdir=/usr/share
/doc/xz-5.2.3
XZ Utils 5.2.3
System type:
checking build system type... x86_64-unknown-linux-gnu
```

Käännetään paketti:

```
make
```

Ajetaan testit:

```
make check
```

```
=====
All 9 tests passed
=====
```

Asennetaan paketti ja varmistetaan, että kaikki oleelliset tiedostot ovat oikeassa hakemistossa:

```
make install
mv -v /usr/bin/{lzma,unlzma,lzcat,xz,unxz,xzcat} /bin
mv -v /usr/lib/liblzma.so.* /lib
ln -svf ../../lib/$(readlink /usr/lib/liblzma.so)
/usr/lib/liblzma.so
```

```
root:/sources/xz-5.2.3# make install
Making install in src
make[1]: Entering directory '/sources/xz-5.2.3/src'
Making install in liblzma
make[2]: Entering directory '/sources/xz-5.2.3/src/liblzma'
Making install in api
make[3]: Entering directory '/sources/xz-5.2.3/src/liblzma/api'
```

```
root:/sources/xz-5.2.3# mv -v /usr/bin/{lzma,unlzma,lzcat,xz,unxz,xzcat} /bin
'/usr/bin/lzma' -> '/bin/lzma'
'/usr/bin/unlzma' -> '/bin/unlzma'
'/usr/bin/lzcat' -> '/bin/lzcat'
'/usr/bin/xz' -> '/bin/xz'
'/usr/bin/unxz' -> '/bin/unxz'
'/usr/bin/xzcat' -> '/bin/xzcat'
root:/sources/xz-5.2.3#
```

```
root:/sources/xz-5.2.3# mv -v /usr/lib/liblzma.so.* /lib
'/usr/lib/liblzma.so.5' -> '/lib/liblzma.so.5'
'/usr/lib/liblzma.so.5.2.3' -> '/lib/liblzma.so.5.2.3'
root:/sources/xz-5.2.3#
```

```
root:/sources/xz-5.2.3# ln -svf ../../lib/$(readlink /usr/lib/liblzma.so) /usr/lib/liblzma.so
'/usr/lib/liblzma.so' -> '/../../lib/liblzma.so.5.2.3'
root:/sources/xz-5.2.3#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf xz-5.2.3
```

```
root:/sources/xz-5.2.3# cd ..  
root:/sources# rm -rf xz-5.2.3  
root:/sources# █
```

2.15.40 Kmod-23

Käännösaika: 0,1 SBU

Asennettavat ohjelmat: *depmod* (linkki *kmod:iin*), *insmod* (linkki *kmod:iin*), *kmod*, *lsmod* (linkki *kmod:iin*), *modinfo* (linkki *kmod:iin*), *modprobe* (linkki *kmod:iin*) ja *rmmod* (linkki *kmod:iin*)

Asennettavat kirjastot: *libkmod.so*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf kmod-23.tar.xz
cd kmod-23
```

```
root:/sources# tar xf kmod-23.tar.xz
root:/sources# cd kmod-23
root:/sources/kmod-23#
```

Valmistellaan Kmod kääntämistä varten. Jos sinulla oli aiemmin ongelmia Zlibin kanssa, poista viimeinen `--with-zlib` -parametri.

```
./configure --prefix=/usr \
            --bindir=/bin \
            --sysconfdir=/etc \
            --with-rootlibdir=/lib \
            --with-xz \
            --with-zlib
```

Puretaan auki:

- `--with-xz`, `--with-zlib` → Nämä asetukset sallivat Kmodin käsitellä kompressoituja kernel-moduuleja.
- `--with-rootlibdir=/lib` → Tämä asetus varmistaa, että erilaiset kirjastoihin liittyvät tiedostot ovat oikeassa hakemistossa.

```
root:/sources/kmod-23# ./configure --prefix=/usr --bindir=/bin --sysconfdir=/etc --with-rootlibdir=/lib --with-xz
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
```

Käännetään paketti:

```
make
```

```
root:/sources/kmod-23# make
make --no-print-directory all-recursive
Making all in .
  CC      libkmod/libkmod.lo
  CC      libkmod/libkmod-list.lo
  CC      libkmod/libkmod-config.lo
  CC      libkmod/libkmod-index.lo
  CC      libkmod/libkmod-module.lo
```

Tämän paketin mukana ei tule testejä, jotka voitaisiin ajaa LFS:n chroot-ympäristössä.

Asennetaan paketti:

```
make install
```

```
root:/sources/kmod-23# make install
Making install in .
  /tools/bin/mkdir -p '/usr/lib'
  ./doltlibtool --mode=install /tools/bin/install -c libkmod/libkmod.la '/usr/lib'
libtool: install: /tools/bin/install -c libkmod/libkmod.so.2.3.1 /usr/lib/libkmod.so.2.3.1
```

Luodaan symbolisia linkkejä:

```
for target in depmod insmod lsmod modinfo modprobe rmmod;
do
    ln -sfv ../bin/kmod /sbin/$target
done
```

```
root:/sources/kmod-23# for target in depmod insmod lsmod modinfo modprobe rmmod; do
>   ln -sfv ../bin/kmod /sbin/$target
> done
'/sbin/depmod' -> '../bin/kmod'
'/sbin/insmod' -> '../bin/kmod'
'/sbin/lsmod' -> '../bin/kmod'
'/sbin/modinfo' -> '../bin/kmod'
'/sbin/modprobe' -> '../bin/kmod'
'/sbin/rmmod' -> '../bin/kmod'
root:/sources/kmod-23#
```

```
ln -sfv kmod /bin/lsmod
```

```
root:/sources/kmod-23# ln -sfv kmod /bin/lsmmod  
'/bin/lsmmod' -> 'kmod'  
root:/sources/kmod-23#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf kmod-23
```

```
root:/sources/kmod-23# cd ..  
root:/sources# rm -rf kmod-23  
root:/sources#
```

2.15.41 Gettext-0.19.8.1

Käännösaika: 2,9 SBU

Asennettavat ohjelmat:

autopoint, envsubst, gettext, gettext.sh, gettextize, msgattrib, msgcat, msgcmp, msgcomm, msgconv, msgen, msgexec, msgfilter, msgfmt, msggrep, msginit, msgmerge, msgunfmt, msguniq, ngettext, recode-sr-latin ja xgettext

Asennettavat kirjastot:

libasprintf.so, libgettextlib.so, libgettextpo.so, libgettextsrc.so, ja preloadable_libintl.so

Asennettavat hakemistot:

/usr/lib/gettext, /usr/share/doc/gettext-0.19.8.1 ja /usr/share/gettext

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf gettext-0.19.8.1.tar.xz
cd gettext-0.19.8.1
```

```
root:/sources# tar xf gettext-0.19.8.1.tar.xz
root:/sources# cd gettext-0.19.8.1
root:/sources/gettext-0.19.8.1#
```

Joillakin koneilla parit testijutut saattavat loopata loputtomiin, joten korjataan ongelma:

```
sed -i '/^TESTS =/d' gettext-runtime/tests/Makefile.in &&
sed -i 's/test-lock..EXEEXT.//' gettext-tools/gnulib-
tests/Makefile.in
```

```
root:/sources/gettext-0.19.8.1# sed -i '/^TESTS =/d' gettext-runtime/tests/Makefile.in
&& sed -i 's/test-lock..EXEEXT.//' gettext-tools/gnulib-tests/Makefile.in
root:/sources/gettext-0.19.8.1#
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/usr \
            --disable-static \
            --docdir=/usr/share/doc/gettext-0.19.8.1
```

Käännetään paketti:

```
make
```

Ajetaan testit (n. 3 SBU):

```
make check
```

```
=====
Testsuite summary for gettext-tools 0.19.8.1
=====
# TOTAL: 197
# PASS: 187
# SKIP: 10
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan paketti:

```
make install
chmod -v 0755 /usr/lib/preloadable_libintl.so
```

```
root:/sources/gettext-0.19.8.1# chmod -v 0755 /usr/lib/preloadable_libintl.so
mode of '/usr/lib/preloadable_libintl.so' changed from 0644 (rw-r--r--) to 0755 (rwxr-xr-x)
root:/sources/gettext-0.19.8.1#
```

Poistetaan purettu hakemisto:

```
cd ..
rm -rf gettext-0.19.8.1
```

```
root:/sources/gettext-0.19.8.1# cd ..
root:/sources# rm -rf gettext-0.19.8.1
root:/sources#
```

2.15.42 Procps-ng-3.3.12

Käännösaika: 0,1 SBU

Asennettavat ohjelmat: *free, pgrep, pidof, pkill, pmap, ps, pwdx, slabtop, sysctl, tload, top, uptime, vmstat, w ja watch*

Asennettavat kirjastot: *libprocps.so*

Asennettavat hakemistot: */usr/include/proc ja /usr/share/doc/procps-ng-3.3.12*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf procps-ng-3.3.12.tar.xz
cd procps-ng-3.3.12
```

```
root:/sources# tar xf procps-ng-3.3.12.tar.xz
root:/sources# cd procps-ng-3.3.12
root:/sources/procps-ng-3.3.12#
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/usr \
            --exec-prefix= \
            --libdir=/usr/lib \
            --docdir=/usr/share/doc/procps-ng-3.3.12 \
            --disable-static \
            --disable-kill
```

Puretaan auki:

- `--disable-kill` → Estää kääntämästä kill-komentoa, jonka Util-linux asentaa.

```
root:/sources/procps-ng-3.3.12# ./configure --prefix=/usr --exec-prefix= --libdir=/usr
/lib --docdir=/usr/share/doc/procps-ng-3.3.12 --disable-static --disable-kill
checking for a BSD-compatible install... /tools/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /tools/bin/mkdir -p
```

Käännetään paketti:

```
make
```

```
root:/sources/procps-ng-3.3.12# make
make all-recursive
make[1]: Entering directory '/sources/procps-ng-3.3.12'
Making all in include
make[2]: Entering directory '/sources/procps-ng-3.3.12/include'
```

Testejä pitää muokata hieman LFS:ää varten. Poistetaan testi, joka epäonnistuu.

Ajetaan testit:

```
sed -i -r 's|(pmap_initname)\\$|\\1|' test-
suite/pmap.test/pmap.exp
make check
```

Yksi ps-testi saattaa epäonnistua, mutta sekin menee läpi, jos testit ajaa uudelleen tämän kappaleen lopussa.

```
root:/sources/procps-ng-3.3.12# sed -i -r 's|(pmap_initname)\\$|\\1|' testsuite/pmap.t
est/pmap.exp
root:/sources/procps-ng-3.3.12# █
```

```
=====
Testsuite summary for procps-ng 3.3.12
=====
# TOTAL: 1
# PASS: 1
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan paketti:

```
make install
```

```
root:/sources/procps-ng-3.3.12# make install
make install-recursive
make[1]: Entering directory '/sources/procps-ng-3.3.12'
Making install in include
make[2]: Entering directory '/sources/procps-ng-3.3.12/include'
make[3]: Entering directory '/sources/procps-ng-3.3.12/include'
```

Siirretään tarpeelliset kirjastot paikkaan, josta ne löytyvät, vaikka `/usr` ei ole liitetty.

```
mv -v /usr/lib/libprocps.so.* /lib
ln -sfv ../../lib/$(readlink /usr/lib/libprocps.so)
/usr/lib/libprocps.so
```

```
root:/sources/procps-ng-3.3.12# mv -v /usr/lib/libprocps.so.* /lib
'/usr/lib/libprocps.so.6' -> '/lib/libprocps.so.6'
'/usr/lib/libprocps.so.6.0.0' -> '/lib/libprocps.so.6.0.0'
root:/sources/procps-ng-3.3.12#
```

```
root:/sources/procps-ng-3.3.12# ln -sfv ../../lib/$(readlink /usr/lib/libprocps.so) /usr
/lib/libprocps.so
'/usr/lib/libprocps.so' -> '../../lib/libprocps.so.6.0.0'
root:/sources/procps-ng-3.3.12#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf procps-ng-3.3.12
```

```
root:/sources/procps-ng-3.3.12# cd ..
root:/sources# rm -rf procps-ng-3.3.12
root:/sources#
```

2.15.43 E2fsprogs-1.43.4

Käännösaika: 4,1 SBU

Asennettavat ohjelmat: *badblocks, chatter, compile_et, debugfs, dumpe2fs, e2freefrag, e2fsck, e2image, e2label, e2undo, e4defrag, filefrag, fsck.ext2, fsck.ext3, fsck.ext4, fsck.ext4dev, logsave, lsattr, mk_cmds, mke2fs, mkfs.ext2, mkfs.ext3, mkfs.ext4, mkfs.ext4dev, mklost+found, re-size2fs ja tune2fs*

Asennettavat kirjastot: *libcom_err.so, libe2p.so, libext2fs.so ja libss.so*

Asennettavat hakemistot: */usr/include/e2p, /usr/include/et, /usr/include/ext2fs, /usr/include/ss, /usr/share/et ja /usr/share/ss*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf e2fsprogs-1.43.4.tar.gz
cd e2fsprogs-1.43.4
```

```
root:/sources# tar xf e2fsprogs-1.43.4.tar.gz
root:/sources# cd e2fsprogs-1.43.4
root:/sources/e2fsprogs-1.43.4#
```

Luodaan build-hakemisto ja siirrytään sinne:

```
mkdir -v build
cd build
```

```
root:/sources/e2fsprogs-1.43.4# mkdir -v build
mkdir: created directory 'build'
root:/sources/e2fsprogs-1.43.4# cd build/
root:/sources/e2fsprogs-1.43.4/build#
```

Valmistellaan paketti kääntämistä varten:

```
LIBS=-L/tools/lib \
CFLAGS=-I/tools/include \
PKG_CONFIG_PATH=/tools/lib/pkgconfig \
../configure --prefix=/usr \
              --bindir=/bin \
```

```
--with-root-prefix="" \
--enable-elf-shlibs \
--disable-libblkid \
--disable-libuuid \
--disable-uidd \
--disable-fsck
```

Puretaan auki:

- `PKG_CONFIG_PATH`, `LIBS`, `CFLAGS` → Sallii `e2fsprogs`:n kääntämisen käyttäen edellisessä kappaleessa asennettua `Util-linux` -pakettia.
- `--with-root-prefix=""` ja `--bindir=/bin` → Jotkin ohjelmat, kuten `e2fsck`, ovat tarpeellisia. Kun `/usr` ei ole liitetty, näiden ohjelmien pitää silti olla saatavilla. Ne kuuluvat hakemistoihin, kuten `/lib` ja `/sbin`. Jos näitä asetuksia ei anneta `E2fsprogs`:n `configure`lle, ohjelmat asennetaan `/usr` -hakemistoon.
- `--enable-elf-shlibs` → Luo jaetut kirjastot, joita jotkin tämän paketin ohjelmat käyttävät.
- `--disable-*` → Estää `E2fsprogs`ia kääntämästä ja asentamasta `libuuid` ja `libblkid` -kirjastoja, `uidd`-taustaprosessia ja `fsck-wrapper`ia, koska `Util-linux` asentaa tuoreemmat versiot.

```
root:/sources/e2fsprogs-1.43.4/build# LIBS=-L/tools/lib \
> CFLAGS=-I/tools/include \
> PKG_CONFIG_PATH=/tools/lib/pkgconfig \
> ../configure --prefix=/usr --bindir=/bin --with-root-prefix="" --enable-elf-shlibs --d
isable-libblkid --disable-libuuid --disable-uidd --disable-fsck
Generating configuration file for e2fsprogs version 1.43.4
Release date is January, 2017
checking build system type... x86_64-pc-linux-gnu
```

Käännetään paketti:

```
make
```

```

root:/sources/e2fsprogs-1.43.4/build# make
cd ./util ; make subst
make[1]: Entering directory '/sources/e2fsprogs-1.43.4/build/util'
      CREATE dirpaths.h
      CC ../../util/subst.c
      LD subst
make[1]: Leaving directory '/sources/e2fsprogs-1.43.4/build/util'

```

Testejä varten täytyy linkata joitakin kirjastoja `/tools/lib` -hakemistosta paikkaan, jonne testiohjelmat katsovat. Ajetaan testit:

```

ln -sfv /tools/lib/lib{blk,uu}id.so.1 lib
make LD_LIBRARY_PATH=/tools/lib check

```

Yksi testi yrittää allokoida 256 MB muistia. Mikäli sinulla ei ole huomattavasti enemmän RAMia, varmista, että testiä varten on sopiva swap space.

```

root:/sources/e2fsprogs-1.43.4/build# ln -sfv /tools/lib/lib{blk,uu}id.so.1 lib
'lib/libblkid.so.1' -> '/tools/lib/libblkid.so.1'
'lib/libuuid.so.1' -> '/tools/lib/libuuid.so.1'
root:/sources/e2fsprogs-1.43.4/build# █

```

```

t_ext_jnt_fat: tune2fs fat external journal: ok
329 tests succeeded      0 tests failed
Creating test_script...
make[1]: Leaving directory '/sources/e2fsprogs-1.43.4/build/tests'
root:/sources/e2fsprogs-1.43.4/build# █

```

Asennetaan binäärit, dokumentaatio ja jaetut kirjastot:

```

make install

```

```

root:/sources/e2fsprogs-1.43.4/build# make install
make[1]: Entering directory '/sources/e2fsprogs-1.43.4/build'
make[1]: 'util/subst.conf' is up to date.
make[1]: Leaving directory '/sources/e2fsprogs-1.43.4/build'
make[1]: Entering directory '/sources/e2fsprogs-1.43.4/build'

```

Asennetaan staattiset kirjastot ja headers-tiedostot:

```

make install-libs

```

```

root:/sources/e2fsprogs-1.43.4/build# make install-libs
making install in lib/et
make[1]: Entering directory '/sources/e2fsprogs-1.43.4/build/lib/et'
      MKINSTALLDIRS /lib /usr/lib

```

Tehdään staattisista kirjastoista kirjoitettavia, jotta debuggaussymbolit voidaan poistaa myöhemmin:

```
chmod -v u+w /usr/lib/{libcom_err,libe2p,libext2fs,libss}.a
```

```
< u+w /usr/lib/{libcom_err,libe2p,libext2fs,libss}.a
mode of '/usr/lib/libcom_err.a' changed from 0444 (r--r--r--) to 0644 (rw-r--r--)
mode of '/usr/lib/libe2p.a' changed from 0444 (r--r--r--) to 0644 (rw-r--r--)
mode of '/usr/lib/libext2fs.a' changed from 0444 (r--r--r--) to 0644 (rw-r--r--)
mode of '/usr/lib/libss.a' changed from 0444 (r--r--r--) to 0644 (rw-r--r--)
root:/sources/e2fsprogs-1.43.4/build#
```

Tämä paketti asentaa gzipatun `.info` -tiedoston, mutta ei päivitä järjestelmänlaajuista `dir`-tiedostoa. Puretaan tämä tiedosto ja päivitetään järjestelmän `dir`-tiedosto seuraavilla komennoilla:

```
gunzip -v /usr/share/info/libext2fs.info.gz
install-info --dir-file=/usr/share/info/dir
/usr/share/info/libext2fs.info
```

```
root:/sources/e2fsprogs-1.43.4/build# gunzip -v /usr/share/info/libext2fs.info>
/usr/share/info/libext2fs.info.gz: 79.6% -- replaced with /usr/share/info/
libext2fs.info
<=/usr/share/info/dir /usr/share/info/libext2fs.info
root:/sources/e2fsprogs-1.43.4/build#
```

Halutessasi luo ja asenna lisää dokumentaatioita:

```
makeinfo -o doc/com_err.info ../lib/et/com_err.texinfo
install -v -m644 doc/com_err.info /usr/share/info
install-info --dir-file=/usr/share/info/dir
/usr/share/info/com_err.info
```

```
< -o doc/com_err.info ../lib/et/com_err.texinfo
<# install -v -m644 doc/com_err.info /usr/share/info
'doc/com_err.info' -> '/usr/share/info/com_err.info'
<le=/usr/share/info/dir /usr/share/info/com_err.info
root:/sources/e2fsprogs-1.43.4/build#
```

Lopuksi poistetaan purettu hakemisto:

```
cd /sources/  
rm -rf e2fsprogs-1.43.4
```

```
root:/sources/e2fsprogs-1.43.4/build# cd /sources/  
root:/sources# rm -rf e2fsprogs-1.43.4  
root:/sources# █
```

2.15.44 Coreutils-8.26

Käännösaika: 3,1 SBU

Asennettavat ohjelmat: [, base32, base64, basename, cat, chcon, chgrp, chmod, chown, chroot, cksum, comm, cp, csplit, cut, date, dd, df, dir, dircolors, dirname, du, echo, env, expand, expr, factor, false, fmt, fold, groups, head, hostid, id, install, join, link, ln, logname, ls, md5sum, mkdir, mkfifo, mknod, mktemp, mv, nice, nl, nohup, nproc, numfmt, od, paste, pathchk, pinky, pr, printenv, printf, ptx, pwd, readlink, realpath, rm, rmdir, runcon, seq, sha1sum, sha224sum, sha256sum, sha384sum, sha512sum, shred, shuf, sleep, sort, split, stat, stdbuf, stty, sum, sync, tac, tail, tee, test, timeout, touch, tr, true, truncate, tsort, tty, uname, unexpand, uniq, unlink, users, vdir, wc, who, whoami ja yes

Asennettavat kirjastot: libstdbuf.so

Asennettavat hakemistot: /usr/libexec/coreutils

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf coreutils-8.26.tar.xz
cd coreutils-8.26
```

```
root:/sources# tar xf coreutils-8.26.tar.xz
root:/sources# cd coreutils-8.26
root:/sources/coreutils-8.26#
```

POSIX vaatii, että Coreutils:n ohjelmat tunnistavat merkistön rajat oikein jopa multibittisissä lokaaleissa. Seuraava komento korjaa tämän ja joitakin muita kansainvälistämiseen liittyviä bugeja.

```
patch -Np1 -i ../coreutils-8.26-il8n-1.patch
```

Sammutetaan testi, joka joillakin koneilla saattaa loopata loputtomiin.

```
sed -i '/test.lock/s/^/#/' gnulib-tests/gnulib.mk
```

```
<# sed -i '/test.lock/s/^/#/' gnulib-tests/gnulib.mk
root:/sources/coreutils-8.26#
```

Valmistellaan paketti kääntämistä varten:

```
FORCE_UNSAFE_CONFIGURE=1 ./configure \  
    --prefix=/usr \  
    --enable-no-install-program=kill,uptime
```

Puretaan auki:

- `FORCE_UNSAFE_CONFIGURE=1` → Tämä ympäristömuuttuja sallii paketin kääntämisen pääkäyttäjänä.
- `--enable-no-install-program=kill,uptime` → Estää Coreutilsia asentamasta binäärejä, jotka muut paketit asentavat myöhemmin.

Käännetään paketti:

```
FORCE_UNSAFE_CONFIGURE=1 make
```

```
root:/sources/coreutils-8.26# FORCE_UNSAFE_CONFIGURE=1 make  
GEN      lib/alloca.h  
GEN      lib/c++defs.h  
GEN      lib/warn-on-use.h
```

Ensin ajetaan testit, jotka on tarkoitettu ajettavaksi root-käyttäjänä:

```
make NON_ROOT_USERNAME=nobody check-root
```

```
=====  
Testsuite summary for GNU coreutils 8.26  
=====
```

# TOTAL:	30
# PASS:	15
# SKIP:	15
# XFAIL:	0
# FAIL:	0
# XPASS:	0
# ERROR:	0

```
=====
```

Loput testit ajetaan nobody-käyttäjänä. Jotkut testit vaativat, että käyttäjä on useamman kuin yhden ryhmän jäsen. Jotta näitä testejä ei ohitettaisi, lisätään tilapäinen ryhmä ja tehdään nobody-käyttäjstä sen jäsen:

```
echo "dummy:x:1000:nobody" >> /etc/group
```

```
root:/sources/coreutils-8.26# echo "dummy:x:1000:nobody" >> /etc/group
root:/sources/coreutils-8.26#
```

Korjataan joitakin käyttöoikeuksia, jotta ei-root-käyttäjä voi kääntää ja ajaa testit:

```
chown -Rv nobody .
```

Varmistetaan, että PATH sisältää su-ympäristössä /tools/bin ja ajetaan testit:

```
su nobody -s /bin/bash \
    -c "PATH=$PATH make RUN_EXPENSIVE_TESTS=yes
check"
```

Test-getlogin -testin tiedetään epäonnistuvan virtuaalisessa konsolissa - kuten chroot-ympäristössä - mutta menevän läpi, jos ajetaan X-terminaalissa.

```
=====
Testsuite summary for GNU coreutils 8.26
=====
# TOTAL: 310
# PASS: 296
# SKIP: 14
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Poistetaan tilapäinen ryhmä:

```
sed -i '/dummy/d' /etc/group
```

```
root:/sources/coreutils-8.26# sed -i '/dummy/d' /etc/group
root:/sources/coreutils-8.26#
```

Asennetaan paketti:

```
make install
```

```
root:/sources/coreutils-8.26# make install
make install-recursive
make[1]: Entering directory '/sources/coreutils-8.26'
Making install in po
make[2]: Entering directory '/sources/coreutils-8.26/po'
installing af.gmo as /usr/share/locale/af/LC_MESSAGES/coreutils.mo
```

Siirretään ohjelmat FHS:n määrittelemiін paikkoihin:

```
mv -v /usr/bin/{cat,chgrp,chmod,chown,cp,date,dd,df,echo}
/bin
```

```
<bin/{cat,chgrp,chmod,chown,cp,date,dd,df,echo} /bin
'/usr/bin/cat' -> '/bin/cat'
'/usr/bin/chgrp' -> '/bin/chgrp'
'/usr/bin/chmod' -> '/bin/chmod'
'/usr/bin/chown' -> '/bin/chown'
'/usr/bin/cp' -> '/bin/cp'
'/usr/bin/date' -> '/bin/date'
'/usr/bin/dd' -> '/bin/dd'
'/usr/bin/df' -> '/bin/df'
'/usr/bin/echo' -> '/bin/echo'
root:/sources/coreutils-8.26#
```

```
mv -v /usr/bin/{false,ln,ls,mkdir,mknod,mv,pwd,rm} /bin
```

```
<v /usr/bin/{false,ln,ls,mkdir,mknod,mv,pwd,rm} /bin
'/usr/bin/false' -> '/bin/false'
'/usr/bin/ln' -> '/bin/ln'
'/usr/bin/ls' -> '/bin/ls'
'/usr/bin/mkdir' -> '/bin/mkdir'
'/usr/bin/mknod' -> '/bin/mknod'
'/usr/bin/mv' -> '/bin/mv'
'/usr/bin/pwd' -> '/bin/pwd'
'/usr/bin/rm' -> '/bin/rm'
root:/sources/coreutils-8.26#
```

```
mv -v /usr/bin/{rmdir,stty,sync,true,uname} /bin
```

```
<6# mv -v /usr/bin/{rmdir,stty,sync,true,uname} /bin
'/usr/bin/rmdir' -> '/bin/rmdir'
'/usr/bin/stty' -> '/bin/stty'
'/usr/bin/sync' -> '/bin/sync'
'/usr/bin/true' -> '/bin/true'
'/usr/bin/uname' -> '/bin/uname'
root:/sources/coreutils-8.26#
```

```
mv -v /usr/bin/chroot /usr/sbin
```

```
root:/sources/coreutils-8.26# mv -v /usr/bin/chroot /usr/sbin
'/usr/bin/chroot' -> '/usr/sbin/chroot'
root:/sources/coreutils-8.26#
```

```
mv -v /usr/share/man/man1/chroot.1
/usr/share/man/man8/chroot.8
```

```
<hare/man/man1/chroot.1 /usr/share/man/man8/chroot.8
'/usr/share/man/man1/chroot.1' -> '/usr/share/man/man8/chroot.8'
root:/sources/coreutils-8.26#
```

```
sed -i s/"1"/"8"/1 /usr/share/man/man8/chroot.8
```

```
<sed -i s/"1"/"8"/1 /usr/share/man/man8/chroot.8
root:/sources/coreutils-8.26#
```

Jotkin LFS-Bootscripts -paketin skriptit riippuvat `head`, `sleep` ja `nice` :sta. Koska `/usr` ei välttämättä ole saatavilla käynnistysprosessin alkuvaiheilla, nuo binäärit pitää siirtää root-partitioon:

```
mv -v /usr/bin/{head,sleep,nice,test,[]} /bin
```

```
root:/sources/coreutils-8.26# mv -v /usr/bin/{head,sleep,nice,test,[]} /bin
'/usr/bin/head' -> '/bin/head'
'/usr/bin/sleep' -> '/bin/sleep'
'/usr/bin/nice' -> '/bin/nice'
'/usr/bin/test' -> '/bin/test'
'/usr/bin/[' -> '/bin/['
root:/sources/coreutils-8.26#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf coreutils-8.26
```

```
root:/sources/coreutils-8.26# cd ..  
root:/sources# rm -rf coreutils-8.26  
root:/sources# █
```

2.15.45 Diffutils-3.5

Käännösaika: 0,4 SBU

Asennettavat ohjelmat: *cmp, diff, diff3 ja sdiff*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf diffutils-3.5.  
cd diffutils-3.5
```

```
root:/sources# tar xf diffutils-3.5.tar.xz  
root:/sources# cd diffutils-3.5  
root:/sources/diffutils-3.5#
```

Korjataan tiedosto, jotta lokaalitiedostot asennetaan:

```
sed -i 's:= @mkdir_p@:= /bin/mkdir -p:' po/Makefile.in.in
```

```
< 's:= @mkdir_p@:= /bin/mkdir -p:' po/Makefile.in.in  
root:/sources/diffutils-3.5#
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/usr
```

```
root:/sources/diffutils-3.5# ./configure --prefix=/usr  
configure: error: unrecognized option: '--prefix=/usr'  
Try './configure --help' for more information  
root:/sources/diffutils-3.5# ./configure --prefix=/usr  
checking for a BSD-compatible install... /usr/bin/install -c
```

Käännetään paketti:

```
make
```

```
root:/sources/diffutils-3.5# make  
Making all in lib  
make[1]: Entering directory '/sources/diffutils-3.5/lib'  
Makefile:1777: target '.deps/alloca.Po' given more than once in the same rule  
GEN      alloca.h
```

Ajetaan testit:

```
make check
```

```
=====
Testsuite summary for GNU diffutils 3.5
=====
```

```
# TOTAL: 167
# PASS: 154
# SKIP: 13
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan paketti:

```
make install
```

```
root:/sources/diffutils-3.5# make install
Making install in lib
make[1]: Entering directory '/sources/diffutils-3.5/lib'
Makefile:1777: target '.deps/alloca.Po' given more than once in the same rule
make install-am
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf diffutils-3.5
```

```
root:/sources/diffutils-3.5# cd ..
root:/sources# rm -rf diffutils-3.5
root:/sources#
```

2.15.46 Gawk-4.1.4

Käännösaika: 0,3 SBU

Asennettavat ohjelmat: *awk* (linkki *gawk.iin*), *gawk*, *gawk-4.1.4* ja *igawk*

Asennettavat kirjastot: *filefuncs.so*, *fnmatch.so*, *fork.so*, *inplace.so*, *ordchr.so*, *readdir.so*, *readfile.so*, *revoutput.so*, *revtwoway.so*, *rwarray.so*, *testtext.so* ja *time.so*

Asennettavat hakemistot: */usr/lib/gawk*, */usr/libexec/awk*, */usr/share/awk* ja */usr/share/doc/gawk-4.1.4*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf gawk-4.1.4.tar.xz
cd gawk-4.1.4
```

```
root:/sources# tar xf gawk-4.1.4.tar.xz
root:/sources# cd gawk-4.1.4
root:/sources/gawk-4.1.4#
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/usr
```

```
root:/sources/gawk-4.1.4# ./configure --prefix=/usr
checking for a BSD-compatible install... ./install-sh -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
```

Käännetään paketti:

```
make
```

```
root:/sources/gawk-4.1.4# make
make 'CFLAGS=-g -O2 -DNDEBUG' 'LDFLAGS=-Wl,-export-dynamic' all-recursive
make[1]: Entering directory '/sources/gawk-4.1.4'
Making all in .
make[2]: Entering directory '/sources/gawk-4.1.4'
```

Ajetaan testit:

```
make check
```

```
===== Done with MPFR tests =====  
make[2]: Entering directory '/sources/gawk-4.1.4/test'  
ALL TESTS PASSED  
make[2]: Leaving directory '/sources/gawk-4.1.4/test'  
make[1]: Leaving directory '/sources/gawk-4.1.4/test'  
root:/sources/gawk-4.1.4#
```

Asennetaan paketti:

```
make install
```

```
root:/sources/gawk-4.1.4# make install  
Making install in .  
make[1]: Entering directory '/sources/gawk-4.1.4'  
make[2]: Entering directory '/sources/gawk-4.1.4'  
/bin/mkdir -p '/usr/bin'  
/install-sh -c -gawk '/usr/bin'
```

Asennetaan dokumentaatio (vapaaehtoisista):

```
mkdir -v /usr/share/doc/gawk-4.1.4
```

```
root:/sources/gawk-4.1.4# mkdir -v /usr/share/doc/gawk-4.1.4  
mkdir: created directory '/usr/share/doc/gawk-4.1.4'  
root:/sources/gawk-4.1.4#
```

```
cp -v doc/{awkforai.txt,*.eps,pdf,jpg}}  
/usr/share/doc/gawk-4.1.4
```

```
<orai.txt,{*.eps,pdf,jpg}} /usr/share/doc/gawk-4.1.4
'doc/awkforai.txt' -> '/usr/share/doc/gawk-4.1.4/awkforai.txt'
'doc/api-figure1.eps' -> '/usr/share/doc/gawk-4.1.4/api-figure1.eps'
'doc/api-figure2.eps' -> '/usr/share/doc/gawk-4.1.4/api-figure2.eps'
'doc/api-figure3.eps' -> '/usr/share/doc/gawk-4.1.4/api-figure3.eps'
'doc/array-elements.eps' -> '/usr/share/doc/gawk-4.1.4/array-elements.eps'
'doc/general-program.eps' -> '/usr/share/doc/gawk-4.1.4/general-program.eps'
'doc/lflashlight.eps' -> '/usr/share/doc/gawk-4.1.4/lflashlight.eps'
'doc/process-flow.eps' -> '/usr/share/doc/gawk-4.1.4/process-flow.eps'
'doc/rflashlight.eps' -> '/usr/share/doc/gawk-4.1.4/rflashlight.eps'
'doc/statist.eps' -> '/usr/share/doc/gawk-4.1.4/statist.eps'
'doc/api-figure1.pdf' -> '/usr/share/doc/gawk-4.1.4/api-figure1.pdf'
'doc/api-figure2.pdf' -> '/usr/share/doc/gawk-4.1.4/api-figure2.pdf'
'doc/api-figure3.pdf' -> '/usr/share/doc/gawk-4.1.4/api-figure3.pdf'
'doc/array-elements.pdf' -> '/usr/share/doc/gawk-4.1.4/array-elements.pdf'
'doc/general-program.pdf' -> '/usr/share/doc/gawk-4.1.4/general-program.pdf'
'doc/lflashlight.pdf' -> '/usr/share/doc/gawk-4.1.4/lflashlight.pdf'
'doc/process-flow.pdf' -> '/usr/share/doc/gawk-4.1.4/process-flow.pdf'
'doc/rflashlight.pdf' -> '/usr/share/doc/gawk-4.1.4/rflashlight.pdf'
'doc/statist.pdf' -> '/usr/share/doc/gawk-4.1.4/statist.pdf'
'doc/statist.jpg' -> '/usr/share/doc/gawk-4.1.4/statist.jpg'
root:/sources/gawk-4.1.4#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf gawk-4.1.4
```

```
root:/sources/gawk-4.1.4# cd ..
root:/sources# rm -rf gawk-4.1.4
root:/sources#
```

2.15.47 Findutils-4.6.0

Käännösaika: 0,9 SBU

Asennettavat ohjelmat: *code, find, locate, oldfind, updated ja xargs*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf findutils-4.6.0.tar.gz
cd findutils-4.6.0
```

```
root:/sources# tar xf findutils-4.6.0.tar.gz
root:/sources# cd findutils-4.6.0
root:/sources/findutils-4.6.0#
```

Sammutetaan testi, joka saattaa joillakin koneilla loopata loputtomiin:

```
sed -i 's/test-lock..EXEEXT.//' tests/Makefile.in
```

```
<# sed -i 's/test-lock..EXEEXT.//' tests/Makefile.in
root:/sources/findutils-4.6.0#
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/usr --localstatedir=/var/lib/locate
```

Puretaan auki:

- `--localstatedir` → Muuttaa `locate`-tietokannan paikan `/var/lib/locate` :ksi, joka on FHS:n mukainen.

Käännetään paketti:

```
make
```

Ajetaan testit:

```
make check
```

```
=====
Testsuite summary for GNU findutils 4.6.0
=====
# TOTAL: 224
# PASS: 212
# SKIP: 12
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

```
=====
Testsuite summary for GNU findutils 4.6.0
=====
# TOTAL: 2
# PASS: 2
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

```
=== find Summary ===

# of expected passes          941
# of untested testcases      2
find (GNU findutils) 4.6.0
```

```
=====
Testsuite summary for GNU findutils 4.6.0
=====
# TOTAL: 6
# PASS: 6
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

```
=== xargs Summary ===  
# of expected passes          96  
xargs (GNU findutils) 4.6.0
```

```
=== locate Summary ===  
# of expected passes          34  
locate (GNU findutils) 4.6.0
```

Asennetaan paketti:

```
make install
```

```
root:/sources/findutils-4.6.0# make install  
Making install in gl  
make[1]: Entering directory '/sources/findutils-4.6.0/gl'  
Making install in lib  
make[2]: Entering directory '/sources/findutils-4.6.0/gl/lib'
```

Jotkin LFS-Bootscripts -paketin skriptit riippuvat find:sta. Koska /usr ei välttämättä ole saatavilla käynnistysprosessin alkuvaiheella, tämä ohjelma täytyy olla root-partitiossa. updatedb-skriptiä pitää myös muokata polun täsmentämiseksi:

```
mv -v /usr/bin/find /bin
```

```
root:/sources/findutils-4.6.0# mv -v /usr/bin/find /bin  
'/usr/bin/find' -> '/bin/find'  
root:/sources/findutils-4.6.0#
```

```
sed -i 's|find:=${BINDIR}|find:="/bin|' /usr/bin/updatedb
```

```
<i 's|find:=${BINDIR}|find:="/bin|' /usr/bin/updatedb  
root:/sources/findutils-4.6.0#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf findutils-4.6.0
```

```
root:/sources/findutils-4.6.0# cd ..  
root:/sources# rm -rf findutils-4.6.0  
root:/sources# █
```

2.15.48 Groff-1.22.3

Käännösaika: 0,5 SBU

Asennettavat ohjelmat: *addftinfo, afmtodit, chem, eqn, eqn2graph, gdiffmk, glilypond, gperl, gpinyin, grap2graph, grn, grodvi, groff, groffer, grog, grolbp, grolj4, gropdf, grops, grotty, hpftodit, indxbib, lkbib, lookbib, mmroff, neqn, nroff, pdfmom, pdfroff, pfbtops, pic, pic2graph, postgrohtml, preconv, pre-grohtml, refer, roff2dvi, roff2html, roff2pdf, roff2ps, roff2text, roff2x, soelim, tbl, tfmtodit ja troff*

Asennettavat hakemistot: */usr/lib/groff, /usr/share/doc/groff-1.22.3 ja /usr/share/groff*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf groff-1.22.3.tar.gz
cd groff-1.22.3
```

```
root:/sources# tar xf groff-1.22.3.tar.gz
root:/sources# cd groff-1.22.3
root:/sources/groff-1.22.3#
```

Groff olettaa, että PAGE-ympäristömuuttuja sisältää oletuspaperikoon. Amerikkalaisten kannattaa laittaa siihen PAGE=letter. Muualla PAGE=A4 on sopivampi. Asetusta voi muuttaa myöhemmin.

Valmistellaan Groff kääntämistä varten:

```
PAGE=A4 ./configure --prefix=/usr
```

```
groff version 1.22.3
-----
Prefix           : /usr
Compiler         : gcc -g -O2
X11 support      : no
Doc build        : yes
-----
root:/sources/groff-1.22.3#
```

Käännetään paketti:

```
make
```

Tämän paketin mukana ei tule testejä.

Asennetaan paketti:

```
make install
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf groff-1.22.3
```

```
root:/sources/groff-1.22.3# cd ..  
root:/sources# rm -rf groff-1.22.3  
root:/sources#
```

2.15.49 GRUB-2.02~beta3

Käännösaika: 0,8 SBU

Asennettavat ohjelmat: *grub-bios-setup, grub-editenv, grub-file, grub-fstest, grub-glue-efi, grub-install, grub-kbdcomp, grub-macbless, grub-menulst2cfg, grub-mkconfig, grub-mkimage, grub-mklayout, grub-mknetdir, grub-mkpasswd-pbkdf2, grub-mkrelpath, grub-mkrescue, grub-mkstandalone, grub-ofpathname, grub-probe, grub-reboot, grub-render-label, grub-script-check, grub-set-default, grub-sparc64-setup ja grub-syslinux2cfg*

Asennettavat hakemistot: */usr/lib/grub, /etc/grub.d, /usr/share/grub ja boot/grub (kun grub-install ajetaan ensimmäisen kerran)*

Ensimmäiseksi puretaan paketti ja siirytään purettuun hakemistoon:

```
tar xf grub-2.02~beta3.tar.xz
cd grub-2.02~beta3
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/usr \
            --sbindir=/sbin \
            --sysconffdir=/etc \
            --disable-efiemu \
            --disable-werror
```

```
*****
GRUB2 will be compiled with following components:
Platform: i386-pc
With devmapper support: No (need libdevmapper header)
With memory debugging: No
With disk cache statistics: No
With boot time statistics: No
efiemu runtime: No (explicitly disabled)
grub-mkfont: No (need freetype2 library)
grub-mount: No (need FUSE library)
starfield theme: No (No build-time grub-mkfont)
With libzfs support: No (need zfs library)
Build-time grub-mkfont: No (need freetype2 library)
Without unifont (no build-time grub-mkfont)
With liblzma from -llzma (support for XZ-compressed mips images)
*****
root:/sources/grub-2.02~beta3#
```

Puretaan auki:

- `--disable-werror` → Tuoreet Flex-versiot aiheuttavat varoituksia. Tämä asetus sallii kääntämisen niistä huolimatta.
- `--disable-efiemu` → Estetään turhien toimintojen ja testiohjelmien kääntäminen, joita ei LFS:ssä tarvita.

Käännetään paketti:

```
make
```

Tämän paketin mukana ei tule testejä.

Asennetaan paketti:

```
make install
```

```
root:/sources/grub-2.02~beta3# make install
make install-recursive
make[1]: Entering directory '/sources/grub-2.02~beta3'
Making install in grub-core/gnulib
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf grub-2.02~beta3
```

```
root:/sources/grub-2.02~beta3# cd ..
root:/sources# rm -rf grub-2.02~beta3
root:/sources#
```

2.15.50 3.6.50 Less-481

Käännösaika: alle 0,1 SBU

Asennettavat ohjelmat: *less, lessecho ja lesskey*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf less-481.tar.gz
cd less-481
```

```
root:/sources# tar xf less-481.tar.gz
root:/sources# cd less-481
root:/sources/less-481#
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/usr --sysconfdir=/etc
```

```
root:/sources/less-481# ./configure --prefix=/usr --sysconfdir=/etc
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
```

Puretaan auki:

- `--sysconfdir=/etc` → Käskee ohjelman luoman paketin etsiä `/etc` -hakemistosta konfiguraatiotiedostoja.

Käännetään paketti:

```
make
```

```
root:/sources/less-481# make
test ! -f stamp-h || CONFIG_FILES= CONFIG_HEADERS=defines.h ./config.status
touch stamp-h
```

Tämän paketin mukana ei tule testejä.

Asennetaan paketti:

```
make install
```

```
root:/sources/less-481# make install
./mkinstalldirs /usr/bin /usr/share/man/man1
/usr/bin/install -c less /usr/bin/less
/usr/bin/install -c lesskey /usr/bin/lesskey
/usr/bin/install -c lessecho /usr/bin/lessecho
/usr/bin/install -c -m 644 ./less.nro /usr/share/man/man1/less.1
/usr/bin/install -c -m 644 ./lesskey.nro /usr/share/man/man1/lesskey.1
/usr/bin/install -c -m 644 ./lessecho.nro /usr/share/man/man1/lessecho.1
root:/sources/less-481#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf less-481
```

```
root:/sources/less-481# cd ..
root:/sources# rm -rf less-481
root:/sources#
```

2.15.51 3.6.51 Gzip-1.8

Käännösaika: 0,1 SBU

Asennettavat ohjelmat: *gunzip, gzexe, gzip, uncompress (kova linkki gunzip:in kanssa), zcat, zcmp, zdiff, zegrep, zfgrep, zforce, zgrep, zless, zmore ja znew*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf gzip-1.8.tar.xz
cd gzip-1.8
```

```
root:/sources# tar xf gzip-1.8.tar.xz
root:/sources# cd gzip-1.8
root:/sources/gzip-1.8#
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/usr
```

```
root:/sources/gzip-1.8# ./configure --prefix=/usr
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
```

Käännetään paketti:

```
make
```

```
root:/sources/gzip-1.8# make
GEN      version.c
GEN      version.h
make all-recursive
make[1]: Entering directory '/sources/gzip-1.8'
```

Ajetaan testit:

```
make check
```

```
=====
Testsuite summary for gzip 1.8
=====
# TOTAL: 18
# PASS: 18
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan paketti:

```
make install
```

```
root:/sources/gzip-1.8# make install
make install-recursive
make[1]: Entering directory '/sources/gzip-1.8'
Making install in lib
make[2]: Entering directory '/sources/gzip-1.8/lib'
make install-am
make[3]: Entering directory '/sources/gzip-1.8/lib'
```

Siirretään ohjelmat, jonka täytyy olla root-tiedostojärjestelmässä.

```
mv -v /usr/bin/gzip /bin
```

```
root:/sources/gzip-1.8# mv -v /usr/bin/gzip /bin
'/usr/bin/gzip' -> '/bin/gzip'
root:/sources/gzip-1.8#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf gzip-1.8
```

```
root:/sources/gzip-1.8# cd ..
root:/sources# rm -rf gzip-1.8
root:/sources#
```

2.15.52 IPRoute2-4.9.0

Käännösaika: 0,2 SBU

Asennettavat ohjelmat: bridge, ctstat (linkki lstat:iin), genl, ifcfg, ifstat, ip, lstat, nstat, route, routel, rtacct, rtmon, rtpr, rtstat (linkki lstat:iin), ss ja tc

Asennettavat hakemistot: /etc/iproute2, /usr/lib/tc ja /usr/share/doc/iproute2-4.9.0

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf iproute2-4.9.0.tar.xz
cd iproute2-4.9.0
```

```
root:/sources# tar xf iproute2-4.9.0.tar.xz
root:/sources# cd iproute2-4.9.0
root:/sources/iproute2-4.9.0#
```

Tämän paketin sisältämää arpd-ohjelmaa ei käännetä, koska se on riippuvainen Berkeley DB:stä, jota ei asennetta LFS:ään. arpd:n dokumentaatiotiedostot ja hakemisto asennetaan kuitenkin. Estetään tämä alla olevilla komennoilla. Jos arpd-binääriä tarvitaan, ohjeet Berkeley DB:n kääntämiseen löytyvät [BLFS](#)-kirjasta.

```
sed -i /ARPD/d Makefile
sed -i 's/arpd.8//' man/man8/Makefile
rm -v doc/arpd.sgml
```

```
root:/sources/iproute2-4.9.0# sed -i /ARPD/d Makefile
root:/sources/iproute2-4.9.0# sed -i 's/arpd.8//' man/man8/Makefile
root:/sources/iproute2-4.9.0# rm -v doc/arpd.sgml
removed 'doc/arpd.sgml'
root:/sources/iproute2-4.9.0#
```

Estetään myös yhtä moduulia kääntämästä, koska se vaatii Iptables:n.

```
sed -i 's/m_ipt.o//' tc/Makefile
```

```
root:/sources/iproute2-4.9.0# sed -i 's/m_ipt.o//' tc/Makefile
root:/sources/iproute2-4.9.0#
```

Käännetään paketti:

```
make
```

```
root:/sources/iproute2-4.9.0# make
sh configure /usr/include
TC schedulers
  ATM      no

libc has setns: yes
SELinux support: no
ELF support: no
libmnl support: no
Berkeley DB: no

docs: latex: no
WARNING: no docs can be built from LaTeX files
sgml2html: no
WARNING: no HTML docs can be built from SGML
```

Tällä paketilla ei ole toimivia testejä.

Asennetaan paketti:

```
make DOCDIR=/usr/share/doc/iproute2-4.9.0 install
```

```
<# make DOCDIR=/usr/share/doc/iproute2-4.9.0 install
lib
make[1]: Nothing to be done for 'all'.

ip
make[1]: Nothing to be done for 'all'.
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf iproute2-4.9.0
```

```
root:/sources/iproute2-4.9.0# cd ..
root:/sources# rm -rf iproute2-4.9.0
root:/sources#
```

2.15.53 3.6.53 Kbd-2.0.4

Käännösaika: 0,1 SBU

Asennetaan ohjelmat: *chvt, dealloct, dumpkeys, fgconsole, getkeycodes, kbinfo, kbd_mode, kbdrate, loadkeys, loadunimap, mapscrn, openvt, psfaddtable (linkki psfxtable:en), psfgettable (linkki psfxtable:en), psfstriptime (linkki psfxtable:en), psfxtable, setfont, setkeycodes, setleds, setmetamode, setvtrgb, showconsolefont, showkey, unicode_start ja unicode_stop*

Asennetaan hakemistot: */usr/share/consolefonts, /usr/share/consoletrans, /usr/share/keymaps, /usr/share/doc/kbd-2.0.4 ja /usr/share/unimaps*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf kbd-2.0.4.tar.xz
cd kbd-2.0.4
```

```
root:/sources# tar xf kbd-2.0.4.tar.xz
root:/sources# cd kbd-2.0.4
root:/sources/kbd-2.0.4#
```

Backspace ja Delete -painikkeiden toiminta ei ole yhtenäistä Kbd-paketin näppäimistökartoissa. Seuraavalla komennolla korjataan ongelma i386-näppäimistökartoille:

```
patch -Np1 -i ../kbd-2.0.4-backspace-1.patch
```

```
root:/sources/kbd-2.0.4# patch -Np1 -i ../kbd-2.0.4-backspace-1.patch
patching file data/keymaps/i386/dvorak/dvorak-l.map
patching file data/keymaps/i386/dvorak/dvorak-r.map
patching file data/keymaps/i386/fqGIod/tr f-latin5.map
```

Poistetaan turha `resizecons`-ohjelma ja sen man-sivut:

```
sed -i 's/\\(RESIZECONS_PROGS=\\)yes/\\1no/g' configure
sed -i 's/resizecons.8 //' docs/man/man8/Makefile.in
```

```
root:/sources/kbd-2.0.4# sed -i 's/\\(RESIZECONS_PROGS=\\)yes/\\1no/g' configure
root:/sources/kbd-2.0.4# sed -i 's/resizecons.8 //' docs/man/man8/Makefile.in
root:/sources/kbd-2.0.4#
```

Valmistellaan Kbd kääntämistä varten:

```
PKG_CONFIG_PATH=/tools/lib/pkgconfig ./configure --pre-  
fix=/usr --disable-vlock
```

```
kbd 2.0.4  
=====  
  
prefix:           /usr  
libdir:           ${exec_prefix}/lib  
bindir:           ${exec_prefix}/bin  
datadir:          ${datarootdir}  
  
compiler:         gcc  
cflags:           -g -O2 -Wall -Wextra -Wmissing-noreturn -Wdisabl  
ed-optimization -Wcast-align -Wshadow -Wmissing-format-attribute -Wmissing-proto  
types -Wstrict-prototypes -Wmissing-declarations  
  
{get,set}keycodes: yes  
resizecons:       no  
optional progs:    no  
vlock:            no  
libkeymap:         no  
  
root:/sources/kbd-2.0.4#
```

Puretaan auki:

- `--disable-vlock` → Estää `vlock`-työkalun kääntämisen, koska se vaa-
tii PAM-kirjaston, joka ei ole saatavilla chroot-ympäristössä.

Käännetään paketti:

```
make
```

```
root:/sources/kbd-2.0.4# make  
make all-recursive  
make[1]: Entering directory '/sources/kbd-2.0.4'  
Making all in src  
make[2]: Entering directory '/sources/kbd-2.0.4/src'
```

Ajetaan testit:

```
make check
```

```
=====  
All 7 tests passed  
=====
```

Asennetaan paketti:

```
make install
```

```
root:/sources/kbd-2.0.4# make install
Making install in src
make[1]: Entering directory '/sources/kbd-2.0.4/src'
Making install in vlock
make[2]: Entering directory '/sources/kbd-2.0.4/src/vlock'
```

Asennetaan dokumentaatio (vapaaehtoisista):

```
mkdir -v /usr/share/doc/kbd-2.0.4
```

```
root:/sources/kbd-2.0.4# mkdir -v /usr/share/doc/kbd-2.0.4
mkdir: created directory '/usr/share/doc/kbd-2.0.4'
root:/sources/kbd-2.0.4#
```

```
cp -R -v docs/doc/* /usr/share/doc/kbd-2.0.4
```

```
root:/sources/kbd-2.0.4# cp -R -v docs/doc/* /usr/share/doc/kbd-2.0.4
'docs/doc/A20' -> '/usr/share/doc/kbd-2.0.4/A20'
'docs/doc/A20/xfix-286mode2' -> '/usr/share/doc/kbd-2.0.4/A20/xfix-286mode2'
'docs/doc/A20/A20.html' -> '/usr/share/doc/kbd-2.0.4/A20/A20.html'
'docs/doc/README-Crosser' -> '/usr/share/doc/kbd-2.0.4/README-Crosser'
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
```

```
rm -rf kbd-2.0.4
```

```
root:/sources/kbd-2.0.4# cd ..
root:/sources# rm -rf kbd-2.0.4
root:/sources#
```

2.15.54 Libpipeline-1.4.1

Käännösaika: 0,1 SBU

Asennettava kirjasto: libpipeline.so

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf libpipeline-1.4.1.tar.gz
cd libpipeline-1.4.1
```

```
root:/sources# tar xf libpipeline-1.4.1.tar.gz
root:/sources# cd libpipeline-1.4.1
root:/sources/libpipeline-1.4.1#
```

Valmistellaan paketti kääntämistä varten:

```
PKG_CONFIG_PATH=/tools/lib/pkgconfig ./configure --prefix=/usr
```

```
<PATH=/tools/lib/pkgconfig ./configure --prefix=/usr
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
```

Puretaan auki:

- `PKG_CONFIG_PATH` → Käytetään pkg-config:ia hankkiaksemme test-kirjaston metadatan paikan, joka käännettiin osiossa [2.8.11 Check-0.11.0](#).

Käännetään paketti:

```
make
```

```
root:/sources/libpipeline-1.4.1# make
make all-recursive
make[1]: Entering directory '/sources/libpipeline-1.4.1'
Making all in gnu/lib/lib
make[2]: Entering directory '/sources/libpipeline-1.4.1/gnu/lib/lib'
```

Ajetaan testit:

```
make check
```

```
=====
Testsuite summary for libpipeline 1.4.1
=====
```

```
# TOTAL: 7
# PASS:  7
# SKIP:  0
# XFAIL: 0
# FAIL:  0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan paketti:

```
make install
```

```
root:/sources/libpipeline-1.4.1# make install
Making install in gnulib/lib
make[1]: Entering directory '/sources/libpipeline-1.4.1/gnulib/lib'
make install-recursive
make[2]: Entering directory '/sources/libpipeline-1.4.1/gnulib/lib'
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf libpipeline-1.4.1
```

```
root:/sources/libpipeline-1.4.1# cd ..
root:/sources# rm -rf libpipeline-1.4.1
root:/sources#
```

2.15.55 3.6.55 Make-4.2.1

Käännösaika: 0,5 SBU

Asennettavat ohjelmat: *make*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf make-4.2.1.tar.bz2
cd make-4.2.1
```

```
root:/sources# tar xf make-4.2.1.tar.bz2
root:/sources# cd make-4.2.1
root:/sources/make-4.2.1#
```

Valmistellaan Make kääntämistä varten:

```
./configure --prefix=/usr
```

```
root:/sources/make-4.2.1# ./configure --prefix=/usr
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
```

Käännetään paketti:

```
make
```

```
root:/sources/make-4.2.1# make
make all-recursive
make[1]: Entering directory '/sources/make-4.2.1'
Making all in glob
make[2]: Entering directory '/sources/make-4.2.1/glob'
```

Ajetaan testit:

```
make check
```

```
574 Tests in 119 Categories Complete ... No Failures :-)
```

Asennetaan paketti:

```
make install
```

```
root:/sources/make-4.2.1# make install
Making install in glob
make[1]: Entering directory '/sources/make-4.2.1/glob'
make[2]: Entering directory '/sources/make-4.2.1/glob'
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf make-4.2.1
```

```
root:/sources/make-4.2.1# cd ..
root:/sources# rm -rf make-4.2.1
root:/sources# █
```

2.15.56 Patch-2.7.5

Käännösaika: 0,2 SBU

Asennettavat ohjelmat: patch

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf patch-2.7.5.tar.xz
cd patch-2.7.5
```

```
root:/sources# tar xf patch-2.7.5.tar.xz
root:/sources# cd patch-2.7.5
root:/sources/patch-2.7.5#
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/usr
```

```
root:/sources/patch-2.7.5# ./configure --prefix=/usr
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
```

Käännetään paketti:

```
make
```

```
root:/sources/patch-2.7.5# make
make all-recursive
make[1]: Entering directory '/sources/patch-2.7.5'
Making all in lib
make[2]: Entering directory '/sources/patch-2.7.5/lib'
```

Ajetaan testit:

```
make check
```

```
=====
Testsuite summary for GNU patch 2.7.5
=====
# TOTAL: 39
# PASS: 37
# SKIP: 1
# XFAIL: 1
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan paketti:

```
make install
```

```
root:/sources/patch-2.7.5# make install
make install-recursive
make[1]: Entering directory '/sources/patch-2.7.5'
Making install in lib
make[2]: Entering directory '/sources/patch-2.7.5/lib'
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf patch-2.7.5
```

```
root:/sources/patch-2.7.5# cd ..
root:/sources# rm -rf patch-2.7.5
root:/sources#
```

2.15.57 Sysklogd-1.5.1

Käännösaika: alle 0,1 SBU

Asennettavat ohjelmat: klogd ja syslogd

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf sysklogd-1.5.1.tar.gz
cd sysklogd-1.5.1
```

```
root:/sources# tar xf sysklogd-1.5.1.tar.gz
root:/sources# cd sysklogd-1.5.1
root:/sources/sysklogd-1.5.1#
```

Korjataan ongelmat, jotka aiheuttavat segmentaatiovirheen joissakin olosuhteissa. Lisäksi korjataan vanhentunut ohjelmarakenne.

```
sed -i '/Error loading kernel symbols/{n;n;d}' ksym_mod.c
sed -i 's/union wait/int/' syslogd.c
```

```
< '/Error loading kernel symbols/{n;n;d}' ksym_mod.c
root:/sources/sysklogd-1.5.1# sed -i 's/union wait/int/' syslogd.c
root:/sources/sysklogd-1.5.1#
```

Käännetään paketti.

```
make
```

```
root:/sources/sysklogd-1.5.1# make
gcc -O3 -DSYSV -fomit-frame-pointer -Wall -fno-strength-reduce -DSYSLOG_INET -D
SYSLOG_UNIXAF -DNO_SCCS -DFSSTND -DSYSLOGD_PIDNAME=\"syslogd.pid\" -c syslogd.c
gcc -o syslogd.o pidfile.o pidfile.c
```

Tämän paketin mukana ei tule testejä.

Asennetaan paketti:

```
make BINDIR=/sbin install
```

```
root:/sources/sysklogd-1.5.1# make BINDIR=/sbin install
/usr/bin/install -o root -g root -m 644 sysklogd.8 /usr/share/man/man8/sysklogd.8
/usr/bin/install -o root -g root -m 644 syslogd.8 /usr/share/man/man8/syslogd.8
/usr/bin/install -o root -g root -m 644 syslog.conf.5 /usr/share/man/man5/syslog.conf.5
/usr/bin/install -o root -g root -m 644 klogd.8 /usr/share/man/man8/klogd.8
/usr/bin/install -m 500 -s syslogd /sbin/syslogd
/usr/bin/install -m 500 -s klogd /sbin/klogd
root:/sources/sysklogd-1.5.1#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf sysklogd-1.5.1
```

```
root:/sources/sysklogd-1.5.1# cd ..
root:/sources# rm -rf sysklogd-1.5.1
root:/sources#
```

2.15.57.1 Syslogd:n konfiguroiminen

Luodaan uusi `/etc/syslog.conf` -tiedosto:

```
cat > /etc/syslog.conf << "EOF"
# Begin /etc/syslog.conf

auth,authpriv.* -/var/log/auth.log
*.*;auth,authpriv.none -/var/log/sys.log
daemon.* -/var/log/daemon.log
kern.* -/var/log/kern.log
mail.* -/var/log/mail.log
user.* -/var/log/user.log
*.emerg *

# End /etc/syslog.conf
EOF
```

```
root:/sources# cat > /etc/syslog.conf << "EOF"
> # Begin /etc/syslog.conf
>
> auth,authpriv.* -/var/log/auth.log
> *.*;auth,authpriv.none -/var/log/sys.log
> daemon.* -/var/log/daemon.log
> kern.* -/var/log/kern.log
> mail.* -/var/log/mail.log
> user.* -/var/log/user.log
> *.emerg *
>
> # End /etc/syslog.conf
> EOF
root:/sources#
```

2.15.58 Sysvinit-2.88dsf

Käännösaika: alle 0,1 SBU

Asennettavat ohjelmat: *bootlogd, fstab-decode, halt, init, killall5, poweroff (linkki halt:iin), reboot (linkki halt:iin), runlevel, shutdown ja telinit (linkki init:iin)*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf sysvinit-2.88dsf.tar.bz2
cd sysvinit-2.88dsf
```

```
root:/sources# tar xf sysvinit-2.88dsf.tar.bz2
root:/sources# cd sysvinit-2.88dsf
root:/sources/sysvinit-2.88dsf#
```

Lisätään korjaus, joka poistaa useita muiden pakettien asentamia ohjelmia, selvittää viestiä ja korjaa kääntäjän varoituksen:

```
patch -Np1 -i ../sysvinit-2.88dsf-consolidated-1.patch
```

```
<ch -Np1 -i ../sysvinit-2.88dsf-consolidated-1.patch
patching file src/Makefile
patching file src/init.c
patching file src/shutdown.c
root:/sources/sysvinit-2.88dsf#
```

Käännetään paketti:

```
make -C src
```

```
root:/sources/sysvinit-2.88dsf# make -C src
make: Entering directory '/sources/sysvinit-2.88dsf/src'
cc -ansi -O2 -fomit-frame-pointer -W -Wall -D_GNU_SOURCE -c -o init.o init.c
cc -ansi -O2 -fomit-frame-pointer -W -Wall -D_GNU_SOURCE -DINIT_MAIN -c -o init.o init.c
```

Tämän paketin mukana ei tule testejä.

Asennetaan paketti:

```
make -C src install
```

```
root:/sources/sysvinit-2.88dsf# make -C src install
make: Entering directory '/sources/sysvinit-2.88dsf/src'
install -m 755 -d /sbin/
for i in ; do \
    install -o root -g root -m 755 $i /bin/ ; \
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf sysvinit-2.88dsf
```

```
root:/sources/sysvinit-2.88dsf# cd ..
root:/sources# rm -rf sysvinit-2.88dsf
root:/sources# █
```

2.15.59 Eudev-3.2.1

Käännösaika: 0,2 SBU

Asennettavat ohjelmat: udevadm ja udevd

Asennettavat kirjastot: libudev.so

Asennettavat hakemistot: /etc/udev, /lib/udev ja /usr/share/doc/udev-20140408

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf eudev-3.2.1.tar.gz
cd eudev-3.2.1
```

```
root:/sources# tar xf eudev-3.2.1.tar.gz
root:/sources# cd eudev-3.2.1
root:/sources/eudev-3.2.1#
```

Korjataan testiskripti:

```
sed -r -i 's|/usr(/bin/test)|\1|' test/udev-test.pl
```

```
<sed -r -i 's|/usr(/bin/test)|\1|' test/udev-test.pl
root:/sources/eudev-3.2.1#
```

Poistetaan tarpeeton rivi, joka aiheuttaa käännösvirheen:

```
sed -i '/keyboard_lookup_key/d' src/udev/udev-builtin-keyboard.c
```

```
<oard_lookup_key/d' src/udev/udev-builtin-keyboard.c
root:/sources/eudev-3.2.1#
```

Lisätään väliaikainen korjaus estääksemme, ettei /tools -hakemistoa kovakoodata:

```
cat > config.cache << "EOF"
HAVE_BLKID=1
BLKID_LIBS="-lblkid"
BLKID_CFLAGS="-I/tools/include"
EOF
```

```
root:/sources/eudev-3.2.1# cat > config.cache << "EOF"
> HAVE_BLKID=1
> BLKID_LIBS="-lblkid"
> BLKID_CFLAGS="-I/tools/include"
> EOF
root:/sources/eudev-3.2.1#
```

Valmistellaan Eudev kääntämistä varten:

```
./configure --prefix=/usr \
            --bindir=/sbin \
            --sbindir=/sbin \
            --libdir=/usr/lib \
            --sysconfdir=/etc \
            --libexecdir=/lib \
            --with-rootprefix= \
            --with-rootlibdir=/lib \
            --enable-manpages \
            --disable-static \
            --config-cache
```

```
root:/sources/eudev-3.2.1# ./configure --prefix=/usr \
> --bindir=/sbin \
> --sbindir=/sbin \
> --libdir=/usr/lib \
> --sysconfdir=/etc \
> --libexecdir=/lib \
> --with-rootprefix= \
> --with-rootlibdir=/lib \
> --enable-manpages \
> --disable-static \
> --config-cache
configure: loading cache config.cache
checking for gcc... gcc
checking whether the C compiler works... yes
```

Käännetään paketti.

```
LIBRARY_PATH=/tools/lib make
```

```
root:/sources/eudev-3.2.1# LIBRARY_PATH=/tools/lib make
make all-recursive
make[1]: Entering directory '/sources/eudev-3.2.1'
Making all in src
make[2]: Entering directory '/sources/eudev-3.2.1/src'
```

Luodaan joitakin hakemistoja, joita tarvitaan testejä varten. Niitä käytetään myös osana asennusta:

```
mkdir -pv /lib/udev/rules.d
```

```
root:/sources/eudev-3.2.1# mkdir -pv /lib/udev/rules.d
mkdir: created directory '/lib/udev'
mkdir: created directory '/lib/udev/rules.d'
root:/sources/eudev-3.2.1# █
```

```
mkdir -pv /etc/udev/rules.d
```

```
root:/sources/eudev-3.2.1# mkdir -pv /etc/udev/rules.d
mkdir: created directory '/etc/udev'
mkdir: created directory '/etc/udev/rules.d'
root:/sources/eudev-3.2.1# █
```

Ajetaan testit:

```
make LD_LIBRARY_PATH=/tools/lib check
```

```
=====
Testsuite summary for eudev 3.2.1
=====
# TOTAL: 2
# PASS: 2
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan paketti:

```
make LD_LIBRARY_PATH=/tools/lib install
```

```
root:/sources/eudev-3.2.1# make LD_LIBRARY_PATH=/tools/lib install
Making install in src
make[1]: Entering directory '/sources/eudev-3.2.1/src'
Making install in shared
make[2]: Entering directory '/sources/eudev-3.2.1/src/shared'
```

Asennetaan joitakin kustomoituja sääntöjä ja support-tiedostoja, jotka ovat käytännöllisiä LFS-ympäristössä:

```
tar -xvf ../udev-lfs-20140408.tar.bz2
```

```
root:/sources/eudev-3.2.1# tar -xvf ../udev-lfs-20140408.tar.bz2
udev-lfs-20140408/
udev-lfs-20140408/Makefile.lfs
udev-lfs-20140408/init-net-rules.sh
udev-lfs-20140408/83-cdrom-symlinks.rules
udev-lfs-20140408/contrib/
udev-lfs-20140408/contrib/debian/
udev-lfs-20140408/contrib/debian/83-cdrom-symlinks.rules
udev-lfs-20140408/contrib/debian/81-cdrom.rules
udev-lfs-20140408/contrib/debian/write_cd_aliases
udev-lfs-20140408/81-cdrom.rules
udev-lfs-20140408/55-lfs.rules
udev-lfs-20140408/write_cd_rules
udev-lfs-20140408/ChangeLog
udev-lfs-20140408/README
udev-lfs-20140408/write_net_rules
udev-lfs-20140408/rule_generator.functions
udev-lfs-20140408/55-lfs.txt
root:/sources/eudev-3.2.1#
```

```
make -f udev-lfs-20140408/Makefile.lfs install
```

```
root:/sources/eudev-3.2.1# make -f udev-lfs-20140408/Makefile.lfs install
mkdir: created directory '/usr/share/doc/udev-20140408'
mkdir: created directory '/usr/share/doc/udev-20140408/lfs'
'udev-lfs-20140408/55-lfs.rules' -> '/etc/udev/rules.d/55-lfs.rules'
'udev-lfs-20140408/81-cdrom.rules' -> '/etc/udev/rules.d/81-cdrom.rules'
'udev-lfs-20140408/83-cdrom-symlinks.rules' -> '/etc/udev/rules.d/83-cdrom-symlinks.rules'
'udev-lfs-20140408/write_cd_rules' -> '/lib/udev/write_cd_rules'
'udev-lfs-20140408/write_net_rules' -> '/lib/udev/write_net_rules'
'udev-lfs-20140408/init-net-rules.sh' -> '/lib/udev/init-net-rules.sh'
'udev-lfs-20140408/rule_generator.functions' -> '/lib/udev/rule_generator.functions'
'udev-lfs-20140408/README' -> '/usr/share/doc/udev-20140408/lfs/README'
'udev-lfs-20140408/55-lfs.txt' -> '/usr/share/doc/udev-20140408/lfs/55-lfs.txt'
root:/sources/eudev-3.2.1#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf eudev-3.2.1
```

```
root:/sources/eudev-3.2.1# cd ..  
root:/sources# rm -rf eudev-3.2.1  
root:/sources# █
```

2.15.60 Util-linux-2.29.1

Käännösaika: 1,0 SBU

Asennettavat ohjelmat: *addpart, agetty, blkdiscard, blkid, blockdev, cal, cfdisk, chcpu, chrt, col, colcrt, colrm, column, ctrlaltdel, delpart, dmesg, eject, fallocate, fdformat, fdisk, findfs, findmnt, flock, fsck, fsck.cramfs, fsck.minix, fsfreeze, fstrim, getopt, hexdump, hwclock, i386, ionice, ipcmk, ipcrm, ipcs, isosize, kill, last, lastb (linkki last:iin), ldattach, linux32, linux64, logger, look, losetup, lsblk, lscpu, lsipc, lslocks, lslogins, mcookie, mesg, mkfs, mkfs.bfs, mkfs.cramfs, mkfs.minix, mkswap, more, mount, mountpoint, namei, nsenter, partx, pg, pivot_root, prlimit, raw, readprofile, rename, renice, resizepart, rev, rtcwake, script, scriptreplay, setarch, setsid, setterm, sfdisk, sulogin, swapon, swaponoff (linkki swapon:iin), swapon, switch_root, tailf, taskset, ul, umount, uname26, unshare, utmpdump, uuid, uuidgen, wall, wdctl, whereis, wipefs, x86_64 ja zramctl*

Asennettavat kirjastot: *libblkid.so, libfdisk.so, libmount.so, libsmartcols.so ja libuuid.so*

Asennettavat hakemistot: */usr/include/blkid, /usr/include/libfdisk, /usr/include/libmount, /usr/include/libsmartcols, /usr/include/uuid, /usr/share/doc/util-linux-2.29.1 ja /var/lib/hwclock*

FHS suosittelee käyttämään `/var/lib/hwclock` -hakemistoa sen sijaan, että käyttäisi tavallista `/etc` -hakemistoa `adjtime`-tiedoston säilytyspaikkana. Luodaan hakemisto:

```
mkdir -pv /var/lib/hwclock
```

```
root:/sources# mkdir -pv /var/lib/hwclock
mkdir: created directory '/var/lib/hwclock'
root:/sources#
```

Puretaan paketti ja siirytään purettuun hakemistoon:

```
tar xf util-linux-2.29.1.tar.xz
cd util-linux-2.29.1
```

```
root:/sources# tar xf util-linux-2.29.1.tar.xz
root:/sources# cd util-linux-2.29.1
root:/sources/util-linux-2.29.1#
```

Valmistellaan paketti kääntämistä varten:

```
./configure ADJTIME_PATH=/var/lib/hwclock/adjtime \
--docdir=/usr/share/doc/util-linux-2.29.1 \
--disable-chfn-chsh \
--disable-login \
--disable-nologin \
--disable-su \
--disable-setpriv \
--disable-runuser \
--disable-pylibmount \
--disable-static \
--without-python \
--without-systemd \
--without-systemdsystemunitdir
```

```
<configure ADJTIME_PATH=/var/lib/hwclock/adjtime \
> --docdir=/usr/share/doc/util-linux-2.29.1 \
> --disable-chfn-chsh \
> --disable-login \
> --disable-nologin \
> --disable-su \
> --disable-setpriv \
> --disable-runuser \
> --disable-pylibmount \
> --disable-static \
> --without-python \
> --without-systemd \
> --without-systemdsystemunitdir
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
```

Puretaan auki:

- `--disable` ja `--without` -asetukset estävät varoitukset käännöskomponenteista, jotka vaativat paketteja, joita ei ole LFS:ssä tai jotka ovat epä johdonmukaisia muiden pakettien asentamien ohjelmien kanssa.

Käännetään paketti:

```
make
```

```
root:/sources/util-linux-2.29.1# make
make all-recursive
make[1]: Entering directory '/sources/util-linux-2.29.1'
Making all in po
make[2]: Entering directory '/sources/util-linux-2.29.1/po'
```

Testien ajaminen root-käyttäjänä voi olla haitallista järjestelmälle. Sen vuoksi ajetaan testit nobody-käyttäjänä:

```
chown -Rv nobody .
su nobody -s /bin/bash -c "PATH=$PATH make -k check"
```

```
-----
All 186 tests PASSED
-----
```

tests/ts/ipcs/limits2 epäonnistuu, kun isäntäkone käyttää tuoretta kernalia. Tämän virheen voi huoletta jättää huomiotta.

Asennetaan paketti:

```
make install
```

```
root:/sources/util-linux-2.29.1# make install
make install-recursive
make[1]: Entering directory '/sources/util-linux-2.29.1'
Making install in po
make[2]: Entering directory '/sources/util-linux-2.29.1/po'
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf util-linux-2.29.1
```

```
root:/sources/util-linux-2.29.1# cd ..
root:/sources# rm -rf util-linux-2.29.1
root:/sources#
```

2.15.61 Man-DB-2.7.6.1

Käännösaika: 0,4 SBU

Asennettavat ohjelmat: *accessdb, apropos (linkki whatis:iin), catman, lexgrog, man, mandb, manpath ja whatis*

Asennettavat kirjastot: *libman.so ja libmandb.so*

Asennettavat hakemistot: */usr/lib/man-db, /usr/lib/tmpfiles.d, /usr/libexec/man-db ja /usr/share/doc/man-db-2.7.6.1*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf man-db-2.7.6.1.tar.xz
cd man-db-2.7.6.1
```

```
root:/sources# tar xf man-db-2.7.6.1.tar.xz
root:/sources# cd man-db-2.7.6.1
root:/sources/man-db-2.7.6.1#
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/usr \
            --docdir=/usr/share/doc/man-db-2.7.6.1 \
            --sysconfdir=/etc \
            --disable-setuid \
            --enable-cache-owner=bin \
            --with-browser=/usr/bin/lynx \
            --with-vgrind=/usr/bin/vgrind \
            --with-grap=/usr/bin/grap \
            --with-systemdtmpfilesdir=
```

```
< ./configure --prefix=/usr \
> --docdir=/usr/share/doc/man-db-2.7.6.1 \
> --sysconfdir=/etc \
> --disable-setuid \
> --enable-cache-owner=bin \
> --with-browser=/usr/bin/lynx \
> --with-vgrind=/usr/bin/vgrind \
> --with-grap=/usr/bin/grap \
> --with-systemdtmpfilesdir=
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
```

Puretaan auki:

- `--disable-setuid` → Estetään ohjelman uid-arvon muuttaminen toiseksi.
- `--enable-cache-owner=bin` → Tämä asettaa järjestelmänlaajuisten cache-tiedostojen omistajaksi käyttäjän nimeltä bin.
- `--with-...` → Näillä parametreilla asetetaan joitakin oletusohjelmia.

Käännetään paketti:

```
make
```

```
root:/sources/man-db-2.7.6.1# make
make all-recursive
make[1]: Entering directory '/sources/man-db-2.7.6.1'
Making all in docs
make[2]: Entering directory '/sources/man-db-2.7.6.1/docs'
make[2]: Nothing to be done for 'all'.
```

Ajetaan testit:

```
make check
```

```
=====
Testsuite summary for man-db 2.7.6.1
=====
# TOTAL: 24
# PASS: 24
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

```
=====
Testsuite summary for man-db 2.7.6.1
=====
# TOTAL: 8
# PASS: 8
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

```
=====
Testsuite summary for man-db 2.7.6.1
=====
# TOTAL: 9
# PASS: 9
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

```
=====
Testsuite summary for man-db 2.7.6.1
=====
# TOTAL: 11
# PASS: 11
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

Asennetaan paketti:

```
make install
```

```
root:/sources/man-db-2.7.6.1# make install
Making install in docs
make[1]: Entering directory '/sources/man-db-2.7.6.1/docs'
make[2]: Entering directory '/sources/man-db-2.7.6.1/docs'
make[2]: Nothing to be done for 'install-exec-am'
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf man-db-2.7.6.1
```

```
root:/sources/man-db-2.7.6.1# cd ..
root:/sources# rm -rf man-db-2.7.6.1
root:/sources#
```

2.15.62 Tar-1.29

Käännösaika: 3,2 SBU

Asennettavat ohjelmat: tar

Asennettava hakemisto: /usr/share/doc/tar-1.29

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf tar-1.29.tar.xz
cd tar-1.29
```

```
root:/sources# tar xf tar-1.29.tar.xz
root:/sources# cd tar-1.29
root:/sources/tar-1.29#
```

Valmistellaan paketti kääntämistä varten:

```
FORCE_UNSAFE_CONFIGURE=1 \
./configure --prefix=/usr \
            --bindir=/bin
```

```
root:/sources/tar-1.29# FORCE_UNSAFE_CONFIGURE=1 \
> ./configure --prefix=/usr \
>             --bindir=/bin
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
```

Puretaan auki:

- `FORCE_UNSAFE_CONFIGURE=1` → Pakottaa mknodin testin ajamisen pääkäyttäjänä. Yleisesti tämän testin ajamista pääkäyttäjän oikeuksin pidetään vaarallisena, mutta kun se ajetaan järjestelmässä, joka on vasta osittain rakennettu, pääkäyttäjäksi pakottaminen on OK.

Käännetään paketti:

```
make
```

```
root:/sources/tar-1.29# make
make all-recursive
make[1]: Entering directory '/sources/tar-1.29'
Making all in doc
make[2]: Entering directory '/sources/tar-1.29/doc'
```

Ajetaan testit (n. 1 SBU):

```
make check
```

```
## ----- ##
## Test results. ##
## ----- ##

188 tests were successful.
16 tests were skipped.
```

Asennetaan paketti:

```
make install
```

```
root:/sources/tar-1.29# make install
Making install in doc
make[1]: Entering directory '/sources/tar-1.29/doc'
make[2]: Entering directory '/sources/tar-1.29/doc'
make[2]: Nothing to be done for 'install-exec-am'
```

```
make -C doc install-html docdir=/usr/share/doc/tar-1.29
```

```
< -C doc install-html docdir=/usr/share/doc/tar-1.29
make: Entering directory '/sources/tar-1.29/doc'
  MAKEINFO tar.html
/bin/mkdir -p '/usr/share/doc/tar-1.29'
/bin/mkdir -p '/usr/share/doc/tar-1.29/tar.html'
/usr/bin/install -c -m 644 'tar.html'/* '/usr/share/doc/tar-1.29/tar.html'
make: Leaving directory '/sources/tar-1.29/doc'
root:/sources/tar-1.29#
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf tar-1.29
```

```
root:/sources/tar-1.29# cd ..  
root:/sources# rm -rf tar-1.29  
root:/sources# █
```

2.15.63 Texinfo-6.3

Käännösaika: 0,5 SBU

Asennettavat ohjelmat: *info, install-info, makeinfo (linkki [texi2any:yn](#)), pdftexi2dvi, pod2texi, texi2any, texi2dvi, texi2pdf ja texindex*

Asennettavat kirjastot: *XSParagraph.so*

Asennettavat hakemistot: */usr/share/texinfo ja /usr/lib/texinfo*

Ensimmäiseksi puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf texinfo-6.3.tar.xz
cd texinfo-6.3
```

```
root:/sources# tar xf texinfo-6.3.tar.xz
root:/sources# cd texinfo-6.3
root:/sources/texinfo-6.3#
```

Valmistellaan paketti kääntämistä varten:

```
./configure --prefix=/usr --disable-static
```

```
root:/sources/texinfo-6.3# ./configure --prefix=/usr --disable-static
configure: WARNING: unrecognized options: --disable-static
checking for a BSD-compatible install... /usr/bin/install -c
```

Puretaan auki:

- `--disable-static` → Ylätason `configure`-skripti valittaa, että tämä on tunnistamaton asetus, mutta `XSParagraph`:n `configure`-skripti tunnistaa sen ja käyttää sitä estääkseen staattisen `XSParagraph.a`:n asentamisen `/usr/lib/texinfo` -hakemistoon.

Käännetään paketti:

```
make
```

Ajetaan testit:

```
make check
```

Asennetaan paketti:

```
make install
```

```
root:/sources/texinfo-6.3# make install
Making install in gnulib/lib
make[1]: Entering directory '/sources/texinfo-6.3/gnulib/lib'
make install-recursive
make[2]: Entering directory '/sources/texinfo-6.3/gnulib/lib'
```

Asennetaan TeX-asennukseen kuuluvat komponentit (vapaaehtoista):

```
make TEXMF=/usr/share/texmf install-tex
```

```
root:/sources/texinfo-6.3# make TEXMF=/usr/share/texmf install-tex
cd doc && make TEXMF=/usr/share/texmf install-tex
make[1]: Entering directory '/sources/texinfo-6.3/doc'
test -n "/usr/share/texmf" || (echo "TEXMF must be set." >&2; exit 1)
/bin/sh /sources/texinfo-6.3/build-aux/install-sh -d /usr/share/texmf/tex/texinf
o /usr/share/texmf/tex/generic/epsf
/usr/bin/install -c -m 644 ./texinfo.tex /usr/share/texmf/tex/texinfo/texinfo.te
x
/usr/bin/install -c -m 644 ./epsf.tex /usr/share/texmf/tex/generic/epsf/epsf.tex
for f in txi-ca.tex txi-cs.tex txi-de.tex txi-en.tex txi-es.tex txi-fr.tex txi-h
u.tex txi-is.tex txi-it.tex txi-ja.tex txi-nb.tex txi-nl.tex txi-nn.tex txi-pl.t
ex txi-pt.tex txi-ru.tex txi-sr.tex txi-tr.tex txi-uk.tex; do \
  /usr/bin/install -c -m 644 ./$f /usr/share/texmf/tex/texinfo/$f; done
make[1]: Leaving directory '/sources/texinfo-6.3/doc'
root:/sources/texinfo-6.3#
```

Puretaan auki:

- `TEXMF=/usr/share/texmf` → `TEXMF`-muuttuja pitää sisällään TeX-puun rootin paikan siltä varalta, että TeX-paketti asennetaan myöhemmin.

Info-dokumentaatiojärjestelmä käyttää tekstitiedostoa, joka sisältää valikkoluette-
lon. Tiedoston polku on `/usr/share/info/dir`. Monien pakettien Makefiles-
tiedostojen satunnaiset ongelmat saattavat aiheuttaa sen, että tämä tiedosto me-
nee epäsynkkaan järjestelmään asennettujen info-sivujen kanssa. Mikäli
`/usr/share/info/dir`-tiedosto pitää joskus luoda uudelleen, seuraavat ko-
mmennot suorittavat sen:

```
pushd /usr/share/info
rm -v dir
for f in *
do install-info $f dir 2>/dev/null
done
popd
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..
rm -rf texinfo-6.3
```

```
root:/sources/texinfo-6.3# cd ..
root:/sources# rm -rf texinfo-6.3
root:/sources# █
```

2.15.64 Nano-2.6.3

Käännösaika: alle 0,1 SBU

Asennettavat ohjelmat: nano ja rnano (symbolinen linkki)

Asennettavat kirjastot: -

Asennettavat hakemistot: /usr/share/nano ja /usr/share/doc/nano-2.6.3

Jos haluat asentaa jonkin muun tekstieditorin kuin Nanon (esim. Vim), katso ohjeet täältä: <http://www.linuxfromscratch.org/blfs/view/8.0/postlfs/editors.html>

Ensimmäiseksi ladataan paketti.

```
cd /mnt/lfs/sources
wget https://www.nano-editor.org/dist/v2.6/nano-2.6.3.tar.xz
```

```
root@dc-jumalan_linukka:~# cd /mnt/lfs/sources/
root@dc-jumalan_linukka:/mnt/lfs/sources# wget https://www.nano-editor.org/dist/v2.6/nano-2.6.3.tar.xz
--2017-04-28 13:18:22-- https://www.nano-editor.org/dist/v2.6/nano-2.6.3.tar.xz
Resolving www.nano-editor.org (www.nano-editor.org)... 213.138.109.86
Connecting to www.nano-editor.org (www.nano-editor.org)|213.138.109.86|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1005448 (982K) [application/x-xz]
Saving to: 'nano-2.6.3.tar.xz'

100%[=====>] 1,005,448  1.32MB/s  in 0.7s
2017-04-28 13:18:25 (1.32 MB/s) - 'nano-2.6.3.tar.xz' saved [1005448/1005448]
root@dc-jumalan_linukka:/mnt/lfs/sources#
```

Puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf nano-2.6.3.tar.xz
cd nano-2.6.3
```

```
root@dc-jumalan_linukka:/mnt/lfs/sources# tar xf nano-2.6.3.tar.xz
root@dc-jumalan_linukka:/mnt/lfs/sources# cd nano-2.6.3
root@dc-jumalan_linukka:/mnt/lfs/sources/nano-2.6.3#
```

Ajetaan configure:

```
./configure --prefix=/usr \
            --sysconfdir=/etc \
            --enable-utf8 \
            --docdir=/usr/share/doc/nano-2.6.3
```

```
root@dc-jumalan_linukka:/mnt/lfs/sources/nano-2.6.3# ./configure --prefix=/usr
\
> --sysconfdir=/etc \
> --enable-utf8 \
> --docdir=/usr/share/doc/nano-2.6.3
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
```

Puretaan auki:

- `--enable-utf8` → Ottaa käyttöön unicode-tuen Nanossa.

Ajetaan make:

```
make
```

```
root@dc-jumalan_linukka:/mnt/lfs/sources/nano-2.6.3# make
make all-recursive
make[1]: Entering directory `/mnt/lfs/sources/nano-2.6.3'
Making all in doc
make[2]: Entering directory `/mnt/lfs/sources/nano-2.6.3/doc'
```

Asennetaan root-käyttäjänä:

```
make install
```

```
root@dc-jumalan_linukka:/mnt/lfs/sources/nano-2.6.3# make install
Making install in doc
make[1]: Entering directory `/mnt/lfs/sources/nano-2.6.3/doc'
Making install in man
make[2]: Entering directory `/mnt/lfs/sources/nano-2.6.3/doc/man'
```

```
install -v -m644 doc/nanorc.sample /etc
```

```
root@dc-jumalan_linukka:/mnt/lfs/sources/nano-2.6.3# install -v -m644 doc/nanorc
.sample /etc
'doc/nanorc.sample' -> '/etc/nanorc.sample'
root@dc-jumalan_linukka:/mnt/lfs/sources/nano-2.6.3#
```

```
install -v -m644 doc/texinfo/nano.html /usr/share/doc/nano-2.6.3
```

```
root@dc-jumalan_linukka:/mnt/lfs/sources/nano-2.6.3# install -v -m644 doc/texinfo/nano.html /usr/share/doc/nano-2.6.3  
'doc/texinfo/nano.html' -> '/usr/share/doc/nano-2.6.3/nano.html'  
root@dc-jumalan_linukka:/mnt/lfs/sources/nano-2.6.3#
```

Lopuksi poistetaan purettu hakemisto:

```
rm -rf nano-2.6.3
```

```
root:/sources# rm -rf nano-2.6.3  
root:/sources#
```

2.15.64.1 Nano:n konfigurointi

Luodaan Nano:n konfiguraatiodosto:

```
cat > /etc/nanorc << "EOF"
" Begin /etc/nanorc

set autoindent
set const
set fill 72
set historylog
set multibuffer
set nohelp
set regexp
set smooth
set suspend

" End /etc/nanorc
EOF
```

```
root@dc-jumalan_linukka:/mnt/lfs/sources/nano-2.6.3# cat > /etc/nanorc << "EOF"
> " Begin /etc/nanorc
>
> set autoindent
> set const
> set fill 72
> set historylog
> set multibuffer
> set nohelp
> set regexp
> set smooth
> set suspend
>
> " End /etc/nanorc
> EOF
root@dc-jumalan_linukka:/mnt/lfs/sources/nano-2.6.3# █
```

2.16 Debuggaussymbolit

Useimmat ohjelmat ja kirjastot käännetään oletuksena debuggaussymbolien kanssa. Kun debugataan ohjelmaa tai kirjastoa, joka on käännetty debuggausinformaation kanssa, debugger voi tarjota memory addressien lisäksi rutiinien ja muuttujien nimet.

Debuggaussymbolien sisällyttäminen kuitenkin laajentaa ohjelmaa tai kirjastoa huomattavasti. Alla on esimerkkejä siitä, kuinka paljon tilaa nämä symbolit vievät:

- bash-binääri debuggaussymbolien kanssa: 1200 KB
- bash-binääri ilman debuggaussymboleja: 480 KB
- Glibc ja GCC -tiedostot (`/lib` ja `/usr/lib`) debuggaussymbolien kanssa: 87 MB
- Glibc ja GCC -tiedostot ilman debuggaussymboleita: 16 MB

Koot voivat vaihdella riippuen siitä, mitä kääntäjää ja C-kirjastoa on käytetty.

Koska useimmat käyttäjät eivät koskaan käytä debuggeria, paljon tilaa säästetään poistamalla nämä symbolit. Seuraavassa osiossa poistetaan kaikki debuggaussymbolit.

2.17 Stripping

Mikäli järjestelmän käyttäjä ei ole ohjelmoija tai ei muuten aio debugata, järjestelmän tilaa voi vapauttaa noin 90 MB poistamalla debuggaussymbolit binääreistä ja kirjastoista. Tämä ei aiheuta muuta haittaa kuin, että ohjelmiston debuggaus ei täysin enää onnistu.

Koska typon tekeminen on helppoa, tässä vaiheessa kannattaa ottaa virtuaaliympäristössä järjestelmästä varmuuskopio. Ei sitten haittaa, jos vahingossa typottaa uuden järjestelmän käyttökelvottomaksi.

Ennen kuin toteutetaan stripping, kannattaa huolellisesti varmistaa, että yhtäkään niistä binääreistä, jotka aiotaan stripata, ei suoriteta kyseisellä hetkellä. Jos on yhtään epävarma siitä, siirtyikö aiemmin onnistuneesti chroot-ympäristöön, kannattaa varmuuden vuoksi nyt kirjautua ulos ja takaisin sisään:

```
logout
chroot $LFS /tools/bin/env -i \
    HOME=/root TERM=$TERM PS1='\u:\w\$ ' \
    PATH=/bin:/usr/bin:/sbin:/usr/sbin \
    /tools/bin/bash --login
```

Stripataan binäärit ja kirjastot:

```
/tools/bin/find /usr/lib -type f -name \*.a \
    -exec /tools/bin/strip --strip-debug {} ';'

/tools/bin/find /lib /usr/lib -type f -name \*.so* \
    -exec /tools/bin/strip --strip-unneeded {} ';'

/tools/bin/find /{bin,sbin} /usr/{bin,sbin,libexec} -type f \
    -exec /tools/bin/strip --strip-all {} ';'

```

```
root:/sources# /tools/bin/find /usr/lib -type f -name \*.a \  
> -exec /tools/bin/strip --strip-debug {} ';' \  
/tools/bin/strip:/usr/lib/libm.a: File format not recognized \  
root:/sources# \  
root:/sources# /tools/bin/find /lib /usr/lib -type f -name \*.so* \  
> -exec /tools/bin/strip --strip-unneeded {} ';' \  
/tools/bin/strip:/usr/lib/libpthread.so: File format not recognized
```

Monista tiedostoista raportoidaan, että niiden tiedostomuotoa ei tunneta. Nämä varoitukset voi turvallisesti ignorata. Se tarkoittaa vain, että ne tiedostot ovat skriptejä eivätkä binäärejä.

2.18 Siistitään

Siivotaan pois joitakin testien ajamisesta jääneitä ylimääräisiä tiedostoja:

```
rm -rf /tmp/*
```

```
root:/sources# rm -rf /tmp/*  
root:/sources#
```

Tästä lähtien, kun siirrytään uudelleen chroot-ympäristöön siitä poistumisen jälkeen, käytetään seuraavaa muokattua chroot-komentoa:

```
chroot "$LFS" /usr/bin/env -i \\\n    HOME=/root TERM="$TERM" PS1='\\u:\\w\\$ ' \\\n    PATH=/bin:/usr/bin:/sbin:/usr/sbin \\\n    /bin/bash --login
```

Tämä johtuu siitä, että ohjelmia `/tools`-hakemistossa ei enää tarvita. Voit halutessasi poistaa `/tools`-hakemiston. Huomaa, että `/tools`-hakemiston poistaminen poistaa myös Tcl:n, Expectin ja dejaGNU:n tilapäiset kopiot, joita käytettiin työkaluketjun testien ajamiseen. Jos tarvitset näitä ohjelmia myöhemmin, ne täytyy kääntää uudelleen ja asentaa. [BLFS](#)-kirjasta löytyy ohjeet siihen.

Mikäli kernelin virtuaaliset tiedostojärjestelmät on irrotettu joko manuaalisesti tai uudelleen käynnistykseen vuoksi, varmista, että ne on liitetty, kun siirrytään chroot-ympäristöön. Tämä prosessi on selitetty kappaleissa [2.11.2 /dev -hakemiston liittäminen ja täyttäminen](#) sekä [2.11.3 Kernelin virtuaalisten tiedostojärjestelmien liittäminen](#).

Aiemmin tässä kappaleessa jäi useita staattisia kirjastoja vaimentamatta, jotta tyydytettiin joidenkin pakettien regressiotestien tarpeet. Nämä kirjastot ovat seuraavista paketeista: `binutils`, `bzip2`, `e2fsprogs`, `flex`, `libtool` ja `zlib`. Voit halutessasi poistaa ne nyt:

```
rm -f /usr/lib/lib{bfd,opcodes}.a  
rm -f /usr/lib/libbz2.a
```

```
rm -f /usr/lib/lib{com_err,e2p,ext2fs,ss}.a
rm -f /usr/lib/libltdl.a
rm -f /usr/lib/libfl.a
rm -f /usr/lib/libfl_pic.a
rm -f /usr/lib/libz.a
```

```
root:/sources# rm -f /usr/lib/lib{bfd,opcodes}.a
root:/sources# rm -f /usr/lib/libbz2.a
root:/sources# rm -f /usr/lib/lib{com_err,e2p,ext2fs,ss}.a
root:/sources# rm -f /usr/lib/libltdl.a
root:/sources# rm -f /usr/lib/libfl.a
root:/sources# rm -f /usr/lib/libfl_pic.a
root:/sources# rm -f /usr/lib/libz.a
root:/sources# █
```

3 JÄRJESTELMÄN KONFIGUROIMINEN

Linux-järjestelmän käynnistyminen pitää sisällään monia vaiheita. Prosessin täytyy liittää sekä virtuaaliset että todelliset tiedostojärjestelmät, alustaa laitteet, aktioida swap, tarkistaa tiedostojärjestelmien eheys, liittää mahdolliset swap-partitiot tai -tiedostot, asettaa järjestelmän kello, asettaa verkkoasetukset, käynnistää järjestelmän vaatimat taustaprosessit ja suorittaa kaikki muut käyttäjän tarvitsemat ja määrittelemät tehtävät. Tämä prosessi pitää organisoida huolellisesti, jotta varmistetaan, että kaikki työtehtävät suoritetaan oikeassa järjestyksessä, mutta mahdollisimman nopeasti.

3.1 System V

System V on klassinen käynnistysprosessi, jota on käytetty Unixissa ja Unixin kaltaisissa järjestelmissä, kuten Linuxissa jo vuodesta 1983. Se sisältää pienen ohjelman nimeltään `init`, joka käynnistää perusohjelmat, kuten `login` (gettyn kautta) ja ajaa skriptin. Tämä skripti, yleensä nimeltään `rc`, kontrolloi joidenkin skriptien suorittamista, jotka puolestaan suorittavat järjestelmän alustamiseen vaadittavat tehtävät.

`/etc/inittab` -tiedosto kontrolloi `init`-ohjelmaa ja se on organisoitu run leveleihin, joissa käyttäjä voi käyttöjärjestelmää suorittaa:

Run level	Kuvaus
0	Halt
1	Single user mode
2	Multiuser, without networking
3	Full multiuser mode
4	<i>User defined mode</i>
5	Full multiuser mode with display manager
6	Reboot

Yleisimmät run levelit ovat 3 ja 5. Tähän järjestelmään liittyvät seuraavat

Edut:

- Vakiintunut, hyvin ymmärretty järjestelmä.
- Helppo kustomoida.

Ja haitat:

- Hitaampi käynnistymään. LFS-järjestelmältä menee 8-12 sekuntia, kun käynnistymisaika mitataan ensimmäisestä kernel-viestistä kirjautumiskehotteeseen. Nettiyhteys saavutetaan yleensä noin 2 sekuntia kirjautumiskehotteeseen jälkeen.
- Käynnistymistehtävien sarjamuotoinen prosessointi. Tämä liittyy edelliseen kohtaan. Viivytys missä tahansa prosessissa, kuten tiedostojärjestelmän tarkastuksessa viivästyttää koko käynnistymisprosessia.
- Ei suoraan tue edistyneitä toimintoja, kuten `cgroups`.

3.2 LFS-Bootscripts-20150222

Kuvaus: Sisältää setin skriptejä LFS-järjestelmän käynnistämiseen/pysäyttämiseen boot-tupissa/sammutuksessa. Konfiguraatiotiedostot ja proseduurit, joita tarvitaan boot-prosessin kustomoimiseen, on kuvailtu seuraavissa osioissa.

Käännösaika: alle 0,1 SBU

Vaadittava levytila: 244 KB

Asennettavat skriptit: checkfs, cleanfs, console, functions, halt, ifdown, ifup, localnet, modules, mountfs, mountvirtfs, network, rc, reboot, sendsignals, setclock, ipv4-static, swap, sysctl, sysklogd, template, udev ja udev_retry

Asennettavat hakemistot: /etc/rc.d, /etc/init.d (symbolic link), /etc/sysconfig, /lib/services ja /lib/lsb (symbolic link)

Tämän paketin sisältämät skriptit ovat myös liitteissä 7-33.

Ensin puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf lfs-bootscripts-20150222.tar.bz2
cd lfs-bootscripts-20150222
```

```
root:/sources# tar xf lfs-bootscripts-20150222.tar.bz2
root:/sources# cd lfs-bootscripts-20150222
root:/sources/lfs-bootscripts-20150222#
```

Asennetaan paketti:

```
make install
```

```
root:/sources/lfs-bootscripts-20150222# make install
install -d -m 755 /etc/rc.d/rc0.d
install -d -m 755 /etc/rc.d/rc1.d
install -d -m 755 /etc/rc.d/rc2.d
install -d -m 755 /etc/rc.d/rc3.d
install -d -m 755 /etc/rc.d/rc4.d
```

Lopuksi poistetaan purettu hakemisto:

```
cd ..  
rm -rf lfs-bootscripts-20150222
```

```
root:/sources/lfs-bootscripts-20150222# cd ..  
root:/sources# rm -rf lfs-bootscripts-20150222  
root:/sources# █
```

3.3 Laitteiden ja moduulien käsittely

Kappaleessa 3 asennettiin Udev-paketti, kun eudev käännettiin. Ennen kuin menemme yksityiskohtiin, pieni katsaus laitteiden käsittelymetodien historiaan on paikallaan.

Linux-järjestelmät käyttivät perinteisesti staattista laitteenluomismetodia, jonka mukaan suuri määrä laitetiedostoja luotiin `/dev`-hakemiston alle (puhutaan tuhansista tiedostoista). Tämä tehtiin riippumatta siitä, oliko vastaavia laitteita edes olemassa. Laitetiedostot luotiin tavallisesti `MAKEDEV`-skriptin avulla, joka sisältää useita kutsuja `mknod`-ohjelmalle oleellisten major- ja minor-laitenumeroiden kanssa kaikkia mahdollisia laitteita varten, joita maailmassa saattaa olla.

Nykyaikaisempaa udev-metodia käyttämällä vain niille laitteille, jotka kernel tunnistaa, luodaan laitetiedosto. Koska laitetiedostot luodaan joka kerta järjestelmän käynnistymisen yhteydessä, ne tallennetaan `devtmpfs`-tiedostojärjestelmään (virtuaalinen tiedostojärjestelmä, joka sijaitsee kokonaisuudessaan järjestelmän muistissa). Laitetiedostot eivät vie paljoa tilaa, joten niihin käytetyn muistin määrä on olematon.

3.3.1 Pari sanaa moduuleista

Kernelin moduulit ovat ytimeen ladattavia osia, jotka voidaan ajon aikana liittää Linuxin ytimeen. Moduuleilla voidaan lisätä kernelin toiminnallisuutta ilman, että järjestelmää tarvitsee käynnistää uudelleen. Moduulit sijaitsevat hakemistossa `/lib/modules/<ytimen_versio>/`. Ladattaville moduuleille voidaan määrittellä asetuksia `/etc/modprobe.conf` -asetustiedostossa tai vapaavalintaisissa tiedostoissa `/etc/modprobe.d/` -hakemiston alla. Ohjelma, jolla voi ladata tai poistaa moduuleja ajon aikana, on nimeltään `modprobe`.

3.3.2 Historia

Helmikuussa 2000 uusi tiedostojärjestelmä nimeltään `devfs` sisällytettiin kernel-versioon 2.3.46. Siitä huolimatta, että se sisällytettiin kernelin lähdekoodiin, tämä metodi laitteiden dynaamiseen luomiseen ei ikinä saanut kernelin kehittäjien varauksetonta tukea.

Isoin ongelma `devfs`:ssä oli tapa, jolla se käsitteli laitteiden tunnistamisen, luomisen ja nimeämisen. Laitetiedostojen nimeämiseen liittyvä ongelma oli ehkäpä kaikkein kriittisin. Jos laitteiden nimiä on mahdollista muokata, nimeämispolitiikan pitäisi olla järjestelmän pääkäyttäjän käsissä sen sijaan, että kehittäjä määrää sen. Tämän ja muiden ongelmien vuoksi (mm. ylläpidon puute) `devfs` merkattiin käytöstä poistuneeksi. Lopulta se poistettiin kernelistä kesäkuussa 2006.

Kernel-versioista 2.5 lähtien esiteltiin uusi tiedostojärjestelmä nimeltään `sysfs`. `sysfs`:n työ on tuottaa näkymä järjestelmän laitekonfiguraatiosta [käyttäjätilan](#) prosesseille.

3.3.3 Udev-toteutus

Tässä kappaleessa tutustutaan udev:iin tarkemmin.

3.3.3.1 Sysfs

`sysfs`-tiedostojärjestelmä mainittiin lyhyesti aiemmin. Joku saattaa ihmetellä, miten `sysfs` tietää järjestelmässä läsnä olevat laitteet ja mitä laitenumeroita tulisi käyttää niitä varten. Ajurit, jotka on käännetty suoraan kerneliin, rekisteröivät objektinsa `sysfs:n` avulla (`devtmpfs` sisäisesti) kun kernel tunnistaa ne. Ajureille, jotka käännetään moduuleina, tämä rekisteröinti tapahtuu moduuleja ladattaessa. Kun `sysfs`-tiedostojärjestelmä (`/sys`) on liitetty, data, jonka ajurit rekisteröivät `sysfs:n` avulla, on saatavilla käyttäjätilan prosesseille ja `udev`:lle prosessointia ja mahdollista muokkausta varten.

3.3.3.2 Laitetiedostojen luominen

Kernelin `devtmpfs`-tiedostojärjestelmä luo laitetiedostot. Kaikki ajurit, jotka haluavat rekisteröidä laitetiedoston, menevät `devtmpfs:n` läpi. Kun `devtmpfs`-instanssi on liitetty `/dev`-hakemistoon, laitetiedosto luodaan nimen, oikeuksien ja omistajatiedon kera.

Tämän jälkeen kernel lähettää `uevent`:in `udev`:hen. Perustuen sääntöihin, jotka on määritelty tiedostoissa `/etc/udev/rules.d`, `/lib/udev/rules.d` ja `/run/udev/rules.d`, `udev` luo symboliset linkit laitetiedostoihin, päivittää oikeudet, omistajan tai ryhmän tai muokkaa sisäistä `udev`-tietokantakirjausta (nimi) tälle objektille.

Säännöt näissä kolmessa hakemistossa on numeroitu ja kaikki kolme hakemistoa käsitellään kuin ne olisivat yksi hakemisto. Jos `udev` ei löydä sääntöä laitteelle, jota se on luomassa, se jättää oikeudet ja omistajuuden niihin arvoihin, mitä `devtmpfs` käytti.

3.3.3.3 Moduulin lataaminen

Laiteajureilla, jotka käännetään moduuleiksi voi olla aliaksia rakennettuna niihin. Aliakset voi tulostaa modinfo komennolla ja ne liittyvät yleensä moduulin tukemien laitteiden väyläkohtaisiin tunnisteluihin. Esimerkkinä voisi olla `snd-fm801-ajuri`, joka tukee PCI-laitteita, joilla on toimittaja-ID `0x1319` ja laite-ID `0x0801`. Silloin alias olisi `pci:v00001319d00000801sv*sd*bc04sc01i*`. Useimmille laitteille väylän ajuri julkaisee ajurin aliaksen, joka olisi saatavilla `sysfs:n` kautta. Esimerkiksi, `/sys/bus/pci/devices/0000:00:0d.0/modalias` -tiedosto voisi sisältää merkijonon `pci:v00001319d00000801sv00001319sd00001319bc04sc01i00`. Udevin oletussäännöissä määritellään, että `udevd` kutsuu `/sbin/modprobe:a MODALIAS` uevent-ympäristömuuttujan kanssa (jonka pitäisi olla sama kuin `modalias`-tiedoston sisällön `sysfs:ssä`). Siksi `udevd` lataa kaikki moduulit, joiden aliakset täsmäävät tähän merkkijonoon jokerimerkkilajennuksen jälkeen.

Tässä esimerkissä se tarkoittaisi sitä, että `snd-fm801:n` lisäksi vanhentunut (ja ei-toivottu) `forte-ajuri` ladattaisiin, jos se on järjestelmässä saatavilla. Alempana selitetään, miten ei-toivottujen ajurien lataamisen voi estää.

Kerneliin voidaan ladata myös moduuleja esimerkiksi verkkoprotokollille, tiedostojärjestelmille ja NLS-tuelle tarpeen vaatiessa.

3.3.3.4 Hotpluggable/Dynamic -laitteiden käsittely

Kun tietokoneeseen kytketään esimerkiksi USB MP3 -soitin, kernel tunnistaa, että laite on nyt kytketty ja luo `uevent:in`. `Udevd` käsittelee tämän `uevent:n` yllä mainitulla tavalla.

3.3.4 Ongelmat moduulien lataamisessa ja laitteiden luomisessa

Seuraavaksi käsitellään joitakin mahdollisia ongelmia automaattisessa laitetiedostojen luomisessa.

3.3.4.1 Kernel-moduulia ei ladata automaattisesti

Udev lataa moduulin ainoastaan, jos sillä on väyläkohtainen alias ja väylän ajuri julkaisee tarpeelliset aliakset sysfs:lle. Muissa tapauksissa moduulin lataaminen pitää toteuttaa muilla tavoilla. Linux-4.9.9:ssä Udevin tiedetään lataavan oikein kirjoitetut ajurit INPUT-, IDE-, PCI-, USB-, SCSI-, SERIO- ja FireWire-laitteille.

Määrittääksesi, onko tarvitsemallasi laiteajurilla tukea Udeville, aja `modinfo` komento ja anna sille argumentiksi halutun moduulin nimi. Tämän jälkeen tutki `/sys/bus` laitehakemisto ja tarkista onko siellä kyseistä modalias-tiedostoa.

Mikäli modalias-tiedosto on olemassa sysfs:ssä, ajuri tukee laitetta ja laite voi kommunikoida järjestelmän kanssa. Jos aliasta ei löydy, ajuri ei tuo toteutusta tai siinä on bugi. Lataa ajuri ilman Udevia ja oletta, että ongelma korjataan myöhemmin.

Mikäli modalias-tiedostoa ei ole `/sys/bus` hakemistossa, tämä tarkoittaa, että kernelin kehittäjät eivät ole vielä lisänneet modalias-tukea tälle väylätyypille. Linux-4.9.9:ssä tämä ongelma ilmenee vanhojen ISA-väylien kanssa. Oletta, että tämä ongelma korjataan myöhemmissä kernel-versioissa.

Udev ei lataa "wrapper"-ajureita (kuten `snd-pcm-oss`) eikä non-hardware -ajureita (kuten `loop`).

3.3.4.2 Kernel-moduulia ei ladata automaattisesti ja Udevin ei ole tarkoituskaan ladata sitä

Jos "wrapper"-moduuli vain parantaa jonkin toisen moduulin tarjoamaa toiminallisuutta (esim. `snd-pcm-oss` parantaa `snd-pcm:n` toiminnallisuutta tekemällä äänikortit saataville OSS-sovelluksille), aseta `modprobe` latamaan wrapper-moduuli

sen jälkeen, kun Udev on ladannut alkuperäisen moduulin. Tehdäksesi tämän, lisää "softdep"-rivi (kutsutaan sitä vaikka löyhäksi riippuvuudeksi) mihin tahansa `/etc/modprobe.d/<filename>.conf` -tiedostoon. Esimerkiksi:

```
softdep snd-pcm post: snd-pcm-oss
```

Huomaa, että "softdep"-parametri myös sallii `pre:` -riippuvuudet, tai sekoituksen sekä `pre:` että `post:` riippuvuuksia. `modprobe.d(5)` -manuaalisivusta löytyy lisää infoa "softdep":n syntaksista ja kyvyistä.

Jos ei ole wrapper moduuli, konfiguroi `modules-bootskripti` lataamaan moduuli järjestelmän käynnistymisen yhteydessä. Tämä tapahtuu lisäämällä moduulin nimi `/etc/sysconfig/modules` -tiedostoon omalle rivilleen.

3.3.4.3 Udev lataa ei-toivottuja moduuleja

Estä ei toivotun moduulin latautuminen lisäämällä se mustalle listalle `/etc/modprobe.d/blacklist.conf` -tiedostoon. Tehdään tämä esimerkiksi `forte`-moduulille:

```
blacklist forte
```

Mustalle listalle laitettut moduulit voidaan silti tarvittaessa ladata myöhemmin käsin `modprobe`-komennolla.

3.3.4.4 Udev luo laitteen virheellisesti tai tekee väärän symbolisen linkin

Tämä tapahtuu yleensä, jos sääntö täsmää laitteeseen, johon sen ei kuulunut täsmätä. Esimerkiksi huonosti kirjoitettu sääntö saattaa täsmätä sekä SCSI-levyyn (kuten halutaankin) että laitevalmistajan vastaavaan geneeriseen SCSI-laitteeseen (jota ei haluta). Etsi ongelman aiheuttava sääntö ja täsmennä sitä `udevadm info` -komennon avulla.

3.3.4.5 Udev-sääntö toimii epäluotettavasti

Tämä saattaa olla edellisen ongelman toinen ilmentymä. Mikäli näin ei ole, ja sääntösi käyttää `sysfs`-attribuutteja, kyseessä saattaa olla kernelin ajoitusongelma, joka paikataan seuraavissa kernel-versioissa. Tilapäisesti ongelman voi kiertää luomalla säännön, joka odottaa käytettyä `sysfs`-attribuuttia ja lisää sen `/etc/udev/rules.d/10-wait_for_sysfs.rules` -tiedoston loppuun. Jos tätä tiedostoa ei ole olemassa, luo sellainen.

3.3.4.6 Udev ei luo laitetta

Oletetaan, että ajuri on käännetty staattisesti kerneliin tai on jo ladattu moduulina, ja ettei Udev luo väärin nimettyä laitetta.

Udevilla ei ole tietoa, jonka se tarvitsee luodakseen laitetiedoston, ellei kernelin ajuri julkaise sen dataa `sysfs`:ään. Tämä on yleisintä kernel-puun ulkopuolisten ajureiden (kolmannen osapuolen kehittämät) kanssa. Tämmöisessä tapauksessa luo staattinen laitetiedosto `/lib/udev/devices` -hakemistoon oikeiden `major/minor` -numeroiden kanssa. Tiedot voi tarkistaa `devices.txt` -tiedostosta kernelin dokumentaatiosta tai kolmannen osapuolen ajurin dokumentaatiosta. Udev kopioi staattisen laitetiedoston `/dev` -hakemistoon.

3.3.4.7 Laitteiden nimeämisjärjestys vaihtuu satunnaisesti käynnistymisen jälkeen

Tämä johtuu siitä, että Udev käsittelee `ueventit` ja lataa moduulit rinnan, ja siten ennalta määrittelemättömässä järjestyksessä. Tämä on asia, jota ei ikinä "korjata". Eli ei voida luottaa siihen, että kernelin laitenimet olisivat pysyviä. Sen sijaan kannattaa luoda omat säännöt, jotka tekevät symboliset linkit pysyvien nimien kanssa, jotka perustuvat laitteen pysyviin attribuutteihin, kuten sarjanumeroon tai Udevin asentamien `*_id` -työkalujen tulosteisiin. Kappaleissa 3.4 ja 3.5 löytyy esimerkkejä tästä.

3.3.5 Hyödyllistä lukemista

Seuraavilta sivuilta löytyy lisää dokumentaatiota aiheesta:

- Userspace implementation of devfs:
http://www.kroah.com/linux/talks/ols_2003_udev_paper/Reprint-Kroah-Hartman-OLS2003.pdf
- Sysfs-tiedostojärjestelmä: <http://www.kernel.org/pub/linux/kernel/people/mochel/doc/papers/ols-2005/mochel.pdf>

3.4 Laitteiden hallinta

Tässä kappaleessa käydään läpi laitteiden hallintaa.

3.4.1 Verkkolaitteet

Udev nimeää oletuksena verkkolaitteet Firmware/BIOS –datan tai fyysisten piirteiden, kuten väylän, slotin tai MAC-osoitteen perusteella. Tämän nimeämistavan tarkoituksena on varmistaa, että verkkolaitteet nimetään johdonmukaisesti, eikä perustuen aikaan, jolloin verkkokortti rekisteröidään järjestelmään. Otetaan esimerkiksi tietokone, jossa on kaksi verkkokorttia. Toinen on Intelin ja toinen Realtekin. Intelin kortti tunnistetaan nimellä `eth0` ja Realtekin kortti `eth1`. Joissain tapauksissa uudelleen käynnistyksen jälkeen kortit numeroidaan päinvastoin.

Uudella nimeämistavalla verkkolaitteiden nimet voisivat olla vaikka `enp5s0` tai `wlp3s0`. Jos tämä nimeämistapa ei ole toivottu, voi perinteisen tai kustomoidun nimeämistavan ottaa käyttöön.

3.4.1.1 Pysyvän nimeämistavan poistaminen käytöstä kernelin komentorivillä

Perinteinen nimeämistapa, joka käyttää `eth0`, `eth1` jne. voidaan palauttaa lisäämällä `net.ifnames=0` kernelin komentorivillä. Tämä on paras käytäntö niissä järjestelmissä, joissa on vain yksi samantyyppinen ethernet-laite. Läppäreissä yleensä on monia ethernet-yhteyksiä, jotka on nimetty `eth0` ja `wlan0`. Komentorivikomento annetaan GRUB-konfiguraatiotiedostoon. Katso osio [4.3.3 GRUB-konfiguraatiotiedoston luominen](#).

3.4.1.2 Kustomoitujen Udev-sääntöjen luominen

Nimeämiskäytäntöjä voidaan muokata luomalla omia Udev-sääntöjä. On olemassa skripti, joka luo alustavat säännöt. Luodaan nämä säännöt:

```
bash /lib/udev/init-net-rules.sh
```

Tutki `/etc/udev/rules.d/70-persistent-net.rules` -tiedostoa löytääksesi mikä nimi on annettu millekin verkkolaitteelle:

```
cat /etc/udev/rules.d/70-persistent-net.rules
```

HUOM! Joissakin tapauksissa, kuten silloin, jos MAC-osoitteet on annettu verkkokortille manuaalisesti tai virtuaalisessa ympäristössä, kuten Qemu tai Xen, verkko-sääntöjen tiedostoa ei välttämättä luoda, koska osoitteita ei ole johdonmukaisesti annettu. Sellaisessa tapauksessa tätä metodia ei voi hyödyntää.

Tiedosto alkaa kommenttiosioilla, jota seuraa kaksi riviä jokaiselle NICille. Ensimmäinen rivi jokaiselle NICille on kuvaus, jossa on laite-ID ja ajuri suluissa, jos ajuri löydetään. Laite-ID:tä tai ajuria ei käytetä määrittämään mikä nimi annetaan rajapinnalle; tämä info on vain viitteeksi. Toinen rivi on Udev-sääntö, joka täsmää verkkokorttiin ja määrittää laitteen nimen.

Kaikki Udev-säännöt koostuvat useista avaimista, erotettuna pilkuilla ja valinnaisella välilyönnillä. Tämän säännön avaimet ja niiden selitykset ovat seuraavat:

- `SUBSYSTEM=="net"` → Käskee Udevin jättää huomiotta kaikki laitteet, jotka eivät ole verkkokortteja.
- `ACTION=="add"` → Käskee Udevin jättää huomiotta tämä sääntö ueventille, joka ei ole add ("remove" ja "change" ueventit myös tapahtuvat, mutta niiden ei tarvitse nimetä uudelleen verkkoliitäntöjä).
- `DRIVERS=="?*"` → Tämä on olemassa, jotta Udev jättää huomiotta VLAN tai bridge sub-interfacet (koska näillä sub-interfaceilla ei ole ajureita). Nämä sub-interfacet ohitetaan, koska niille annettava nimi olisi sama kuin isäntälaitteiden nimi.

- `ATTR{address}` → Tämän avaimen arvo on NIC:n MAC-osoite.
- `ATTR{type}=="1"` → Tämä varmistaa, että sääntö täsmää ainoastaan ensisijaiseen rajapintaan siinä tapauksessa, että on joitakin langattomia ajureita, jotka luovat monia virtuaalisia rajapintoja. Toissijaiset rajapinnat ohitetaan samasta syystä, kuin VLAN ja bridge sub-interfacet ohitetaan: muuten olisi samoista nimistä johtuva nimikonflikti.
- `NAME` → Tämän avaimen arvo on nimi, jonka Udev antaa tälle rajapinnalle.

`NAME`:n arvo on tärkeä juttu. Varmista, että tiedät, mikä nimi on annettu millekin verkkokortille ennen kuin etenet. Varmista myös, että käytät tätä `NAME`-arvoa kun luot konfiguraatiotiedostoja alempana.

3.4.2 CD-ROM –symbolinen linkki

Jotkin ohjelmat, jotka kenties halutaan asentaa myöhemmin (esim. monet mediasoittimet) olettavat, että `/dev/cdrom` ja `/dev/dvd` symboliset linkit ovat olemassa ja osoittavat CD-ROM tai DVD-ROM -laitteeseen. Voi myös olla kätevää laittaa niiden symboliset linkit viittaamaan `/etc/fstab`:iin. Udevin mukana tulee skripti, joka generoi sääntötiedostot, jotka luovat nämä symboliset linkit puolestasi riippuen liitettyjen laitteiden ominaisuuksista. Käyttäjän on päätettävä, kumpaa kahdesta toimintatavasta skriptin halutaan käyttävän.

Skripti voi operoida `"by-path"` -tilassa (käytetään oletuksena USB ja FireWire -laitteille), jossa sen luomat säännöt riippuvat fyysisestä polusta CD- tai DVD-laitteeseen. Toinen tila on `"by-id"`, jota käytetään oletuksena IDE- ja SCSI-laitteille. Tässä tilassa skriptin luomat säännöt riippuvat tunnistemerkkijonoista, jotka on tallennettu CD- tai DVD-laitteeseen itseensä. Polku määritetään Udevin `path_id` -skriptillä, ja tunnistemerkkijonot luetaan laitteistosta sen `ata_id` tai `scsi_id` -ohjelmilla, riippuen siitä, minkä tyyppinen laite on kyseessä.

Molemmissa lähestymistavoissa on puolensa; soveltuvampi tapa riippuu siitä, minkälaisia laitemuutoksia on odotettavissa. Jos on oletettavissa, että laitteen fyysinen polku (portit ja/tai slotit, joihin se on kytketty) vaihtuu (jos esimerkiksi levy vaihdetaan eri IDE-porttiin tai eri USB-liitimeen), pitäisi käyttää `"by-id"` -tilaa. Toisaalta jos laitteen tunnistetieto muuttuu (esimerkiksi jos se menee rikki ja se korvataan eri laitteella, jolla kuitenkin on samat ominaisuudet ja joka on kytketty samoihin liittimiin), pitäisi käyttää `"by-path"` -tilaa.

Jos molemmat muutokset ovat mahdollisia, valitse tila sen mukaan, kumpi tapahtuu useammin.

HUOM! Ulkoisten laitteiden (esim. USB:llä liitettävä CD-asema) ei pitäisi käyttää `by-path` -tilaa, koska joka kerta kun laite kytketään uuteen ulkoiseen porttiin, sen fyysinen polku vaihtuu. Kaikilla ulkoisesti kytkettävillä laitteilla on sama ongelma, jos kirjoitat Udev-säännöt siten, että ne tunnistavat laitteet fyysisen polun avulla. Tämä koskee siis muitakin kuin vain CD- ja DVD-laitteita.

Jos haluat nähdä, mitä arvoja Udev-skriptit käyttävät, etsi CD-ROM -laitteelle vastaava hakemisto `/sys:n` alta (esim. `/sys/block/hdd`) ja aja seuraava komento:

```
udevadm test /sys/block/hdd
```

Tutki rivejä, jotka sisältävät monta `*_id` -ohjelmatulostetta. "by-id" -tila käyttää `ID_SERIAL` -arvoa, jos se on olemassa eikä ole tyhjä. Muuten se käyttää `ID_MODEL`:in ja `ID_REVISION`:in kombinaatiota. "by-path" -tila käyttää `ID_PATH` -arvoa.

Jos oletustila ei vastaa tarpeitasi, voit tehdä seuraavan muutoksen `/etc/udev/rules.d/83-cdrom-symlinks.rules` -tiedostoon (jossa *mode* on joko "by-id" tai "by-path"):

```
sed -i -e 's/"write_cd_rules"/"write_cd_rules mode"/' \
/etc/udev/rules.d/83-cdrom-symlinks.rules
```

Huomaa, että tässä vaiheessa ei ole tarpeellista luoda rules-tiedostoja tai symbolisia linkkejä, koska olet bind-mountannut isännän `/dev` -hakemiston LFS-järjestelmään ja oletetaan, että symboliset linkit ovat olemassa isännässä. Järjestelmän omat säännöt ja symboliset linkit luodaan, kun LFS-järjestelmä käynnistetään ensimmäisen kerran.

Jos sinulla on useita CD-ROM -laitteita, käynnistyksen yhteydessä luodut symboliset linkit saattavat osoittaa eri laitteisiin kuin mitä ne isäntäkoneessa osoittavat. Tämä johtuu siitä, että laitteita ei löydetä ennustettavassa järjestyksessä. Arvojen asetukset, jotka luodaan kun LFS-järjestelmä käynnistetään ensimmäisen kerran ovat stabiileja, joten tämä on ongelma ainoastaan siinä tapauksessa, että tarvittavat symbolisten linkkien osoittavan molemmissa järjestelmissä samoihin laitteisiin. Mikäli näin, niin tutki (ja mahdollisesti muokkaa) luotua `/etc/udev/rules.d/70-persistent-cd.rules` -tiedostoa käynnistyksen jälkeen varmistaaksesi, että annetut linkit täsmäävät tarpeisiisi.

3.4.3 Kahdennetut laitteet

Kuten osiossa [4.3 Laitteiden ja moduulien käsittely](#) selitettiin, järjestys, jossa laitteet `/dev` -hakemistoon lisätään on pohjimmiltaan täysin satunnainen. Jos sinulla on esimerkiksi USB-webbikamera ja TV-viritin, joskus `/dev/video0` viittaa kameraan ja `/dev/video1` viittaa virittimeen. Joskus taas uudelleen käynnistyksen jälkeen ne vaihtuvat päinvastoin. Kaikille laitteistotyypeille, lukuun ottamatta ääni- ja verkkokortteja, tämän voi korjata luomalla Udev-säännöt symbolisille linkeille. Verkkokortit käsitellään osiossa [4.5 Verkon konfigurointi](#), ja äänikorttien konfigurointi löytyy [BLFS](#)-kirjasta.

Jos sinulla on laitteita, joita tämä ongelma koskee (vaikka ongelmaa ei olisi nykyisessä Linux-jakelussasi), etsi laitetta vastaava hakemisto `/sys/class` tai `/sys/block` -hakemiston alta. Videolaitella hakemisto saattaa olla `/sys/class/video4linux/videoX`. Päätele ne attribuutit, jotka identifioivat laitteen (yleensä toimittajan ja tuotteen ID:t ja/tai sarjanumero toimii):

```
udevadm info -a -p /sys/class/video4linux/video0
```

Kirjoita säännöt, jotka luovat symboliset linkit. Esimerkiksi näin:

```
cat > /etc/udev/rules.d/83-duplicate_devs.rules << "EOF"

# Persistent symlinks for webcam and tuner
KERNEL=="video*", ATTRS{idProduct}=="1910", ATTRS{idVendor}=="0d81", \
    SYMLINK+="webcam"
KERNEL=="video*", ATTRS{device}=="0x036f", ATTRS{vendor}=="0x109e", \
    SYMLINK+="tvtuner"

EOF
```

Tuloksena on, että `/dev/video0` ja `/dev/video1` -laitteet silti viittaavat satunnaisesti virittimeen ja webbikameraan, mutta on olemassa symboliset linkit `/dev/tvtuner` ja `/dev/webcam`, jotka aina osoittavat oikeaan laitteeseen.

3.5 Verkon konfigurointi

Tämä osio pätee vain, jos verkkokortti pitää konfiguroida.

Jos verkkokorttia ei käytetä, ei todennäköisesti ole mitään tarvetta luoda yhtään konfiguraatiotiedostoa liittyen verkkokortteihin. Tässä tapauksessa verkkoasetuksiin liittyvät symboliset linkit poistetaan kaikista run-tason hakemistoista (`etc/rc.d/rc*.d`) bootscriptien asentamisen jälkeen ([osio 4.2](#)).

3.5.1 Verkkoliittymän konfiguraatiotiedostojen luominen

Se, mitkä liittymät network-skripti nostaa ylös tai poistaa käytöstä, riippuu usein tiedostoista `/etc/sysconfig/` -hakemistossa. Tämän hakemiston pitäisi sisältää tiedostot jokaiselle konfiguroitavalle liittymälle, kuten `ifconfig.xyz`, jossa "xyz":n tilalla pitää olla NIC:n nimi (esim. `eth0`). Tämän tiedoston sisällä on liittymän attribuutit, kuten sen IP-osoitteet ja aliverkkomaskit. Tiedostonimen rungon täytyy olla `ifconfig`.

Mikäli edellisen osion menettelytapaa ei käytetty, Udev asettaa verkkokortin nimet perustuen järjestelmän fyysisiin piirteisiin. Tällöin nimeksi voi tulla esimerkiksi `enp2s1`. Mikäli et ole varma, mikä verkkokorttisi nimi on, aja `ip link` sen jälkeen, kun olet käynnistänyt järjestelmäsi. On tärkeää, että `ifconfig.xyz` nimetään oikein verkkokortin nimen mukaan (esim. `ifconfig.enp2s1` tai `ifconfig.eth0`). Muuten verkkoliittymääsi ei alusteta käynnistuksen yhteydessä.

Seuraava komento luo esimerkkitiedoston `eth0`-laitteelle staattisella IP-osoitteella:

```
cd /etc/sysconfig/  
cat > ifconfig.eth0 << "EOF"  
ONBOOT=yes  
IFACE=eth0  
SERVICE=ipv4-static  
IP=192.168.1.2  
GATEWAY=192.168.1.1
```

```
PREFIX=24  
BROADCAST=192.168.1.255  
EOF
```

Muuttujien arvoiksi jokaisessa tiedostossa asetetaan omaan ympäristöön sopivat arvot. Lisää infoa löytyy `ifup`:n man-sivuilta.

Puretaan auki:

- Jos `ONBOOT`-muuttujalle on asetettu arvo `"yes"`, System V network -skripti ottaa NIC:n (Network Interface Card) käyttöön järjestelmän käynnistymisen yhteydessä. Jos muuttujalle asetetaan mikä tahansa muu arvo, network-skripti jättää NIC:n huomiotta ja sitä ei automaattisesti oteta käyttöön. Interfacen voi manuaalisesti käynnistää ja pysäyttää `ifup` ja `ifdown` -komentoilla.
- `IFACE`-muuttuja määrittää interfacen nimen, esimerkiksi `eth0`. Tieto vaaditaan kaikille verkkolaitteiden konfiguraatietiedostoille.
- `SERVICE`-muuttuja määrittää IP-osoitteen hankkimiseen käytettävän metodin. LFS-Bootscripits -paketilla on modulaarinen IP-osoitteiden jakelutapa. Lisätiedostojen luominen `/lib/services/` -hakemistoon mahdollistaa muut IP-osoitteiden jakelutavat. Tätä käytetään DHCP:hen (Dynamic Host Configuration Protocol), jota käsitellään [BLFS](#)-kirjassa.
- `GATEWAY`-muuttuja pitää sisällään oletusgatewayn IP-osoitteen, jos sellainen on olemassa. Mikäli ei ole, koko muuttuja kommentoidaan pois.
- `PREFIX`-muuttuja pitää sisällään verkkomaskissa käytettyjen bittien numerot. Jokainen IP-osoitteen oktetti on 8 bittiä. Jos aliverkon maski on 255.255.255.0, se käyttää ensimmäistä kolmea oktettia (24 bittiä) määritelmäkseen aliverkon numeron. Jos verkkomaski on 255.255.255.240, se käyttäisi 28 ensimmäistä bittiä. Yleensä 24 bittiä suurempia prefiksejä käyttävät DSL ja ISPs (Internet Service Providers). Tässä esimerkissä (`PREFIX=24`) verkkomaski on 255.255.255.0. Muuta `PREFIX`-muuttujaa aliverkkomaskiksi mukaisesti. Jos muuttujan arvoa ei määritellä, käytetään oletusarvoa 24.

3.5.2 /etc/resolv.conf -tiedoston luominen

Jos järjestelmän on tarkoitus olla yhteydessä Internetiin, se tarvitsee DNS:n (Domain Name Service) apua ratkaistakseen IP-osoitteiden domain-nimiä ja päinvastoin. Tämä saavutetaan parhaiten asettamalla DNS-palvelimen IP-osoite (saatavilla ISP:ltä tai verkon ylläpitäjältä) `/etc/resolv.conf` -tiedostoon. Luodaan tiedosto ajamalla seuraava:

```
cat > /etc/resolv.conf << "EOF"
# Begin /etc/resolv.conf

domain <Your Domain Name>
nameserver <IP address of your primary nameserver>
nameserver <IP address of your secondary nameserver>

# End /etc/resolv.conf
EOF
```

```
root:/sources# cat > /etc/resolv.conf << "EOF"
> # Begin /etc/resolv.conf
>
> domain localhost
> nameserver 8.8.8.8
> nameserver 8.8.4.4
>
> # End /etc/resolv.conf
> EOF
root:/sources#
```

domain-lausekkeen voi jättää pois tai korvata `search`-lausekkeella. Katso `resolv.conf`:n man-sivusta lisätietoja.

Korvaa `<IP address of the nameserver>` DNS-palvelimen IP-osoitteella. Jos tarvitset tai haluat ainoastaan yhden DNS-palvelimen, poista toinen `nameserver`-rivi tiedostosta. IP-osoite voi olla myös paikallisen verkon reititin.

HUOM! Googlen julkiset IPv4 DNS-palvelinten osoitteet ovat 8.8.8.8 ja 8.8.4.4.

3.5.3 Järjestelmän hostnimen konfigurointi

`/etc/hostname` -tiedostoa käytetään käynnistyksen aikana järjestelmän nimeämiseen.

Luodaan `/etc/hostname` -tiedosto ja asetetaan hostname:

```
echo "<lfs>" > /etc/hostname
```

```
root:/sources# echo "lfs" > /etc/hostname
root:/sources#
```

`<lfs>` korvataan tietokoneelle annetulla nimellä. Älä aseta FQDN:ää (Fully Qualified Domain Name) tähän. Se asetetaan `/etc/hosts` -tiedostoon.

3.5.4 /etc/hosts -tiedoston muokkaaminen

IP-osoite, FQDN (fully-qualified domain name) ja mahdolliset aliakset laitetaan `/etc/hosts` -tiedostoon. Syntaksi on:

```
IP_address myhost.example.org aliases
```

Ellei tietokoneen ole tarkoitus olla näkyvä Internetissä (rekisteröity domain ja julkinen IP-osoite), varmista, että IP-osoite on yksityisen verkon IP-osoitteiden alueella. Esimerkiksi seuraavat alueet ovat käytettävissä:

Private Network Address Range	Normal Prefix
10.0.0.1 - 10.255.255.254	8
172.x.0.1 - 172.x.255.254	16
192.168.y.1 - 192.168.y.254	24

x voi olla mikä tahansa numero välillä 16-31. y voi olla mikä tahansa numero välillä 0-255.

Pätevä yksityinen IP-osoite voi olla esimerkiksi 192.168.1.1. Pätevä FQDN tälle IP:lle voisi olla `lfs.example.org`.

Vaikka ei käyttäisi verkkokorttia, pätevä FQDN vaaditaan silti, jotta tietyt ohjelmat toimisivat oikein.

Luodaan `/etc/hosts` -tiedosto:

```
cat > /etc/hosts << "EOF"
# Begin /etc/hosts (network card version)

127.0.0.1 localhost
<192.168.1.1> <HOSTNAME.example.org> [alias1] [alias2 ...]

# End /etc/hosts (network card version)
EOF
```

<192.168.1.1> ja <HOSTNAME.example.org> arvot korvataan sinun ympäristöön sopivilla arvoilla. Vaihtoehtoiset aliasnimet voi jättää pois.

Jos verkkokorttia ei aiota konfiguroida, luodaan `/etc/hosts` -tiedosto seuraavasti:

```
cat > /etc/hosts << "EOF"
# Begin /etc/hosts (no network card version)

127.0.0.1 <HOSTNAME.example.org> <HOSTNAME> localhost

# End /etc/hosts (no network card version)
EOF
```

3.6 System V Bootscript - käyttö ja konfigurointi

3.6.1 Kuinka System V Bootskriptit toimivat?

Linux käyttää käynnistysjärjestelmää nimeltään SysVinit. SysVinit perustuu run-levels:in konseptiin. Konsepti voi hieman vaihdella riippuen järjestelmästä, joten ei voida olettaa, että koska hommat toimivat yhdellä Linux-distrolla, ne toimisivat samaan tapaan myös LFS:ssä. LFS:llä on oma tapansa tehdä asioita, mutta se noudattaa yleisesti hyväksyttyjä standardeja.

SysVinitiin viitataan tästä eteenpäin termillä "`init`". `init` toimii käyttämällä run-levels:ä. On olemassa seitsemän run-leveliä. Ne on numeroitu nollasta kuuteen (0-6). Itse asiassa run-levelsejä on enemmänkin, mutta ne on erityistarpeita varten, eikä niitä käytetä tavallisesti (kts. `init(8)`, jos haluat lisätietoja). Jokainen run-level vastaa toimintaa, jonka tietokoneen on tarkoitus tehdä, kun se käynnistyy. Oletus run-level on 3. Alla taulukko eri run-levelleistä:

Run level	Kuvaus
0	Halt
1	Single user mode
2	Multiuser, without networking
3	Full multiuser mode
4	<i>User defined mode</i>
5	Full multiuser mode with display manager
6	Reboot

3.6.2 Sysvinit'in konfiguroiminen

Kernelin alustamisen aikana ensimmäinen ohjelma, joka ajetaan, on joko määriteltä komentorivillä tai oletuksena `init`. Tämä ohjelma lukee alustustiedoston `/etc/inittab`. Luodaan tämä tiedosto:

```
cat > /etc/inittab << "EOF"
# Begin /etc/inittab

id:3:initdefault:

si::sysinit:/etc/rc.d/init.d/rc S

l0:0:wait:/etc/rc.d/init.d/rc 0
l1:S1:wait:/etc/rc.d/init.d/rc 1
l2:2:wait:/etc/rc.d/init.d/rc 2
l3:3:wait:/etc/rc.d/init.d/rc 3
l4:4:wait:/etc/rc.d/init.d/rc 4
l5:5:wait:/etc/rc.d/init.d/rc 5
l6:6:wait:/etc/rc.d/init.d/rc 6

ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now

su:S016:once:/sbin/sulogin

1:2345:respawn:/sbin/agetty --noclear tty1 9600
2:2345:respawn:/sbin/agetty tty2 9600
3:2345:respawn:/sbin/agetty tty3 9600
4:2345:respawn:/sbin/agetty tty4 9600
5:2345:respawn:/sbin/agetty tty5 9600
6:2345:respawn:/sbin/agetty tty6 9600

# End /etc/inittab
EOF
```

```

root:/sources# cat > /etc/inittab << "EOF"
> # Begin /etc/inittab
>
> id:3:initdefault:
>
> si::sysinit:/etc/rc.d/init.d/rc S
>
> l0:0:wait:/etc/rc.d/init.d/rc 0
> l1:S1:wait:/etc/rc.d/init.d/rc 1
> l2:2:wait:/etc/rc.d/init.d/rc 2
> l3:3:wait:/etc/rc.d/init.d/rc 3
> l4:4:wait:/etc/rc.d/init.d/rc 4
> l5:5:wait:/etc/rc.d/init.d/rc 5
> l6:6:wait:/etc/rc.d/init.d/rc 6
>
> ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
>
> su:S016:once:/sbin/sulogin
>
> 1:2345:respawn:/sbin/agetty --noclear tty1 9600
> 2:2345:respawn:/sbin/agetty tty2 9600
> 3:2345:respawn:/sbin/agetty tty3 9600
> 4:2345:respawn:/sbin/agetty tty4 9600
> 5:2345:respawn:/sbin/agetty tty5 9600
> 6:2345:respawn:/sbin/agetty tty6 9600
>
> # End /etc/inittab
> EOF
root:/sources#

```

Tämän alustustiedoston selitys on `inittab`:in man-sivuilla. LFS:ää ajatellen avainkomento on `rc`. Yllä oleva alustustiedosto ohjeistaa `rc`:n ajamaan kaikki skriptit, jotka alkavat S-kirjaimella `/etc/rc.d/rcS.d` -hakemistossa, jonka jälkeen ajetaan kaikki S-kirjaimella alkavat skriptit `/etc/rc.d/rc?.d` -hakemistossa, jossa `initdefault` arvo määrittää kysymymerkin.

`rc`-skripti suorittaa `/lib/lsb/init-functions` skriptin. Prosessin aikana luetaan myös konfiguraatiotiedosto `/etc/sysconfig/rc.site`, joka on valinnainen. Mitkä tahansa järjestelmän konfiguraatiotiedoston parametrit, jotka käydään seuraavissa osioissa läpi, voidaan määritellä tähän tiedostoon.

Debuggausta varten, `functions`-skripti loggaa kaiken tulosteen `/run/var/bootlog` -tiedostoon. Koska `/run` -hakemisto on `tmpfs`, hakemiston

sisältö säilytetään suorituksen aikana vain keskusmuistissa, mutta tieto lisätään kovalevylle kirjoitettavan tiedoston loppuun (`/var/log/boot.log`) käynnistymisprosessin lopussa.

3.6.2.1 Run Levels:ien vaihtaminen

Run-levelsien vaihtaminen tapahtuu komennolla `init <runlevel>`, jossa `<runlevel>` on haluttu run-level. Esimerkiksi tietokoneen uudelleen käynnistämiseksi käyttäjä voisi suorittaa `init 6` -komennon, joka on alias `reboot`-komennolle. Samaan tapaan `init 0` on alias `halt`-komennolle.

`/etc/rc.d` -hakemiston alla on useita hakemistoja, jotka näyttävät samalta kuin `rc?.d` (jossa `?` korvataan run-levelin numerolla) ja `rcsysinit.d`, ja kaikki sisältävät symbolisia linkkejä. Jotkut alkavat `K`-kirjaimella, jotkut `S`-kirjaimella, ja kaikissa on kaksi numeroa alkukirjaimen jälkeen. `K` tarkoittaa palvelun pysäyttämistä (kill) ja `S` tarkoittaa palvelun käynnistämistä (start). Numerot (00-99) määrittävät järjestyksen, jossa skriptit ajetaan. Mitä pienempi numero, sitä aiemmin skripti ajetaan. Kun `init` vaihtaa toiseen run-leveliin, oikeat palvelut joko käynnistetään tai pysäytetään, riippuen valitusta run-levelistä.

Varsinaiset skriptit ovat `/etc/rc.d/init.d` -hakemistossa. Hakemistossa ovat suoritettavat skriptit, ja kaikki symboliset linkit osoittavat niihin. `K`-linkit ja `S`-linkit osoittavat samaan skriptiin `/etc/rc.d/init.d` -hakemistossa. Tämä johtuu siitä, että skriptit voidaan kutsua eri parametrien kanssa, kuten `start`, `stop`, `restart`, `reload` ja `status`. Kun `K`-linkki kohdataan, oikea skripti ajetaan `stop`-argumentin kanssa. Kun `S`-linkki kohdataan, oikea skripti ajetaan `start`-argumentin kanssa.

On kuitenkin yksi poikkeus. Linkit, jotka alkavat `S`-kirjaimella `rc0.d` ja `rc6.d` -hakemistoissa eivät käynnistä mitään. Ne kutsutaan parametrilla `stop` pysäyttämään jotakin. Tässäkin on olevinaan logiikka, joka menee seuraavasti: kun käyttäjä aikoo uudelleen käynnistää tai sammuttaa järjestelmän, mitään ei tarvitse startata. Järjestelmä pitää vain sammuttaa.

Avataan vielä argumentit, jotka määrittävät mitä skriptit tekevät:

- `start` → Palvelu käynnistetään.
- `stop` → Palvelu pysäytetään.

- `restart` → Palvelu pysäytetään ja käynnistetään uudelleen.
- `reload` → Palvelun konfiguraatio päivitetään. Tätä käytetään, kun palvelun konfiguraatiotiedostoa on muokattu, mutta palvelua ei tarvitse käynnistää uudelleen.
- `status` → Kertoo onko palvelu päällä ja minkä PIDien kanssa.

Voit vapaasti muokata tapaa, jolla käynnistysprosessi toimii. Tässä annetut tiedot toimivat vain esimerkkinä.

3.6.3 Udev Bootscripts

`/etc/rc.d/init.d/udev -initskripti` käynnistää `udev`:n ja triggeröi "coldplug"-laitteet, jotka kernel on jo alustanut. Skripti myös poistaa `uevent` handlerin oletusarvon `/sbin/hotplug`. Tämä siksi, koska kernelin ei enää tarvitse kutsua ulkoista binääriä. Sen sijaan `udev` kuuntelee verkkosocketia kernelin aiheuttamien `uevents`:ien varalta.

`/etc/rc.d/init.d/udev_retry -initskripti` huolehtii tapahtumien uudelleen-kutsusta alijärjestelmiä varten, joiden säännöt saattavat nojata tiedostojärjestelmiin, joita ei ole liitetty käyttöjärjestelmään ennen kuin `mountfs`-skripti ajetaan (`/usr` ja `/var` ovat hyviä esimerkkejä tästä). `Udev_retry`-skripti ajetaan `mountfs`-skriptin jälkeen, joten tapahtumien (jos ne laukaistaan uudelleen) pitäisi nyt toisella kerralla onnistua. Konfigurointi tapahtuu `/etc/sysconfig/udev_retry` -tiedostossa; kaikki sanat tässä tiedostossa kommentteja lukuun ottamatta ovat alijärjestelmien nimiä, joita kutsutaan uudelleenyrityksen aikana. Löytääksesi alijärjestelmän laitteen, aja `udevadm info --attribute-walk <device>`, missä `<device>` on absoluuttinen polku `/dev` tai `/sys` -hakemistossa, kuten `/dev/sr0` tai `/sys/class/rtc`.

3.6.4 Järjestelmän kellon konfigurointi

`setclock`-skripti lukee ajan tietokoneen kellosta. Kello tunnetaan myös BIOSin kellona tai CMOS-kellona (Complementary Metal Oxide Semiconductor). Jos laitteiston aika on asetettu UTC-aikaan, skripti muuttaa laitteiston kellonajan paikalliseen aikaan käyttäen `/etc/localtime` -tiedostoa (joka kertoo `hwclock`-ohjelmalle millä aikavyöhykkeellä käyttäjä on). Koska ei ole mahdollista selvittää onko laitteiston kello asetettu UTC-aikaan, se tieto pitää konfiguroida manuaalisesti.

`Udev` ajaa `setclockin`, kun kernel käynnistyksen aikana havaitsee sen mahdolliseksi. Komennon voi ajaa myös manuaalisesti `stop`-parametrilla. Silloin järjestelmän aika tallennetaan CMOS-kelloon.

Jos et muista onko laitteiston kello asetettu UTC:hen, aja seuraava komento:

```
hwclock --localtime --show
```

```
root:/sources# hwclock --localtime --show
2017-04-28 13:52:08.958321+0300
root:/sources#
```

Se näyttää, mikä laitteiston kellon mukaan on tämän hetkinen aika. Jos aika täsmää siihen, mitä oma kellosi näyttää, tietokoneen kello on asetettu paikalliseen aikaan. Jos `hwclock`:n tuloste ei täsmää paikalliseen aikaan, on mahdollista, että se on asetettu UTC-aikaan. Varmista tämä lisäämällä tai vähentämällä aikavyöhykettä ajatellen oikea määrä tunteja `hwclock`:n näyttämään aikaan. Otetaan esimerkki: jos olet MST-aikavyöhykkeellä, joka on toiselta nimeltään GMT -0700, lisää seitsemän tuntia paikalliseen aikaan.

Vaihda UTC-muuttujan arvo alla nolaksi (0), jos laitteiston kelloa ei ole asetettu UTC-aikaan.

Luodaan uusi tiedosto, /etc/sysconfig/clock:

```
cat > /etc/sysconfig/clock << "EOF"
# Begin /etc/sysconfig/clock

UTC=1

# Set this to any options you might need to give to
hwclock,
# such as machine hardware clock type for Alphas.
CLOCKPARAMS=

# End /etc/sysconfig/clock
EOF
```

Seuraavasta linkistä löytyy lisätietoja LFS:n kellon säätämisestä: <http://www.linuxfromscratch.org/hints/downloads/files/time.txt>

HUOM! CLOCKPARAMS ja UTC -parametrit voi vaihtoehtoisesti asettaa /etc/sysconfig/rc.site -tiedostossa.

3.6.5 Linux-konsolin konfiguroiminen

Tässä osiossa käsitellään `console`-bootskriptin konfiguroimista. `console`-bootskripti asettaa näppäimistöasettelun, fontin ja kernel log tason. Jos non-ASCII -merkkejä ei käytetä (esim. copyright-merkki, brittiläinen punta ja euro-merkki) ja näppäimistö on mallia US, ison osan tästä osiosta voi ohittaa. Ilman konfiguraatio-tiedostoa (tai vastaavia asetuksia `rc.site` -tiedostossa) `console`-bootskripti ei tee mitään.

`console`-skripti lukee `/etc/sysconfig/console` -tiedostoa konfigurointia varten. Konfigurointia varten tulee määrittää, mitä näppäimistöasettelua ja fonttia halutaan käyttää. Apua valintaan löytyy esimerkiksi seuraavasta linkistä: <http://www.tldp.org/HOWTO/HOWTO-INDEX/other-lang.html>. Jos sopivaa määritelmää ei löytynyt, selaa `/usr/share/keymaps` ja `/usr/share/consolefonts` -hakemistoista sopivia näppäimistöasetuksia ja näyttöfontteja. `loadkeys(1)` ja `setfont(8)` -ohjesivuilta löytyy lisää ohjeita konfigurointiin.

`/etc/sysconfig/console` -tiedosto koostuu konfiguraatioasetuksista, jokainen omalle rivilleen ja seuraavanlaisessa muodossa: `VARIABLE="value"`. Alla muutamia esimerkkejä muuttujista ja niiden mahdollisista arvoista:

- `LOGLEVEL` → Tämä muuttuja määrittelee tason, millä ydin lähettää tietoa konsoliin. Mahdolliset tasot vaihtelevat ykkösestä ("1" = ei viestejä) kahdeksaan ("8"). Oletustaso on "7".
- `KEYMAP` → Tämä muuttuja määrittelee argumentit `loadkeys`-ohjelmalle. Tyypillisesti arvo on ladattavan näppäimistöasettelun nimi, esim. "it". Jos tätä muuttujaa ei ole asetettu, bootskript ei suorita `loadkeys`-ohjelmaa, ja asetteluna käytetään kernelin oletusnäppäimistöä. Huomaa, että joillakin näppäimistöasetuksilla on useita saman nimisiä versioita (cz ja sen variaatiot `qwerty/` ja `qwertz/` -hakemistoissa, es ja sen variaatiot `olpc/` ja `qwerty/`, jne.). Näissä tapauksissa isäntähakemistoon tulisi olla oikeassa (esim. `qwerty/es`) muodossa, että oikea asettelu ladataan.

- `KEYMAP_CORRECTIONS` → Tämä (harvoin käytetty) muuttuja määrittelee argumentit `loadkeys`-ohjelman toiselle kutsulle. Tämä on tarpeen, jos näppäinasetteluun oletukset eivät täysin sovellu ja asetuksia täytyy hienosäätää. Esimerkiksi, jos haluaa sisällyttää euro-merkin näppäimistöasetteluun, jossa sitä ei tavallisesti ole, pitää tälle muuttujalle antaa arvo `"euro2"`.
- `FONT` → Tämä muuttuja määrittelee argumentit `setfont`-ohjelmalle. Tyyppillisesti tämä pitää sisällään fonttinimen, `"-m"`, ja ladattavan sovellusmerkistökartan nimen. Otetaan esimerkki: ladataksesi `"lat1-16"` -fontin yhdessä `"8859-1"` -sovellusmerkistökartan kanssa (soveltuva USA:ssa), aseta tämän muuttujan arvoksi `"lat1-16 -m 8859-1"`. UTF-8 -tilassa kernel käyttää sovellusmerkistökarttaa koottujen 8-bittisten näppäinkoodien muuttamiseen näppäimistökartassa UTF-8:aan, ja siten `"-m"` -parametrin argumentti pitäisi asettaa koottujen näppäinkoodien asettamiseen näppäimistökartassa.
- `UNICODE` → Aseta tälle muuttujalle arvo `"1"`, `"yes"` tai `"true"` asettaaksesi konsoli UTF-8 -tilaan. Tämä on hyödyllistä UTF-8:aan pohjautuvissa lokaa-leissa, mutta muistta tapauksissa haitallinen asetus.
- `LEGACY_CHARSET` → Monille näppäimistöpohjille ei ole olemassa Unicode-näppäimistökarttaa Kbd-paketissa. `console-bootscrip`t muuttaa saatavilla olevan näppäimistökartan UTF-8:aan lennosta, jos tämä muuttuja on asetettu saatavilla olevan non-UTF-8 -näppäimistökartan koodittamiseen.

Joissakin näppäinkartoissa on kuolleita näppäimiä, toisin sanoen näppäimiä, jotka eivät itsessään tuota yhtään merkkiä, mutta lisäävät aksentin seuraavan näppäimen tuottamalle merkillle. Jotkut näppäinkartat saattavat myös määrittää yhdistelmäsääntöjä, kuten: `"paina Ctrl+A+E saadaksesi Æ"`. Linux-4.9.9 tulkaa kuolleet näppäimet ja yhdistelmäsäännöt näppäinkartassa oikein ainoastaan silloin, kun yhteen kootuista lähdemerkeistä ei kasva yli 8-bittisiä merkkejä. Tämä vajavaisuus

ei vaikuta Euroopan näppäinkarttoihin, koska niissä aksentit lisätään ASCII-merkeihin, joissa ei ole aksentteja, tai kaksi ASCII-merkkiä kootaan yhdessä. UTF-8 -tilassa se on kuitenkin ongelma, esimerkiksi kreikan kielessä, missä joskus tarvitsee laittaa aksentti kirjaimelle "alpha". Ratkaisu on joko välttää UTF-8:n käyttöä tai asentaa X window system, jolla ei ole tätä rajoitusta.

Joillekin kielille, kuten kiinan, japanin ja korean kieli, Linux-konsolia ei voida konfiguroida näyttämään tarvittavia merkkejä. Käyttäjien, jotka tarvitsevat tällaisia kieliiä, pitäisi asentaa X Window System, fontit, jotka kattavat tarpeelliset merkit ja sopiva input-metodi (esim. SCIM, joka tukee monia kieliiä).

HUOM! `/etc/sysconfig/console` -tiedosto kontrolloi ainoastaan Linuxin tekstikonsolin lokalisointia. Sillä ei ole mitään tekemistä sopivan näppäimistökartan tai terminaalifonttien asettamisessa X Window -järjestelmässä, ssh-istunnoissa tai sarjakonsoleissa. Näihin eivät ylempänä kaksi viimeisintä listassa mainittua asiaa päde.

Luodaan suomalaiselle näppäimistökartalle sopiva tiedosto:

```
cat > /etc/sysconfig/console << "EOF"
# Begin /etc/sysconfig/console

KEYMAP="fi"
FONT="lat1-16 -m 8859-1"

# End /etc/sysconfig/console
EOF
```

```
root:/sources# cat > /etc/sysconfig/console << "EOF"
> # Begin /etc/sysconfig/console
>
> KEYMAP="fi"
> FONT="lat1-16 -m 8859-1"
>
> # End /etc/sysconfig/console
> EOF
root:/sources#
root:/sources#
```

3.6.6 Tiedostojen luominen käynnistyksen yhteydessä

Joskus halutaan luoda tiedostoja käynnistyksen yhteydessä. Esimerkkinä voisi toimia vaikka `/tmp/.ICE-unix` -hakemisto. Tämä tehdään lisäämällä asetus `/etc/sysconfig/createfiles` -konfiguraatioskriptiin. Createfiles tiedoston muoto ja asetukset löytyvät kyseisestä tiedostosta kommentteina.

3.6.7 syslogd-skriptin konfiguroiminen

`sysklogd`-skripti turvautuu `syslogd`-ohjelmaan osana System V alustusta. `-m 0` -asetus poistaa jaksottaisen aikaleiman käytöstä, jonka `syslogd` oletuksena kirjoittaa lokitiedostoihin 20 minuutin välein. Jos haluat asettaa tämän toiminnon, niin editoi `/etc/sysconfig/rc.site` -tiedostoa ja määritä muuttujalle `SYSKLOGD_PARMS` haluttu arvo. Esimerkiksi poistaaksesi kaikki parametrit, aseta muuttujalle tyhjä arvo:

```
SYSKLOGD_PARMS=
```

Tsekkaa man `syslogd` muita asetuksia varten.

3.6.8 rc.site -tiedosto

Valinnainen `/etc/sysconfig/rc.site` -tiedosto pitää sisällään asetukset, jotka automaattisesti asetetaan jokaiselle SystemV -bootskriptille. Tämä on vaihtoehtoinen tapa asettaa arvot, jotka määritellään `hostname`, `console` ja `clock` -tiedostoissa `/etc/sysconfig/` -hakemistossa. Jos muuttujia on määritelty kumpaankin paikkaan, niin silloin skriptikohtaiset arvot astuvat voimaan.

`rc.site` sisältää myös parametrit, jotka voivat kustomoida käynnistysprosessin muita osa-alueita. `IPROMPT` -muuttujan asettaminen esimerkiksi mahdollistaa bootskriptien valikoivan ajamisen. Muut vaihtoehdot on kuvailtu tiedoston kommentteissa. Tiedoston oletusversio on seuraavanlainen:

```
# rc.site
# Optional parameters for boot scripts.

# Distro Information
# These values, if specified here, override the defaults
#DISTRO="Linux From Scratch" # The distro name
#DISTRO_CONTACT="lfs-dev@linuxfromscratch.org" # Bug report
address
#DISTRO_MINI="LFS" # Short name used in filenames for dis-
tro config
# Define custom colors used in messages printed to the
screen

# Please consult `man console_codes` for more information
# under the "ECMA-48 Set Graphics Rendition" section
#
# Warning: when switching from a 8bit to a 9bit font,
# the linux console will reinterpret the bold (1;) to
# the top 256 glyphs of the 9bit font. This does
# not affect framebuffer consoles
```

```
# These values, if specified here, override the defaults
#BRACKET="\033[1;34m" # Blue
#FAILURE="\033[1;31m" # Red
#INFO="\033[1;36m"     # Cyan
#NORMAL="\033[0;39m"   # Grey
#SUCCESS="\033[1;32m"  # Green
#WARNING="\033[1;33m"  # Yellow

# Use a colored prefix
# These values, if specified here, override the defaults
#BMPREFIX=""
#SUCCESS_PREFIX="${SUCCESS} * ${NORMAL}"
#FAILURE_PREFIX="${FAILURE}*****${NORMAL}"
#WARNING_PREFIX="${WARNING} *** ${NORMAL}"

# Manually set the right edge of message output (characters)
# Useful when resetting console font during boot to override
# automatic screen width detection
#COLUMNS=120

# Interactive startup
#IPROMPT="yes" # Whether to display the interactive boot-prompt
#itime="3"     # The amount of time (in seconds) to display the prompt

# The total length of the distro welcome string, without escape codes
#wlen=$(echo "Welcome to ${DISTRO}" | wc -c )
#welcome_message="Welcome to ${INFO}${DISTRO}${NORMAL}"
```

```
# The total length of the interactive string, without es-
cape codes
#ilen=$(echo "Press 'I' to enter interactive startup" | wc
-c )
#i_message="Press '${FAILURE}I${NORMAL}' to enter interac-
tive startup"

# Set scripts to skip the file system check on reboot
#FASTBOOT=yes

# Skip reading from the console
#HEADLESS=yes

# Write out fsck progress if yes
#VERBOSE_FSCK=no

# Speed up boot without waiting for settle in udev
#OMIT_UDEV_SETTLE=y

# Speed up boot without waiting for settle in udev_retry
#OMIT_UDEV_RETRY_SETTLE=yes

# Skip cleaning /tmp if yes
#SKIPTMPCLEAN=no

# For setclock
#UTC=1
#CLOCKPARAMS=

# For consolelog (Note that the default, 7=debug, is noisy)
#LOGLEVEL=7

# For network
#HOSTNAME=mylfs
```

```
# Delay between TERM and KILL signals at shutdown
#KILLDELAY=3

# Optional syslogd parameters
#SYSLOGD_PARMS="-m 0"

# Console parameters
#UNICODE=1
#KEYMAP="de-latin1"
#KEYMAP_CORRECTIONS="euro2"
#FONT="lat0-16 -m 8859-15"
#LEGACY_CHARSET=
```

3.6.8.1 Boot ja Shutdown -skriptien muokkaaminen

LFS-bootskriptit käynnistävät ja sammuttavat järjestelmän kohtuullisen tehokkaasti, mutta `rc.site` -tiedostossa on muutamia asetuksia, joilla näitä voi nopeuttaa. Lisäksi tiedostossa voidaan muokata käyttöjärjestelmän viestintään liittyviä asetuksia. Kyseinen tiedosto löytyy siis `/etc/sysconfig/rc.site` ja siihen voi määritellä esimerkiksi alla listattuja asetuksia:

- udev-bootskriptissä suoritetaan `udev settle`, jonka suorittaminen voi kestää, riippuen järjestelmässä olevista laitteista. Jos ympäristössä on vain yksinkertaisia osiointeja ja yksi verkkokortti, käynnistysprosessin tuskin kannattaa suorittaa tätä komentoa. Ohittaaksesi komennon, aseta muuttuja `OMIT_UDEV_SETTLE=y`.
- Myös `udev_retry` -bootskripti suorittaa `udev settle`:n oletuksena. Tämä komento tarvitaan vain, jos `/var` -hakemisto liitetään (mount) erikseen. Tämä siksi, että kello tarvitsee tiedoston `/var/lib/hwclock/adjtime`. Joissain muissakin tapauksissa järjestelmä voi joutua odottamaan udevin suorittamista, mutta monissa asennuksissa näin ei ole. Komennon ohittamisen vaikutuksia voi kokeilla asettamalla muuttuja `OMIT_UDEV_RETRY_SETTLE=y`.
- Oletuksena tiedostojärjestelmän tarkistus ei tulosta suorituksesta näyttöön mitään infoa ja se näkyy viiveenä käynnistysprosessin aikana. Saadaksesi `fsck` tulosteet näkyviksi, aseta muuttuja `VERBOSE_FSCK=y`.
- Uudelleen käynnistuksen yhteydessä saatat haluta ohittaa tiedostojärjestelmän tarkastuksen (`fsck`) kokonaan. Tehdäksesi tämän, luo tiedosto `/fastboot` tai käynnistä järjestelmä komennolla `/sbin/shutdown -f -r now`. Voit myös pakottaa kaikkien tiedostojärjestelmien tarkastuksen luomalla `/forcefsck` tiedoston tai ajamalla `shutdown`-komennon parametrilla `-F` parametrin `-f` sijaan.
- Asettamalla muuttujan `FASTBOOT=y`, poistetaan `fsck`:n suoritus käynnistysprosessin yhteydessä. Tätä ei suositella pysyväksi asetukseksi.

- Tavallisesti kaikki tiedostot `/tmp` -hakemistossa poistetaan käynnistyksen aikana. Riippuen hakemistossa olevien tiedostojen tai hakemistojen määrästä, tämä voi aiheuttaa huomattavan viiveen käynnistysprosessiin. Ohittaaksesi näiden tiedostojen poistamisen, aseta muuttuja `SKIPT-MPCLEAN=y`.
- Sammutuksen aikana `init`-prosessi lähettää `TERM`-signaalin jokaiselle prosessille, jonka se on käynnistänyt (esim. `agetty`). Tämän jälkeen `init`-prosessi odottaa asetetun ajan (oletuksena 3 sekuntia), että aliprosessi lopettaa suorituksensa. Jos aliprosessin suoritus ei pääty, `init`-prosessi lähettää niille prosesseille `KILL`-signaalin ja odottaa jälleen. Tätä toistetaan `sendsignals`-skriptissä kaikille niille käynnissä oleville prosesseille, joita niiden oma toteutus ei sammuta. Viive `init`:ille voidaan asettaa parametrilla. Esimerkiksi viiveen poistamiseksi kokonaan voidaan asettaa paramteriarvo `-t0` kun konetta sammutetaan tai käynnistetään uudelleen (esim. `/sbin/shutdown -t0 -r now`). `sendsignals`-skriptin viiveen voi poistaa asettamalla parametrin `KILLDELAY=0`.

3.7 Bash Shell Startup -tiedostot

Shell-ohjelma `/bin/bash` (tästä eteenpäin “shell”) käyttää startup-tiedostojen koelmaa luodakseen suoritussympäristön. Jokaisella tiedostolla on oma käyttötarkoitus ja ne vaikuttavat kirjautumiseen ja interaktiivisiin ympäristöihin eri tavoin. Tiedostot `/etc` -hakemistossa tarjoavat yleiset, kaikkia käyttäjiä koskevat asetukset. Jos vastaava asetus on olemassa kotihakemistossa sijaitsevassa asetustiedostossa, se ylikirjoittaa `/etc`-hakemiston tiedostoissa määritellyt asetukset.

Vuorovaikutteinen shell-istunto käynnistetään, jos käyttäjä onnistuu kirjautumaan koneelle. Kirjautumisen tarkistaa `/bin/login`-komento, joka lukee `/etc/passwd` -tiedostoa. Kirjautunut käyttäjä voi käynnistää uusia istuntoja shell-näkymästä (non-login shell) komentoriviltä (esim. `$ /bin/bash`). Ei-interaktiivista shelliä käytetään kun halutaan suorittaa skriptiä. Tällöin shell suorittaa pelkästään skriptiä eikä käyttäjältä odoteta syötettä komentojen suorittamiseksi.

`/etc/profile` ja `~/.bash_profile` -tiedostot luetaan, kun shell käynnistetään vuorovaikutteisena eli interaktiivisena login shellinä.

`/etc/profile` asettaa muun muassa joitakin natiiville kielituelle tarpeellisia ympäristömuuttujia. Kielitukiasetukset vaikuttavat seuraavasti:

- Ohjelmien tuloste käännetään käyttäjän valitsemalle kielelle
- Merkkien oikea luokittaminen kirjaimiksi, numeroiksi ja muihin luokkiin. Tämä on tarpeellista, jotta bash hyväksyy non-ASCII -merkit komentorivillä ei-englanninkielisissä lokaaleissa.
- Maalle ominainen aakkosjärjestys
- Sopiva oletuspaperikoko
- Rahan, ajan ja päivämäärään oikea muotoilu

Listan kaikista Glibc:n tukemista lokaaleista saa ajamalla seuraavan komennon:

```
locale -a
```

Charmapseilla voi olla aliaksia, esim. "ISO-8859-1" :een viitataan usein myös nimillä "iso8859-1" ja "iso88591". Jotkut sovellukset eivät käsittele erilaisia synonyymeja oikein (esim. ne vaativat, että "UTF-8" on kirjoitettu "UTF-8" eikä "utf8"), joten pääsääntöisesti on turvallisinta valita valmiiksi määritelty nimi tiettyä lokaalia varten. Määrittääksesi nimen oikeassa muodossa, aja seuraava komento, missä `<locale name>` on `locale -a` -komennon antama tuloste haluamaasi lokaalia varten ("en_GB.iso8859" esimerkissä).

```
LC_ALL=<locale name> locale charmap
```

"fi_FI.iso88591" -lokaalia varten yllä oleva komento tulostaa seuraavaa:

ISO-8859-1

Tämä johtaa lopulliseen "fi_FI.ISO-8859-1" -lokaalin asettamiseen. On tärkeää, että yllä olevalla heuristiikalla löydetty lokaali testataan ennen kuin se lisätään Bashin startup-tiedostoihin:

```
LC_ALL=<locale name> locale language
```

```
root:/sources# LC_ALL="fi_FI.iso88591" locale language
Finnish
```

```
LC_ALL=<locale name> locale charmap
```

```
root:/sources# LC_ALL="fi_FI.iso88591" locale charmap
ISO-8859-1
```

```
LC_ALL=<locale name> locale int_curr_symbol
```

```
root:/sources# LC_ALL="fi_FI.iso88591" locale int_curr_symbol
EUR
```

```
LC_ALL=<locale name> locale int_prefix
```

```
root:/sources# LC_ALL="fi_FI.iso88591" locale int_prefix  
358
```

Yllä olevien komentojen pitäisi tulostaa kielen nimi, lokaalin käyttämä character encoding, paikallinen valuutta ja maan puhelinten suuntanumero. Jos yksikin yllä olevista epäonnistuu ja antaa samanlaisen viestin kuin alla, se tarkoittaa, että lokaalisi ei ole joko asentunut kappaleessa 3 tai Glibc ei tue sitä.

```
locale: Cannot set LC_* to default locale: No such file or  
directory
```

Jos näin tapahtuu, sinun pitäisi asentaa joko haluttu lokaali käyttäen `localedef` -komentoa tai harkita toista lokaalia.

Joistakin LFS:n ulkopuolisista paketeista saattaa myös puuttua tuki valitsemallesi lokaalille. Yksi esimerkki on X-kirjasto (osa X Window Systemiä), joka tulostaa seuraavan virheviestin, jos lokaali ei tarkalleen täsmää yhtäkään sen sisäisissä tiedostoissa olevista character mapeista:

```
Warning: locale not supported by Xlib, locale set to C
```

Useissa tapauksissa Xlib olettaa, että character map listataan isoilla kirjaimilla ja sääntöjen mukaisilla viivoilla. Esimerkiksi "ISO-8859-1" mieluummin kuin "iso88591". On myös mahdollista löytää sopiva spesifikaatio poistamalla lokaalin spesifikaation charmap-osa. Tämän voi tsekata ajamalla `locale charmap` -komento molemmissa lokaaleissa. Esimerkiksi "de_DE.ISO-8859-15€euro" pitäisi vaihtaa muotoon "de_DE€euro", jotta Xlib tunnistaisi tämän lokaalin.

Muut paketit saattavat myös toimia väärin (eivät silti välttämättä näytä virheviestejä), jos lokaalin nimi ei täsmää niiden odotuksia. Näissä tapauksissa tutkimalla, miten muut Linux-distrot tukevat lokaaliasi saattaa löytää jotain hyödyllistä informaatiota.

Kun oikeat lokaalit asetukset on määritetty, luodaan `/etc/profile` -tiedosto:

```
cat > /etc/profile << "EOF"
# Begin /etc/profile

export LANG=<ll>_<CC>.<charmap><@modifiers>

# End /etc/profile
EOF
```

```
root:/sources# cat > /etc/profile << "EOF"
> # Begin /etc/profile
>
> export LANG=fi_FI.iso88591
>
> # End /etc/profile
> EOF
root:/sources#
```

Jotkin ohjelmat eivät tue UTF-8 -pohjaisia lokaaleja kunnolla. Tällä hetkellä ongelmia dokumentoidaan ja korjataan. Lisätietoa esimerkiksi seuraavasta linkistä: <http://www.linuxfromscratch.org/blfs/view/8.0/introduction/locale-issues.html>

3.8 /etc/inputrc -tiedoston luominen

`inputrc`-tiedosto on Readline-kirjaston konfiguraatiotiedosto, joka lisää toiminnallisuutta käyttäjän syötteisiin komentoriviltä. Se toimii muuttamalla näppäimistön syötteet tietyiksi toiminnoiksi. Bash käyttää Readlinea, kuten myös useimmat muut shellit ja monet ohjelmat.

Yleensä käyttäjäkohtaista toiminnallisuutta ei tarvita, joten alla oleva komento luo globaalin `/etc/inputrc` -tiedoston, jota käyttävät kaikki, jotka kirjautuvat sisään. Käyttäjäkohtaisia muutoksia asetuksiin voi tehdä luomalla `.inputrc` -tiedosto käyttäjän kotihakemistoon ja lisätä oma muokkaukset siihen.

Lisää tietoa siitä, miten editoida `inputrc`-tiedostoa löytyy `info bash` komennolla *Readline Init File* -osion alta. `info readline` on myös hyvä tietolähde.

Alla on yleinen `inputrc`-tiedosto, johon on myös kommentoitu asetuksen tarkoitus. Huomaa, että kommentit eivät voi olla samalla rivillä komentojen kanssa. Luodaan tiedosto seuraavaalla komennolla:

```
cat > /etc/inputrc << "EOF"
# Begin /etc/inputrc
# Modified by Chris Lynn roryo@roryo.dynup.net

# Allow the command prompt to wrap to the next line
set horizontal-scroll-mode Off

# Enable 8bit input
set meta-flag On
set input-meta On

# Turns off 8th bit stripping
set convert-meta Off

# Keep the 8th bit for display
set output-meta On
```

```
# none, visible or audible
set bell-style none

# All of the following map the escape sequence of the value
# contained in the 1st argument to the readline specific
functions
"\eOd": backward-word
"\eOc": forward-word

# for linux console
"\e[1~": beginning-of-line
"\e[4~": end-of-line
"\e[5~": beginning-of-history
"\e[6~": end-of-history
"\e[3~": delete-char
"\e[2~": quoted-insert

# for xterm
"\eOH": beginning-of-line
"\eOF": end-of-line

# for Konsole
"\e[H": beginning-of-line
"\e[F": end-of-line

# End /etc/inputrc
EOF
```

```
root:/sources# cat > /etc/inputrc << "EOF"
> # Begin /etc/inputrc
> # Modified by Chris Lynn <roryo@roryo.dynup.net>
>
> # Allow the command prompt to wrap to the next line
> set horizontal-scroll-mode Off
>
> # Enable 8bit input
```

3.9 /etc/shells -tiedoston luominen

`shells`-tiedosto sisältää listan järjestelmän login shelleistä. Sovellukset käyttävät tätä tiedostoa määrittääkseen, onko joku shell saatavilla. Jokaiselle shellille pitäisi olla oma rivi, joka sisältää shellin absoluuttisen polun.

Esimerkiksi `chsh` shell lukee tästä tiedostosta voiko oikeudeton (unprivileged) käyttäjä vaihtaa oman kirjautumisensa login shellin. Jos komentonimeä ei ole listattu, muutosta ei sallita.

Jotkut ohjelmat kuten GDM tai FTP-taustaprosessi (jotka perinteisesti estävät pääsyn käyttäjiltä, joiden shellejä ei ole listattu tässä tiedostossa) vaativat määrittämisen `shells`-tiedostoon.

```
cat > /etc/shells << "EOF"
# Begin /etc/shells

/bin/sh
/bin/bash

# End /etc/shells
EOF
```

```
root:/sources# cat > /etc/shells << "EOF"
> # Begin /etc/shells
>
> /bin/sh
> /bin/bash
>
> # End /etc/shells
> EOF
root:/sources#
```

4 LFS-JÄRJESTELMÄN TEKEMINEN BOOTATTAVAKSI

Seuraavaksi tehdään LFS-järjestelmästä itsenäisesti käynnistyvä käyttöjärjestelmä. Tässä kappaleessa luodaan `fstab`-tiedosto, käännetään [kernel](#) uutta LFS-järjestelmää varten ja asennetaan GRUB-käynnistyslatain, jotta LFS-järjestelmä voidaan valita käynnistysvalikosta.

4.1 /etc/fstab -tiedoston luominen

Jotkin ohjelmat käyttävät /etc/fstab -tiedostoa määrittääkseen, mihin tiedostojärjestelmien pitää oletuksena olla liitetty, missä järjestyksessä ja mitkä niistä pitää tarkistaa eheysongelmien varalta ennen liittämistä. Luodaan uusi tiedostojärjestelmien taulu:

```
cat > /etc/fstab << "EOF"
# Begin /etc/fstab

# file system  mount-point  type      options                                dump  fsck
#                                           order

/dev/<xxx>      /                <fff>     defaults                            1     1
/dev/<yyy>      swap            swap      pri=1                              0     0
proc           /proc           proc      nosuid,noexec,nodev               0     0
sysfs          /sys            sysfs     nosuid,noexec,nodev               0     0
devpts         /dev/pts        devpts    gid=5,mode=620                     0     0
tmpfs          /run            tmpfs     defaults                           0     0
devtmpfs       /dev            devtmpfs  mode=0755,nosuid                   0     0

# End /etc/fstab
EOF
```

```
root:/sources# cat > /etc/fstab << "EOF"
> # Begin /etc/fstab
>
> # file system  mount-point  type      options                                dump  fsck
> #                                           order
>
> /dev/sdb1      /                ext4      defaults                            1     1
> /dev/sdb2      /boot            ext4      defaults                            1     2
> /dev/sdb3      /home            ext4      defaults                            1     2
> /dev/sdb5      /usr             ext4      defaults                            1     2
> /dev/sdb6      /opt             ext4      defaults                            1     2
> /dev/sdb7      /usr/src         ext4      defaults                            1     2
> /dev/sda2      swap            swap      pri=1                              0     0
> proc           /proc           proc      nosuid,noexec,nodev               0     0
> sysfs          /sys            sysfs     nosuid,noexec,nodev               0     0
> devpts         /dev/pts        devpts    gid=5,mode=620                     0     0
> tmpfs          /run            tmpfs     defaults                           0     0
> devtmpfs       /dev            devtmpfs  mode=0755,nosuid                   0     0
>
> # End /etc/fstab
> EOF
root:/sources#
```

Korvaa `<xxx>`, `<yyy>` ja `<fff>` järjestelmälle sopivilla arvoilla (esimerkiksi `sda2`, `sda5` ja `ext4`). Lisätietoa tiedoston eri kentistä löytyy `man 5 fstab` komennolla.

MS-DOS tai Windows -perustaiset tiedostojärjestelmät (ts. `vfat`, `ntfs`, `smbfs`, `cifs`, `iso9660`, `udf`) tarvitsevat `"iocharset"` -mount-asetuksen, jotta ei-ASCII-merkit tiedostonimissä tulkitaan oikein. Tämän asetuksen arvon asetetaan samaksi kuin paikallinen merkistö muutamien muokkauksin: Kulloisenkin merkistöasettelun määritelmä (löytyy File Systems -> Native Language Support) on käännetty suoraan kerneliin tai moduuliksi. `vfat` ja `smbfs` -tiedostojärjestelmille tarvitaan myös `codepage`-asetus. Se tulisi asettaa maakohtaiseen MS-DOS:ssa käytettävään `codepage`-arvoon. Jos esimerkiksi liitetään USB-levyä `ru_RU.KOI8-R` lokalisoinnilla, silloin `/etc/fstab` -tiedostoon mount riville tulisi lisätä seuraava asetus:

```
noauto,user,quiet,showexec,iocharset=koi8r,codepage=866
```

Vastaavat asetukset `ru_RU.UTF-8` -lokalisaatiolle ovat:

```
noauto,user,quiet,showexec,iocharset=utf8,codepage=866
```

HUOM! Jälkimmäisessä tapauksessa kernel lähettää seuraavan viestin:

```
FAT: utf8 is not a recommended IO charset for FAT filesystems,  
      filesystem will be case sensitive!
```

Varoituksen voi jättää huomiotta, koska `"iocharset"`-asetuksen kaikki muut arvot johtavat tiedostonimien näyttämiseen UTF-8 -lokaaleissa väärin.

Codepage ja `iocharset` muuttujille on mahdollista asettaa oletusarvot koskemaan joitakin tiedostojärjestelmiä kernelin konfiguraation aikana. Asetuksiin liittyvät parametrit ovat nimeltään `"Default NLS Option"` (`CONFIG_NLS_DEFAULT`), `"Default Remote NLS Option"` (`CONFIG_SMB_NLS_DEFAULT`), `"Default codepage for FAT"` (`CONFIG_FAT_DEFAULT_CODEPAGE`) ja `"Default iocharset for FAT"` (`CONFIG_FAT_DEFAULT_IOCHARSET`). NTFS-tiedostojärjestelmiä ei tueta.

`barrier=1` -mount-asetuksella on mahdollista lisätä ext3-tiedostojärjestelmän vi-
kasietoisuutta, jos käytössä oleva kovalevy tukee ominaisuutta. Asetus määritel-
lään `/etc/fstab` -tiedostossa. Tarkistaaksesi tukeeko levyasema tätä asetusta,
aja [hdparm](#) levyasemalle. Esim. jos alla oleva komento palauttaa tyhjän tulosteen,
`barrier`-asetusta ei tueta:

```
hdparm -I /dev/sda | grep NCQ
```

HUOM! Myöskään LVM (Logical Volume Management) -pohjaiset partitiot eivät
voi käyttää `barrier`-asetusta.

4.2 Linux-4.9.9

Arvioitu käännösaika: 4.4 - 66.0 SBU (tyypillisesti n. 6 SBU)

Vaadittava levytila: 960 - 4250 MB (tyypillisesti n. 1100 MB)

Asennettavat tiedostot: *config-4.9.9*, *vmlinuz-4.9.9-lfs-8.0* ja *System.map-4.9.9*

Asennettavat hakemistot: */lib/modules*, */usr/share/doc/linux-4.9.9*

4.2.1 Kernelin asentaminen

Viimeinen rakennettava osio on kernel. Se on LFS:n monimutkaisin paketti konfiguroitavaksi. Kernelin rakentamiseen kuuluu seuraavat vaiheet: konfiguroiminen, kääntäminen ja asentaminen. Tarvittaessa `README`-tiedostosta löytyy vaihtoehtoisia konfiguroimistapoja.

Ensin puretaan paketti ja siirrytään purettuun hakemistoon:

```
tar xf linux-4.9.9.tar.xz
cd linux-4.9.9
```

```
root:/sources# tar xf linux-4.9.9.tar.xz
root:/sources# cd linux-4.9.9
root:/sources/linux-4.9.9#
```

Valmistellaan paketti kääntämistä varten:

```
make mrproper
```

```
root:/sources/linux-4.9.9# make mrproper
root:/sources/linux-4.9.9#
```

Tämä varmistaa, että kernelin lähdekoodi on varmasti puhdas mahdollisista aikaisemmista asetuksista. Kernelin kehittäjät suosittelevat, että komento ajetaan ennen jokaista kernelin kääntämistä.

Konfiguroi kernel valikkopohjaisen käyttöliittymän kautta. Yleistä infoa kernelin konfiguraatiosta: <http://www.linuxfromscratch.org/hints/downloads/files/kernel-configuration.txt>

BLFS:ssä on infoa tiettyihin kernelin konfiguraatiovaatimuksiin liittyen LFS:n ulkopuolisiin paketteihin: <http://www.linuxfromscratch.org/blfs/view/8.0/longindex.html#kernel-config-index>

Lisää infoa kernelin konfiguroimisesta ja rakentamisesta saa myös täältä: <http://www.kroah.com/lkn/>

HUOM! Hyvä tapa aloittaa kernelin konfiguroiminen on ajaa `make defconfig`. Tämä luo oletuskonfiguraation, joka ottaa huomioon nykyisen järjestelmäarkkitehtuurin.

Huolehdi seuraavien asetusten konfiguroinnista. Muuten järjestelmä ei välttämättä toimi oikein tai käynnisty ollenkaan:

```
Device Drivers --->
  Generic Driver Options --->
    [ ] Support for uevent helper [CONFIG_UEVENT_HELPER]
    [*] Maintain a devtmpfs filesystem to mount at /dev
    [CONFIG_DEVTMPFS]
```

Myös muihin asetuksiin kannattaa tutustua, riippuen toki järjestelmän vaatimuksista ja halutuista ominaisuuksista. BLFS-pakettien vaatimista asetuksista voit lukea [täältä](#).

HUOM! Jos isäntälaitteistosi käyttää UEFia, yllä mainitun `make defconfig` -komennon pitäisi automaattisesti lisätä joitakin EFI:in liittyviä kernel-asetuksia.

Salliaksesi LFS-kernelin käynnistymisen isäntäkoneen UEFI boot-ympäristöstä, kernelissäsi pitää olla tämä asetetus valittuna:

```
Processor type and features --->
  [*] EFI stub support [CONFIG_EFI_STUB]
```

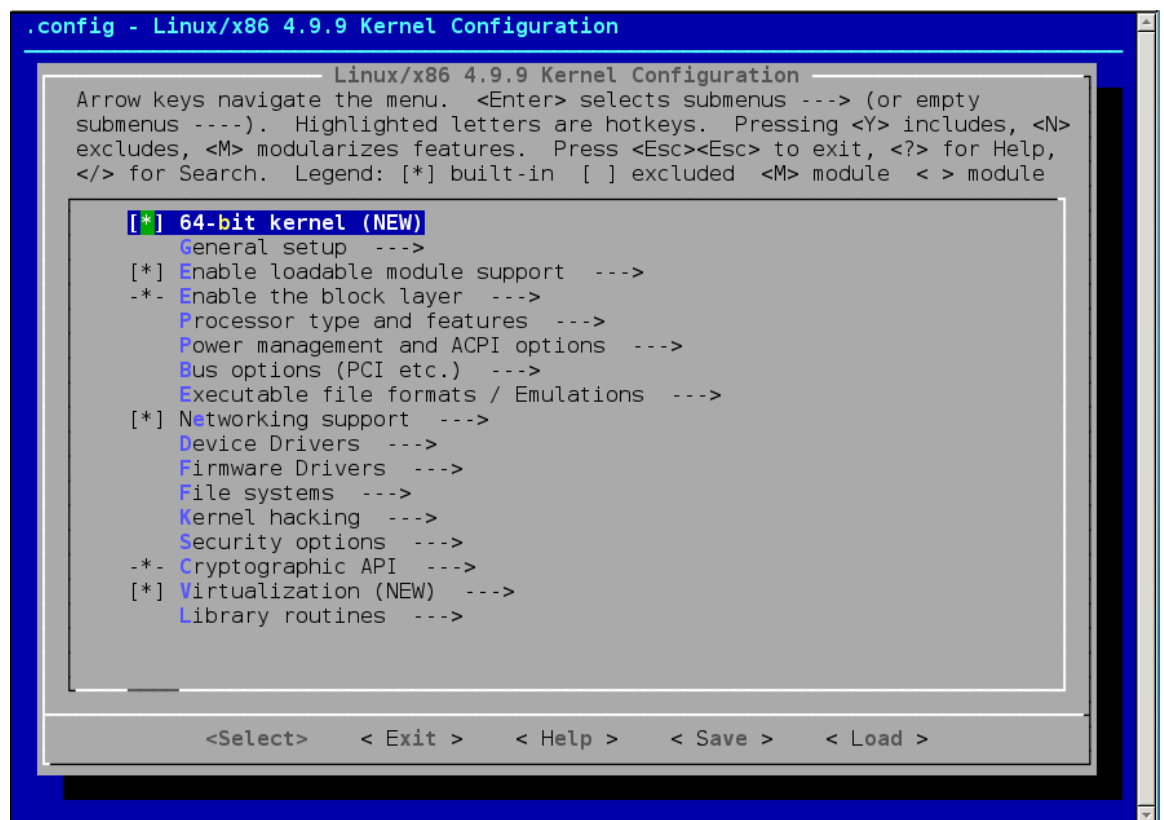
Täydellisempi selitys UEFI-ympäristöjen hallinnasta LFS-järjestelmään liittyen löytyy täältä: <http://www.linuxfromscratch.org/hints/downloads/files/lfs-uefi.txt>

Puretaan valintoja hiukan auki:

`Support for uevent helper` → Asetuksen valitseminen saattaa sekaantua laitehallinnan kanssa, kun käytetään Udevia/Eudevä.

`Maintain a devtmpfs` → Tämä luo automaattiset laitetiedostot, jotka kernel ottaa käyttöön, jopa ilman Udev:a. Udev:a suoritetaan näiden jälkeen ja sen tehtävä on hallita käyttöoikeuksia ja symbolisia linkkejä. Tämä asetus pitää olla valittuna, jos käytetään Udev / Eudev:a.

```
make menuconfig
```



Puretaan auki valinnaisten make-ympäristömuuttujien tarkoitukset:

- `LANG=<host_LANG_value> LC_ALL` → Asetetaan host:n oletuslokalisaatio.

- Jos arvo asetetaan, niin silloin pitää muistaa korvata `<host_LANG_value>` hostin `$LANG` -muuttujan arvolla. Vaihtoehtoisesti voidaan käyttää myös hostin `$LC_ALL` tai `$LC_TYPE` -muuttujien arvoja.
- Vaihtoehtoisesti, `make oldconfig` saattaa olla sopivampi monissa tilanteissa. Lue `README`-tiedostosta lisätietoja.
- Halutessasi voit ohittaa kernelin konfiguroinnin kopioimalla kernel config -tiedoston, `.config`, isäntäjärjestelmästä (olettaen, että se on saatavilla) paketoimattomaan `linux-4.9.9` -hakemistoon. Tätä vaihtoehtoa ei kuitenkaan suositella. On parempi käydä läpi koko konfiguraatiovalikko ja luoda kernelin konfiguraatio alusta asti.

Käännetään kernel ja moduulit:

```
make
```

Jos käytetään kernelin moduuleja, moduulien konfigurointi `/etc/modprobe.d` -hakemistossa voi olla tarpeen. Infoa koskien moduuleja ja kernelin konfiguraatiota on osiossa [3.3 Laitteiden ja moduulien käsittely](#) ja kernelin dokumentaatiossa `linux-4.9.9/Documentation` -hakemistossa. Voit tsekata myös `modprobe.d(5)`.

Asennetaan moduulit, jos kernelin konfiguraatio määrittelee niitä käytettävän:

```
make modules_install
```

Kun kernelin kääntäminen on valmis, muutamia tiedostoja tulee kopioida `/boot` -hakemistoon.

HUOM! Jos isäntäjärjestelmässä on erillinen `/boot` -partitio, alla kopioitavat tiedostot pitää siirtää sinne. Helpoin tapa tehdä tämä on yhdistää `/boot` hostilla `/mnt/lfs/boot` -hakemistoon. Suorita seuraava komento isäntäjärjestelmässä pääkäyttäjänä:

```
mount --bind /boot /mnt/lfs/boot
```

```
root@dc-jumalan_linukka:/usr/share# mount --bind /boot /mnt/lfs/boot
```

Polku kernel imageen saattaa vaihdella riippuen käytettävästä alustasta. Tiedostonimi alla voidaan muuttaa maun mukaan, mutta tiedostonimen rungon pitäisi olla `vmlinuz`, jotta se on yhteensopiva seuraavassa osiossa kuvailtavan käynnistysprosessin automaattisen setupin kanssa. Seuraava komento olettaa, että käytössä on x86-arkkitehtuuri:

```
cp -v arch/x86/boot/bzImage /boot/vmlinuz-4.9.9-lfs-8.0
```

```
root:/sources/linux-4.9.9# cp -v arch/x86/boot/bzImage /boot/vmlinuz-4.9.9-lfs-8.0
'arch/x86/boot/bzImage' -> '/boot/vmlinuz-4.9.9-lfs-8.0'
root:/sources/linux-4.9.9#
```

`System.map` on symbolitiedosto kerneliä varten. Se kartoittaa jokaisen funktion aloituskohdat kernel API:ssa ja kernelin datarakenteiden osoitteet suoritettavaa kerneliä varten. Sitä käytetään lähteenä, kun tutkitaan kernelin ongelmia. Kopioidaan `map`-tiedosto `boot` hakemistoon:

```
cp -v System.map /boot/System.map-4.9.9
```

```
root:/sources/linux-4.9.9# cp -v System.map /boot/System.map-4.9.9
'System.map' -> '/boot/System.map-4.9.9'
root:/sources/linux-4.9.9#
```

`make menuconfig`-komennon luoma kernelin konfiguraatiotiedosto `.config` sisältää kaikki asetukset juuri käännetyille kernelille. On hyödyllistä pitää tämä tiedosto tallessa tulevaa tarvetta varten:

```
cp -v .config /boot/config-4.9.9
```

```
root:/sources/linux-4.9.9# cp -v .config /boot/config-4.9.9
'.config' -> '/boot/config-4.9.9'
root:/sources/linux-4.9.9#
```

Asennetaan Linux-kernelin dokumentaatio:

```
install -d /usr/share/doc/linux-4.9.9
```

```
root:/sources/linux-4.9.9# install -d /usr/share/doc/linux-4.9.9
root:/sources/linux-4.9.9#
```

```
cp -r Documentation/* /usr/share/doc/linux-4.9.9
```

```
root:/sources/linux-4.9.9# cp -r Documentation/* /usr/share/doc/linux-4.9.9
root:/sources/linux-4.9.9#
```

On tärkeää huomata, että tiedostoja kernelin lähdekoodihakemistossa ei omista root. Kun paketti puretaan pääkäyttäjänä (kuten teimme chrootin sisällä), tiedostoilla ovat ne käyttäjä- ja ryhmä-ID:t, jotka olivat paketoijan tietokoneella. Tämä ei yleensä ole ongelma, koska lähdekoodit poistetaan asennuksen jälkeen. Jos lähdekoodia säilytetään pidempään, niin on hyvä huomioida mahdollisuus, että paketoijan käyttämä käyttäjä-ID määritellään paikallisesti koneelle luodulle käyttäjälle. Tässä tapauksessa tunnuksen käyttäjä saisi kirjoitusoikeuden kernelin lähdekoodiin.

HUOM! Monissa tapauksissa kernelin konfiguraatio pitää päivittää myöhemmin [BLFS](#):ssä asennettavia paketteja varten. Toisin kuin muiden pakettien kohdalla, kernelin lähdekoodia ei tarvitse välttämättä poistaa sen jälkeen, kun uusi kernel on asennettu.

Mikäli kernelin lähdekoodi halutaan säilyttää, suorita `chown -R 0:0 linux-4.9.9` -hakemistossa varmistaaksesi, että root omistaa kaikki tiedostot.

```
root:/sources/linux-4.9.9# chown -R 0:0 /sources/linux-4.9.9
root:/sources/linux-4.9.9#
```

VAROITUS. Joku kernel-dokumentaatio suosittelee luomaan symbolisen linkin `/usr/src/linux` -hakemistosta osoittaen kernelin source-hakemistoon. Tämä on ominaista kerneleille ennen 2.6-sarjaa. Tätä symbolista linkkiä ei missään tapauksessa saa luoda LFS-järjestelmässä, koska se aiheuttaa ongelmia niiden pakettien kohdalla, jotka mahdollisesti haluat rakentaa valmiissa LFS-järjestelmässä.

VAROITUS. Järjestelmän include-hakemistossa (/usr/include) sijaitsevat headerit pitäisi aina vastata niitä, jotka Glibc on kääntänyt, eli toisin sanoen ne 'puhdistetut' headerit, jotka asennettiin osiossa [3.6.1 Linux-4.9.9 API Headers](#). Sen vuoksi niitä ei ikinä pitäisi korvata raw kernel headereilla tai muillakaan kernelin headereilla.

4.2.2 Linuxin moduulien lataamisjärjestyksen konfiguroiminen

Linuxin moduulit ladataan pääsääntöisesti automaattisessa järjestyksessä, mutta joskus siihen pitää puuttua. Komento, joka lataa moduulit (`modprobe` tai `insmod`) käyttää `/etc/modprobe.d/usb.conf` -tiedostoa järjestyksen muuttamista varten. Tiedosto luodaan, että esimerkiksi moduuleina rakennetut USB-ajurit (`ehci_hcd`, `ohci_hcd` ja `uhci_hcd`) ladataan oikeassa järjestyksessä; `ehci_hcd` pitää ladata ennen `ohci_hcd`:ta ja `uhci_hcd`:ta, jotta vältetään turhilta varoituksilta käynnistymisen aikana.

Luodaan uusi tiedosto `/etc/modprobe.d/usb.conf` ajamalla seuraava:

```
install -v -m755 -d /etc/modprobe.d
```

```
root:/sources/linux-4.9.9# install -v -m755 -d /etc/modprobe.d
install: creating directory '/etc/modprobe.d'
root:/sources/linux-4.9.9#
```

```
cat > /etc/modprobe.d/usb.conf << "EOF"
# Begin /etc/modprobe.d/usb.conf

install ohci_hcd /sbin/modprobe ehci_hcd ; /sbin/modprobe -
i ohci_hcd ; true
install uhci_hcd /sbin/modprobe ehci_hcd ; /sbin/modprobe -
i uhci_hcd ; true

# End /etc/modprobe.d/usb.conf
EOF
```

```
root:/sources/linux-4.9.9# cat > /etc/modprobe.d/usb.conf << "EOF"
> # Begin /etc/modprobe.d/usb.conf
>
> install ohci_hcd /sbin/modprobe ehci_hcd ; /sbin/modprobe -i ohci_hcd ; true
> install uhci_hcd /sbin/modprobe ehci_hcd ; /sbin/modprobe -i uhci_hcd ; true
>
> # End /etc/modprobe.d/usb.conf
> EOF
root:/sources/linux-4.9.9#
```

4.3 GRUBin käyttäminen käynnistysprosessin aloittamiseksi

VAROITUS. GRUBin epäonnistunut konfiguroiminen voi aiheuttaa sen, että koko järjestelmä on toimintakyvytön ilman vaihtoehtoista boot-laitetta, kuten CD-ROMia. Tätä osiota ei vaadita LFS-järjestelmän boottamiseen. Halutessasi voit kuitenkin tehdä tässä osiossa esitettyjä muutoksia boot-loaderiisi (esim. Grub-Legacy, GRUB2 tai LILO).

Varmista, että sinulla on erillinen käynnistyslevy koneen käynnistämistä varten, jos järjestelmän käynnistysosioista tulee käyttökelvoton. Mikäli sinulla ei vielä ole käynnistysmediaa, voit luoda sellaisen. Jotta alla oleva toimisi, sinun täytyy hypätä [BLFS](#):ään ja asentaa `xorriso` [libisoburn](#)-paketista. Huomaa, että libisoburn vaatii, että libburn-1.4.6 ja libisofs-1.4.6 on asennettuina ensin. Jos et tarvitse käynnistyslevyä, tämän osan voi ohittaa.

```
cd /tmp
grub-mkrescue --output=grub-img.iso
xorriso -as cdrecord -v dev=/dev/cdrw blank=as_needed grub-
img.iso
```

```
Writing to '/dev/cdrw' completed successfully.
xorriso : NOTE : Re-assessing -outdev '/dev/cdrw' ('/dev/sr0')
xorriso : NOTE : Disc status unsuitable for writing
Drive current: -outdev '/dev/cdrw'
Media current: DVD+R/DL
Media status : is written , is closed
Media summary: 1 session, 5200 data blocks, 10.2m data,      0 free
root@dc-jumalan_linukka:/tmp#
```

Ennen viimeistä komentoa aseta poltettava DVD-levy asemaan.

HUOM! Käynnistääksesi LFS-järjestelmän koneessa, jossa on UEFI enabloituna, kerneliin täytyy olla valittuna edellisessä osiossa kuvailtu `CONFIG_EFI_STUB` -asetus. LFS voidaan kuitenkin käynnistää myös ilman sitä GRUB2:n avulla. Tehdäksesi niin, UEFI Mode ja Secure Boot -asetukset pitää poistaa isäntäjärjestelmän BIOS-asetuksista. Lisätietoja: <http://www.linuxfromscratch.org/hints/downloads/files/lfs-uefi.txt>

4.3.1 GRUBin nimeämiskäytännöt

GRUB käyttää omaa nimeämisrakennetta asemille ja partitioille: (hdn, m) , jossa n on kovalevyn numero ja m on partition numero. Kovalevyn numerot alkavat nollasta, mutta partition numerot alkavat yhdestä normaaleille partitioille ja viidessä extended-partitioille. Tämä eroaa aiemmista versioista, joissa molemmat numerot alkoivat nollasta. Esimerkiksi partitio `sda1` on $(hd0, 1)$ ja `sdb3` on $(hd1, 3)$. GRUB ei pidä CD-ROM -asemia kovalevyinä.

4.3.2 Konfigurointi

GRUB asentuu kovalevyn ensimmäiselle fyysiselle raidalle. Tämä alue ei ole osa mitään tiedostojärjestelmää. Sieltä GRUB osoittaa käyttöjärjestelmän käynnistymiseksi vaadittaviin GRUB-moduuleihin. Oletussijainti on `/boot/grub/` joiden avulla käyttöjärjestelmän lataaminen jatkuu.

Käynnistysosion ja sen tiedstojen sijainnista päättää käyttäjä. Yksi suositus on, että luodaan erillinen boot-osio (100 MB) vain käynnistykseen liittyvälle informaatiolle. Siten jokainen linux käyttöjärjestelmä huolimatta siitä onko se LFS tai jokin kaupallinen distro, voivat päästä käsiksi samoihin boot-tiedostoihin. Tämän kirjan alussa boot-hakemistosta tehtiin oma osionsa. Huolehdi, että siirrät kaikki tiedostot nykyisestä `/bootdirectory` (esim. linux-kernel, jonka rakensit edellisessä osiossa) uuteen partitioon. Huolehdi myös, että osio liitetään `/boot :ina`. Tieto päivitetään myös `/etc/fstab` tiedostoon.

Seuraavassa esimerkissä oletetaan, että root (tai erillinen boot) -partitio on `sda2`.

Asenna GRUB-tiedostot `/boot/grub` -hakemistoon ja kirjoita kovalevyn käynnistysraidalle (MBR) uusi GRUB käynnistyslatain:

VAROITUS. Seuraava komento kirjoittaa nykyisen boot loaderin päälle. Älä aja komentoa, ellei tämä ole toivottua. Esimerkiksi, jos käytät kolmannen osapuolen boot-manageria hallinnoimaan Master Boot Recordia (MBR).

```
grub-install /dev/sda
```

4.3.3 GRUB-konfiguraatiotiedoston luominen

Luo `/boot/grub/grub.cfg`:

```
cat > /boot/grub/grub.cfg << "EOF"
# Begin /boot/grub/grub.cfg
set default=0
set timeout=5

insmod ext2
set root=(hd0,2)

menuentry "GNU/Linux, Linux 4.9.9-lfs-8.0" {
    linux    /boot/vmlinuz-4.9.9-lfs-8.0 root=/dev/sda2
    ro
}
EOF
```

GRUBin näkökulmasta kernel-tiedostot liittyvät käytettyyn partitioon. Jos käytit erilistä `/boot`-hakemistoa, poista `/boot` yllä olevasta `linux`-rivistä. Sinun pitää myös muuttaa `set root` -rivi osoittamaan boot-partitioon.

GRUB on erittäin tehokas ohjelma ja se tarjoaa valtavan määrän asetuksia eri laitteista, käyttöjärjestelmistä ja partitiotyypeistä käynnistämistä varten. Asetuksia on useita, kuten [splash screenejä](#) ja äänten soittamista. Näitä asetuksia ei kuitenkaan käydä läpi tässä oppaassa vaan jätetään lukijan oman mielenkiinnon varaan.

VAROITUS. On olemassa komento, `grub-mkconfig`, joka kirjoittaa konfiguraatiotiedoston automaattisesti. Se käyttää skriptejä `/etc/grub.d/` -hakemistosta ja tuhoaa kaikki tekemäsi muutokset. Nämä skriptit on suunniteltu ensisijaisesti non-source distroille ja niitä ei suositella käytettäväksi LFS:ssä. Jos asennat kaupallisen Linux-distron, on todennäköistä, että tämä ohjelma suoritetaan asennuksen aikana. Varmista, että olet varmuuskopioinut oman `grub.cfg` -tiedostosi.

5 LOPPU

Onneksi olkoon! Uusi LFS-järjestelmä on asennettu. Voi olla hyvä idea luoda `/etc/lfs-release` -tiedosto. Sen avulla on helppo tarkastaa, mikä LFS-versio on asennettu järjestelmään. Luo tiedosto ajamalla seuraava:

```
echo 8.0 > /etc/lfs-release
```

```
root:/# echo 8.0 > /etc/lfs-release
root:/#
```

On myös hyvä idea luoda tiedosto, joka näyttää uuden järjestelmäsi statuksen ver-raten [Linux Standard Baseen \(LSB\)](#):

```
cat > /etc/lsb-release << "EOF"
DISTRIB_ID="Linux From Scratch"
DISTRIB_RELEASE="8.0"
DISTRIB_CODENAME="<your name here>"
DISTRIB_DESCRIPTION="Linux From Scratch"
EOF
```

Laita jotain `'DISTRIB_CODENAME'` -kenttään tehdäksesi järjestelmästä uniikin.

5.1 Ilmoittaudu LFS-käyttäjäksi

Nyt, kun olet kahlannut kirjan loppuun ja asentanut järjestelmäsi onnistuneesti, voit halutessasi rekisteröityä LFS-käyttäjäksi. Sinun tarvitsee antaa nimesi ja ensimmäinen käyttämäsi LFS-versio:

<http://www.linuxfromscratch.org/cgi-bin/lfscounter.php>

5.2 Järjestelmän uudelleen käynnistäminen

Nyt, kun kaikki ohjelmat on asennettu, on aika käynnistää tietokone uudelleen. Huomaa kuitenkin, että tämän kirjan mukaan luotu järjestelmä on hyvin minimaalinen, eikä todennäköisesti sisällä toimintoja, jotka tarvitsisit jatkaaksesi eteenpäin. Muutaman lisäpaketin asentaminen BLFS-kirjasta, kun olet yhä chroot-ympäristössä voi olla hyvä idea. Tämä siksi, että elämä olisi helpompaa, kun uudelleen käynnistät tietokoneesi uuteen LFS:ääsi. Tässä muutamia ehdotuksia:

- Tekstipohjainen WWW-selain, kuten [Lynx](#), mahdollistaa BLFS-kirjan lukemisen yhdessä virtuaalisessa terminaalissa samaan aikaan, kun rakennat paketteja toisessa.
- [GPM](#)-paketti mahdollistaa copy/paste -toiminnot virtuaalisissa terminaleissa.
- Jos staattinen IP-konfigurointi ei vastaa verkotuksesi vaatimuksia, [dhcpcd](#)-paketti voi olla hyödyllinen.
- [sudo](#) voi olla hyödyllinen.
- Jos haluat päästä käsiksi uuteen järjestelmääsi etänä GUI-ympäristöstä, asenna [openssh](#) ja sen edeltäjä, [openssl](#).
- Asenna [wget](#), jotta saat helposti haettuja tiedostoja netistä.
- Jos vähintään yhdessä levyasemistasi on GUID Partition Table (GPT), joko [gptfdisk](#) tai [parted](#) on hyödyllinen.
- Seuraavien konfiguraatiotiedostojen läpikäyminen on soveliaista tässä kohdassa:
 - `/etc/bashrc`
 - `/etc/dircolors`
 - `/etc/fstab`
 - `/etc/hosts`

- o /etc/inputrc
- o /etc/profile
- o /etc/resolv.conf
- o /etc/nanorc (tai vimrc, jos asensit vim:in)
- o /root/.bash_profile
- o /root/.bashrc
- o /etc/sysconfig/network
- o /etc/sysconfig/ifconfig.eth0

Nyt, kun asianmukaiset huomautukset on saatu pois alta, siirrytään eteenpäin. Seuraavaksi käynnistetään uusi LFS ensimmäistä kertaa. Ensimmäiseksi poistutaan chroot-ympäristöstä:

```
exit
```

Irrotetaan virtuaaliset tiedostojärjestelmät:

```
umount -v $LFS/dev/pts
umount -v $LFS/dev
umount -v $LFS/run
umount -v $LFS/proc
umount -v $LFS/sys
```

Jos useita partitioita luotiin, muut partitiot pitää irrottaa ennen pääpartition irrottamista:

```
umount -v $LFS/usr
umount -v $LFS/home
umount -v $LFS
```

Jos muita partitioita ei ole, irrota vain pääpartitio:

```
umount -v $LFS
```

Uudelleenkäynnistetään järjestelmä:

```
shutdown -r now
```

Olettaen, että GRUB-[käynnistyslatain](#) asennettiin, kuten aiemmin hahmoteltiin, menu on asetettu käynnistämään LFS 8.0 automaattisesti.

Kun uudelleenkäynnistys on valmis, LFS-järjestelmä on valmis käyttöä varten ja lisää ohjelmia voidaan asentaa tarpeen mukaan.

5.3 Mitä seuraavaksi?

Toivottavasti tämä kirja osoittautui hyödylliseksi ja olet oppinut paljon uusia asioita. Takki saattaa tuntua tyhjältä tällaisen rupeaman jälkeen. Seuraavassa on listattu asioita auttamaan eteenpäin pääsemisessä:

- Ylläpito: Bugeja ja tietoturvaraportteja raportoidaan säännöllisesti kaikille ohjelmille. Koska LFS-järjestelmä on käännetty sourcesta, on käyttäjästä kiinni pitäytyä ajan tasalla raporteista. On olemassa monia online-resursseja näiden raporttien seuraamiseen:
 - CERT (Computer Emergency Response Team) = CERT:illä on postituslista, joka julkaisee tietoturvaraportteja koskien monia käyttöjärjestelmiä ja ohjelmia. <http://www.us-cert.gov/cas/signup.html>
 - Bugtraq = Bugtraq on myös postituslista. Se julkaisee uusia havaittuja tietoturvaongelmia ja toisinaan myös mahdollisia korjauksia niihin. <http://www.securityfocus.com/archive>
- BLFS: Projekti, joka jatkaa siitä mihin LFS jää. Se auttaa käyttäjää kehittämään järjestelmää hänen tarpeidensa mukaan tarjoamalla ohjeita monien pakettien asennukseen ja konfigurointiin LFS-järjestelmän päälle. <http://www.linuxfromscratch.org/blfs/>
- LFS Hints = Kokoelma LFS-yhteisön vapaaehtoisten toimittamia opetusellisia dokumentteja. <http://www.linuxfromscratch.org/hints/list.html>
- Postituslistat: On olemassa useita LFS-postituslistoja, joihin voit liittyä, jos tarvitset apua, haluat pysyä ajan tasalla tai olla muuten osallisena projektissa. <http://www.linuxfromscratch.org/mail.html>
- The Linux Documentation Project (TLPD): Sisältää paljon HOWTO-dokumentteja, oppaita ja man-sivuja. <http://www.tldp.org/>

6 SANASTOA

ASSEMBLER

Assembly-kielen kääntäjä.

BASH

UNIX [shell](#) ja komentokieli.

BLFS (Beyond Linux From Scratch)

Projekti, joka jatkaa siitä mihin LFS jää. Se auttaa käyttäjää kehittämään järjestelmää hänen tarpeidensa mukaan tarjoamalla ohjeita monien pakettien asennukseen ja konfigurointiin LFS-järjestelmän päälle.

BOOT LOADER (Käynnistyslatain)

Ohjelma, joka aloittaa käyttöjärjestelmän latauksen tietokoneen käynnistyksen jälkeen. Sanotaan myös vyöryttäjäksi.

BUILD

Tarkoittaa pakettien kääntämistä. Kääntämällä itse paketti on mahdollista muokata monia erilaisia asioita asennusvaiheessa. Omaa järjestelmää luodessa rakennetaan kaikki paketit itse.

CHROOT

chroot:lla voi käynnistää ohjelmia siten, että ne luulevat jonkun nykyisen tiedostojärjestelmän alihakemistoa juurihakemistokseen.

CROSS COMPILER (Ristikääntäjä)

Kääntäjä, joka kykenee luomaan suoritettavaa koodia eri alustalle, millä itse pyörii. Esimerkiksi kääntäjä, joka pyörii Windows 10 PC:llä, mutta luo koodia, joka pyörii Android-puhelimessa, on ristikääntäjä.

DISTRIBUTION, DISTRO

Jakelu; puhuttaessa Linux distroista tarkoitetaan esimerkiksi Ubuntua, Mintiä, Fedoraa jne. <https://distrowatch.com/>

DYNAAMINEN LINKKERI

Ajonaikainen linkittäjä; `ld.so`; lataa ohjelman tarvitsemat jaetut kirjastot, valmistelee ohjelman ajettavaksi ja sitten ajaa sen.

EOF

End Of File; Loppumerkki; kertoo ohjelmalle, että tiedoston loppu on saavutettu, eikä lisää tietoa ole saatavilla.

EXT4

Edistyneempi tiedostojärjestelmäversio ext3:sta. Se tukee parempaa skaalautuvuutta ja luotettavuutta isoille tietojärjestelmille (64 bit).

EXTENDED PARTITION

Levy voidaan jakaa vain neljään [primary partitiioon](#) eli osioon, mutta on kuitenkin mahdollista tehdä useita osioita jakamalla yksi (tai useampi) neljästä [primary partitiosta](#) aliosioiksi. Primary osio, jota jaetaan aliosioiksi, kutsutaan nimellä *extended partition*.

FILESYSTEM HIERARCHY STANDARD (FHS)

FHS-standardi määrittelee joukon hakemistoja ja niiden pääasiallisen käyttötarkoituksen. FHS:n mukaisessa tiedostojärjestelmässä kaikki tiedostot ja hakemistot on tallennettu juurihakemiston (/) alle, vaikka ne olisivat erillisillä fyysisillä laitteilla.

HASH

Näyttää kaikkien ajettujen komentojen polun. Hash-pöytä tyhjenee, kun hash resetoidaan komennolla `hash -r`. Hash vaikuttaa tapaan, jolla shell-ympäristö muistaa löydettyjen työkalujen sijainnit.

INODE

Tiedoston metatiedot sisältävä tietorakenne Unixien tiedostojärjestelmissä. i-node sisältää tiedot tiedoston tyypistä (tiedosto, hakemisto, viittaus eli linkki, nimetty putki, laitetiedosto, ...), oikeuksista, omistajasta, ryhmästä ja linkkien määrästä sekä tiedoston koosta ja linkit tiedostolle varattuihin blokkeihin. Sen sijaan i-node ei sisällä tietoa tiedoston nimestä tai nimistä.

KERNEL

Ydin; Ohjelma, joka muodostaa tietokoneen käyttöjärjestelmän keskeisimmän ytimen. Sillä on täysi kontrolli kaikkeen, mitä tapahtuu järjestelmässä. Kernel ei itsessään ole suoraan vuorovaikutuksessa käyttäjään, vaan [shelliin](#) ja muihin ohjelmiin, sekä järjestelmän laitteistoon. Kernel tarjoaa tietokoneen osat ohjelmien käyttöön sekä huolehtii käyttöoikeuksista ja resurssien jaosta prosessien ja eri käyttäjien kesken.

KIRJASTO

Tietotekniikasta puhuttaessa kirjastolla tarkoitetaan ohjelmien tarvitsemia toimintoja. Linuxissa ajettavat ohjelmat tarvitsevat käytännössä aina alleen vähintäänkin libc-kirjaston. Ohjelmaa käynnistettäessä ajonaikainen linkittäjä lataa ja valmistelee sen tarvitsemat jaetut kirjastot ennen varsinaisen ohjelmätiedoston käynnistämistä.

KOVAKOODATTU

Kirjoitettu lähdekoodiin, eikä ole syötettävissä tai muualta haettavissa.

KOVA LINKKI (Hard link)

Tiedostojärjestelmän hakemistossa sijaitseva viittaus fyysisellä tallennusvälineellä sijaitsevaan tiedostoon. Kova linkki osoittaa tiedostoon [inode](#)-numerolla. Tiedoston, johon halutaan osoittaa, on oltava siis olemassa ja sijaittava samassa tiedostojärjestelmässä, jonne linkkiä yritetään tehdä. Vaikka tiedoston alkuperäinen nimi muutettaisiin, kova linkki osoittaa silti samaan tiedostoon.

KÄYNNISTYSLATAIN (Boot loader)

Ohjelma, joka aloittaa käyttöjärjestelmän latauksen tietokoneen käynnistyksen jälkeen.

KÄYTTÄJÄTILA

User space; muistialue, jossa suurin osa ohjelmista ja osa ajureista toimii. Vastakohta on kernel space, joka on varattu käyttöjärjestelmän osille.

KÄÄNTÄMINEN

Kääntämisellä tarkoitetaan toimenpidettä, jossa ohjelman lähdekoodin perusteella luodaan konekielinen ohjelma, joka on tietokoneen ymmärtämässä muodossa.

LAITETIEDOSTO

Device node; erikoistiedosto; viittaus tietokoneen eri laitteisiin. Voi edustaa kiintolevyä, äänikortteja ja muita ”oikeita laitteita” sekä järjestelmän sisäisiä toimintoja, kuten esimerkiksi satunnaislukugeneraattoria (esim. `/dev/random`).

LIITOSPISTE (Mount point)

Hakemisto (yleensä tyhjä) sillä hetkellä saatavilla olevassa tiedostojärjestelmässä, jonne lisätiedostojärjestelmä liitetään.

LINKKERI, LINKITTÄJÄ

Linker; Ohjelma, joka yhdistää aliohjelmakirjastot (lyhyemmin kirjastot) mukaan konekieliseen ohjelmaan.

LINUX STANDARD BASE (LSB)

Linux-käyttöjärjestelmälle kehitetty standardi ja sertifiointi järjestelmän ohjelmistoista ja asetuksista. LSB:n tavoitteena on estää Linux-käyttöjärjestelmien pirstaloituminen eri käyttöjärjestelmiksi, jotka eivät ole yhteensopivia keskenään.

LOGICAL PARTITION (Looginen osio)

Levy voidaan jakaa vain neljään [ensisijaiseen partitioon](#) eli osioon, mutta on kuitenkin mahdollista tehdä useita osioita jakamalla yksi (tai useampi) neljästä primary-osioista aliosioiksi. Näitä aliosioita kutsutaan loogisiksi osioiksi.

LOKALISOINTI (Kotoistus)

Lokalisointi eli kotoistus tarkoittaa sovelluksien kääntämistä ja sopeuttamista kieleen ja kulttuuriin. Sovellusten ja ohjeiden kääntämisen lisäksi kotoistuksen kohteita ovat kalenterit, merkistöt, päivämäärien näyttötavat yms.

LÄHDEKOODI

Source code; ”sorsa”; Ohjelmointikielellä toteutettu ohjelman täydellinen kuvaus tekstimuodossa. Lähdekoodi pitää kääntää, jotta siitä saadaan toimiva tietokoneohjelma. Lähdekoodi on ihmisen tulkittavissa, jos osaa kyseisen ohjelmointikielen. Käännetty konekielinen koodi on ihmiselle yleensä mahdotonta tulkita.

MAKE

Kääntää ohjelman. `Make install` kopioi käännetyt binäärit lopulliseen asennushakemistoonsa. `Make check` -komennolla voidaan tarkistaa tehty käänнос. `Make clean` poistaa luodut binäärit ja Makefilen lähdekoodihakemistosta.

MAKEFILE

Configuren luoma tiedosto, jonka perusteella make kääntää ohjelman lähdekoodin. Joissain paketeissa on valmiina makefile, jolloin configurea ei tarvitse ajaa.

MOUNT

Komentorivikomennolla mount liitetään hakemistopuuhun eri tiedostojärjestelmiä, esimerkiksi levyosioita, verkkolevyjä ja levykuvia. Liitoksen purkamista kutsutaan irrottamiseksi (unmounting).

NULL (/dev/null)

Nollalaite; musta aukko; dataviemäri; /dev-hakemistossa sijaitseva laitetiedosto, johon kaikilla käyttäjillä on kirjoitusoikeus, mutta johon kirjoitetettu data katoaa. Mikä tahansa ohjelma voi avata tiedoston. /dev/null on tyhjä. Se ei sisällä mitään. Kaikki tiedostoon kirjoitettu data ilmoitetaan ohjelmalle hyväksytyksi, mutta dataa ei tallenneta mihinkään. /dev/null -tiedostoa käytetään prosessin virheelisen ulostulon tai tiedoston sisällön hävittämiseen.

POSIX

Portable **O**perating **S**ystem **I**nterface for uni**X**; Unix-käyttöjärjestelmille kehitetty IEEE-standardi-/määrittelykokonaisuus. Linux toteuttaa POSIX-yhteensopivuuden Linux Standard Base (LSB) -standardin avulla.

PRIMARY PARTITION (Ensisijainen partitio)

Mikä tahansa neljästä mahdollisesta ensimmäisen asteen partitiosta, johon HDD:n voi jakaa. Ensisijaisia partitioita voi olla vain neljä. Vain ensisijaista partitiota voi käyttää aktiivisena partitiona. Aktiivinen partitio sisältää käyttöjärjestelmän, jonka tietokone yrittää oletuksena ladata muistiin, kun kone käynnistetään.

SHELL

Komentotulkki; ohjelma, joka toimii käyttäjän ja Linux-[kernelin](#) välissä. Shell lukee käyttäjän antamia komentoja ja kääntää ne kernelin ymmärtämään muotoon.

SKELETON-HAKEMISTO

/etc/skel -hakemisto sisältää tiedostot ja kansiot, jotka kopioidaan uuden käyttäjän kotihakemistoon (login-hakemisto), kun käyttäjä luodaan.

SPLASH SCREEN

Graafinen käyttöliittymä (GUI); Interaktiivisen ohjelman näyttämä aloitusnäyttö, jossa tavallisesti on logo, versioinfo ja copyright-huomautus.

SWAP

Swap toimii ekstra-RAM:ina silloin, jos RAM loppuu koneesta kesken. Jos kone käyttää jatkuvasti swapia, tietää, että RAM:ia on liian vähän. Jatkuva swapin käyttö hidastaa konetta huomattavasti.

SYMBOLINEN LINKKI

Tiedostopuussa sijaitseva viittaus, joka voi osoittaa tiedostoon, kokonaiseen tiedostojärjestelmään, itseensä tai kohteeseen, jota ei ole edes olemassa. Symbolisia linkkejä käytetään jatkuvasti linkittämään kirjastoja ja varmistamaan, että tiedostot ovat johdonmukaisessa paikassa liikuttamatta tai kopioimatta alkuperäisiä tiedostoja.

Symbolinen linkki osoittaa tiedostoon sen nimellä. Siispä vaikka alkuperäinen tiedosto korvattaisiin toisella siten, että nimi pysyy samana, osoittaa se nyt uuteen tiedostoon. Nimen ei tarvitse olla olemassa ja se voi jopa sijaita eri tiedostojärjestelmässä.

THIN CLIENT (päätelaitte)

Kevyt asiakaspääte; riisuttu tietokone. Tarjoaa esimerkiksi pelkästään WWW-se-laimen. Suurin osa järjestelmän toiminnoista ja tiedostoista sijaitsee palvelimella. Vanhat tietokoneet kierrätetään usein päätelaitteiksi, jolloin niiden elinikä pitenee.

TOOLCHAIN (Työkaluketju)

Työkaluketju on setti erillisiä kehitystyökaluja, jotka on linkitetty toisiinsa. Työkaluketjua käytetään kääntämisessä. Se sisältää yleensä esimerkiksi GCC:n, binutilsin ja glibc:n. Työkaluketju voi sisältää myös muita työkaluja esimerkiksi debuggausta varten.

UDEV

Linuxin laitteistohallintajärjestelmä, jonka pääasiallinen tehtävä on ylläpitää /dev-hakemistohierarkiaa.

YMPÄRISTÖMUUTTUJA

Parametri, joka on voimassa lokaalin [shellin](#) ulkopuolella.

7 PAKETTIILISTA

"Asennus riippuu" -kohdassa on listattu paketit, jotka pitää olla saatavilla ennen kuin asentaa kyseisen paketin. "Testit riippuvat" -kohdassa on listattu paketit, jotka vaaditaan "Asennus riippuu" -kohdassa olevien pakettien lisäksi testien ajamiseen. "Täytyy olla asennettu ennen" -kohdassa on listattu paketit, jotka vaativat kyseisen paketin olevan asennettu ennen kuin ne voidaan asentaa.

Arvioidussa käännösajassa on + -merkillä erotettu [2.8-luvussa](#) ja [2.15-luvussa](#) tahtuvat käännökset.

Kaikkien pakettien latausten koko yhteensä on noin 352 MB.

Acl-2.2.52

Kuvaus: Paketti sisältää työkaluja Access Control List:n hallitsemiseen, joita käytetään määrittämään tarkemmin oikeuksia tiedostoille ja hakemistoille.

Arvioitu käännösaika: alle 0,1 SBU

Latauksen koko: 380 KB

Vaadittava levytila: 4,8 MB

Lataa: <http://download.savannah.gnu.org/releases/acl/acl-2.2.52.src.tar.gz>

MD5 sum: a61415312426e9c2212bd7dc7929abda

Täytyy olla asennettu ennen: Coreutils, Sed, Tar ja Vim

Asennus riippuu: Attr, Bash, Binutils, Coreutils, GCC, Gettext, Grep, M4, Make, Perl, Sed ja Texinfo

Testit riippuvat: Automake, Diffutils, Findutils ja Libtool

Attr-2.4.47

Kuvaus: Paketti sisältää työkaluja tiedostojärjestelmän objektien laajennettujen attribuuttien hallinnoimiseen.

Arvioitu käännösaika: alle 0,1 SBU

Latauksen koko: 336 KB

Vaadittava levytila: 3,3 MB

Kotisivut: <http://savannah.nongnu.org/projects/attr>

Lataa: <http://download.savannah.gnu.org/releases/attr/attr-2.4.47.src.tar.gz>

MD5 sum: 84f58dec00b60f2dc8fd1c9709291cc7

Täytyy olla asennettu ennen: Acl ja Libcap

Asennus riippuu: Bash, Binutils, Coreutils, GCC, Gettext, Grep, M4, Make, Perl, Sed ja Texinfo

Testit riippuvat: Automake, Diffutils, Findutils ja Libtool

Autoconf-2.69

Kuvaus: Paketti sisältää ohjelmia, joilla tuotetaan shell-skriptejä, jotka voivat automaattisesti konfiguroida lähdekoodia.

Arvioitu käännösaika: alle 0,1 SBU (n. 3,5 SBU testien kanssa)

Latauksen koko: 1,186 KB

Vaadittava levytila: 17,3 MB

Kotisivut: <http://www.gnu.org/software/autoconf/>

Lataa: <http://ftp.gnu.org/gnu/autoconf/autoconf-2.69.tar.xz>

MD5 sum: 50f97f4159805e374639a73e2636f22e

Täytyy olla asennettu ennen: Automake

Asennus riippuu: Bash, Coreutils, Grep, M4, Make, Perl, Sed ja Texinfo

Testit riippuvat: Automake, Diffutils, Findutils, GCC ja Libtool

Automake-1.15

Kuvaus: Paketti sisältää ohjelmia Makefiles:ien generoimiseen Autoconf:lle käytettäväksi.

Arvioitu käännösaika: alle 0,1 SBU (n. 7,5 SBU testien kanssa)

Latauksen koko: 1,462 KB

Vaadittava levytila: 110 MB

Kotisivut: <http://www.gnu.org/software/automake/>

Lataa: <http://ftp.gnu.org/gnu/automake/automake-1.15.tar.xz>

MD5 sum: 9a1ddb0e053474d9d1105cfe39b0c48d

Täytyy olla asennettu ennen: -

Asennus riippuu: Autoconf, Bash, Coreutils, Gettext, Grep, M4, Make, Perl, Sed ja Texinfo

Testit riippuvat: Binutils, Bison, Bzip2, DejaGNU, Diffutils, Expect, Findutils, Flex, GCC, Gettext, Gzip, Libtool ja Tar

Bash-4.4

Kuvaus: Paketti sisältää Bourne-Again Shell:n (bash).

Arvioitu käännösaika: 0,4 SBU + 1,7 SBU

Latauksen koko: 9,158 KB

Vaadittava levytila: 61 MB

Kotisivut: <http://www.gnu.org/software/bash/>

Lataa: <http://ftp.gnu.org/gnu/bash/bash-4.4.tar.gz>

MD5 sum: 148888a7c95ac23705559b6f477dfe25

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Bison, Coreutils, Diffutils, Gawk, GCC, Glibc, Grep, Make, Ncurses, Patch, Readline, Sed ja Texinfo

Testit riippuvat: Shadow

Bc-1.06.95

Kuvaus: Paketti sisältää mielivaltaisen tarkan laskennan (arbitrary precision arithmetic) prosessointikielen.

Arvioitu käännösaika: 0,1 SBU

Latauksen koko: 288 KB

Vaadittava levytila: 3,6 MB

Kotisivut: <http://www.gnu.org/software/bc/>

Lataa: <http://alpha.gnu.org/gnu/bc/bc-1.06.95.tar.bz2>

MD5 sum: 5126a721b73f97d715bb72c13c889035

Täytyy olla asennettu ennen: Linux Kernel

Asennus riippuu: Bash, Binutils, Bison, Coreutils, GCC, Glibc, Grep, Make ja Readline

Testit riippuvat: Gawk

Binutils-2.27

Kuvaus: Paketti sisältää linkkerin, kääntäjän ja muita työkaluja objektitiedostojen käsittelyyn.

Arvioitu käännösaika: Osa 1: 1 SBU; Osa 2: 1,1 SBU + 5,7 SBU

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, Diffutils, File, Gawk, GCC, Glibc, Grep, Make, Perl, Sed, Texinfo ja Zlib

Testit riippuvat: DegaGNU ja Expect

Latauksen koko: 25,488 KB

Vaadittava levytila: Osa 1: 519 MB + Osa 2: 533 MB = yht. 1052 MB

Kotisivut: <http://www.gnu.org/software/binutils/>

Lataa: <http://ftp.gnu.org/gnu/binutils/binutils-2.27.tar.bz2>

MD5 sum: 2869c9bf3e60ee97c74ac2a6bf4e9d68

Bison-3.0.4

Kuvaus: Paketti sisältää jäsentäjägeneraattorin.

Arvioitu käännösaika: 0,3 SBU

Latauksen koko: 1,928 KB

Vaadittava levytila: 32 MB

Kotisivut: <http://www.gnu.org/software/bison/>

Lataa: <http://ftp.gnu.org/gnu/bison/bison-3.0.4.tar.xz>

MD5 sum: c342201de104cc9ce0a21e0ad10d4021

Täytyy olla asennettu ennen: Kbd ja Tar

Asennus riippuu: Bash, Binutils, Coreutils, GCC, Gettext, Glibc, Grep, M4, Make, Perl ja Sed

Testit riippuvat: Diffutils, Findutils ja Flex

Bzip2-1.0.6

Kuvaus: Paketti sisältää ohjelmia tiedostojen pakkaamiseen ja purkamiseen.

Arvioitu käännösaika: alle 0,1 SBU

Latauksen koko: 764 KB

Vaadittava levytila: 5,2 MB

Kotisivut: <http://www.bzip.org/>

Lataa: <http://www.bzip.org/1.0.6/bzip2-1.0.6.tar.gz>

MD5 sum: 00b516f4704d4a7cb50a1d97e6e8e15b

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, GCC, Glibc, Make ja Patch

Testit riippuvat: -

Check-0.11.0

Kuvaus: Yksikköjen testauskehys C:lle.

Arvioitu käännösaika: 0,1 SBU

Latauksen koko: 736 KB

Vaadittava levytila: 11 MB

Kotisivut: <https://libcheck.github.io/check>

Lataa: <https://github.com/libcheck/check/releases/download/0.11.0/check-0.11.0.tar.gz>

MD5 sum: 9b90522b31f5628c2e0f55dda348e558

Täytyy olla asennettu ennen: -

Asennus riippuu: GCC, Grep, Make, Sed ja Texinfo

Testit riippuvat: -

Coreutils-8.26

Kuvaus: Paketti sisältää monia perustyökaluja, kuten cat, rm ja ls.

Arvioitu käännösaika: 0,6 SBU + 3,1 SBU

Latauksen koko: 5,676 KB

Vaadittava levytila: 136 MB

Kotisivut: <http://www.gnu.org/software/coreutils/>

Lataa: <http://ftp.gnu.org/gnu/coreutils/coreutils-8.26.tar.xz>

MD5 sum: d5aa2072f662d4118b9f4c63b94601a6

Täytyy olla asennettu ennen: Bash, Diffutils, Findutils, Man-DB, ja Eudev

Asennus riippuu: Bash, Binutils, Coreutils, GCC, Gettext, Glibc, GMP, Grep, Make, Patch, Perl, Sed ja Texinfo

Testit riippuvat: Diffutils, E2fsprogs, Findutils, Shadow ja Util-linux

DejaGNU-1.6

Kuvaus: Paketti sisältää kehykset muiden ohjelmien testaukseen.

Arvioitu käännösaika: alle 0,1 SBU

Latauksen koko: 512 KB

Vaadittava levytila: 3,2 MB

Kotisivut: <http://www.gnu.org/software/dejagnu/>

Lataa: <http://ftp.gnu.org/gnu/dejagnu/dejagnu-1.6.tar.gz>

MD5 sum: 1fdc2eb0d592c4f89d82d24dfdf02f0b

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Coreutils, GCC, Grep, Make ja Sed

Testit riippuvat: -

Diffutils-3.5

Kuvaus: Paketti sisältää ohjelmia, jotka näyttävät tiedostojen ja hakemistojen erot.

Arvioitu käännösaika: 0,2 SBU + 0,4 SBU

Latauksen koko: 1,330 KB

Vaadittava levytila: 22 MB

Kotisivut: <http://www.gnu.org/software/diffutils/>

Lataa: <http://ftp.gnu.org/gnu/diffutils/diffutils-3.5.tar.xz>

MD5 sum: 569354697ff1cfc9a9de3781361015fa

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, Gawk, GCC, Gettext, Glibc, Grep, Make, Sed ja Texinfo

Testit riippuvat: Diffutils ja Perl

E2fsprogs-1.43.4

Kuvaus: Paketti sisältää työkaluja ext2-tiedostojärjestelmien hallitsemiseen. Se tukee myös ext3:a ja ext4:ä.

Arvioitu käännösaika: 4,1 SBU

Latauksen koko: 7,376 KB

Vaadittava levytila: 57 MB

Home page: <http://e2fsprogs.sourceforge.net/>

Download: <http://downloads.sourceforge.net/project/e2fsprogs/e2fsprogs/v1.43.4/e2fsprogs-1.43.4.tar.gz>

MD5 sum: 0bd1c74f357f6e9ae2ab6fa6229b9aea

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, Diffutils, Gawk, GCC, Glibc, Grep, Gzip, Make, Sed, Texinfo ja Util-linux

Testit riippuvat: Procps-ng ja Psmisc

Eudev-3.2.1

Kuvaus: Paketti sisältää ohjelmia laitetiedostojen dynaamiseen luomiseen.

Arvioitu käännösaika: 0,2 SBU

Latauksen koko: 1,778 KB

Vaadittava levytila: 78 MB

Lataa: <http://dev.gentoo.org/~blueness/eudev/eudev-3.2.1.tar.gz>

MD5 sum: 49f71f6c271ffa95b3c20d757d45c2e5

Asennus riippuu: Bash, Binutils, Coreutils, Gawk, GCC, Glibc, Grep, Gperf, Make ja Sed

Täytyy olla asennettu ennen: -

Testit riippuvat: -

Expat-2.2.0

Kuvaus: Paketti sisältää C-kirjaston XML-jäsentämiseen.

Arvioitu käännösaika: alle 0,1 SBU

Latauksen koko: 405 KB

Vaadittava levytila: 6,1 MB

Kotisivut: <http://expat.sourceforge.net/>

Lataa: <http://prdownloads.sourceforge.net/expat/expat-2.2.0.tar.bz2>

MD5 sum: 2f47841c829facb346eb6e3fab5212e2

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, Diffutils, GCC, Glibc, Grep, Make, Patch, Sed ja Tcl

Testit riippuvat: -

Expect-5.45

Kuvaus: Paketti sisältää ohjelman skriptattujen dialogien suorittamiseen toisilla interaktiivisilla ohjelmilla. Expect "keskustelee" toisten ohjelmien kanssa skriptin mukaan. Skriptin avulla se tietää, mitä ohjelmalta voi odottaa ja mikä on oikea vastaus siihen.

Arvioitu käännösaika: 0,1 SBU

Latauksen koko: 614 KB

Vaadittava levytila: 4,3 MB

Kotisivut: <http://expect.sourceforge.net/>

Lataa: <http://prdownloads.sourceforge.net/expect/expect5.45.tar.gz>

MD5 sum: 44e1a4f4c877e9ddc5a542dfa7ecc92b

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, Diffutils, GCC, Glibc, Grep, Make, Patch, Sed ja Tcl

Testit riippuvat: -

File-5.30

Kuvaus: Paketti sisältää työkalun, joka määrittää annetun tiedoston/tiedostojen tyyppin.

Arvioitu käännösaika: 0,1 SBU

Latauksen koko: 771 KB

Vaadittava levytila: 16 MB

Kotisivut: <http://www.darwinsys.com/file/>

Lataa: <ftp://ftp.astron.com/pub/file/file-5.30.tar.gz>

HUOM! Paketti ei välttämättä löydy enää yllä mainitusta latausosoitteesta, koska sivun adminit toisinaan poistavat vanhat versiot sitä mukaa kun uusia julkaistaan.

MD5 sum: f35eaf05489ae566eafc4d26bb1dcd90

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, Diffutils, Gawk, GCC, Glibc, Grep, Make, Sed ja Zlib

Testit riippuvat: -

Findutils-4.6.0

Kuvaus: Paketti sisältää ohjelmia tiedostojen löytämiseen.

Arvioitu käännösaika: 0,3 SBU + 0,9 SBU

Latauksen koko: 3,692 KB

Vaadittava levytila: 35 MB

Kotisivut: <http://www.gnu.org/software/findutils/>

Lataa: <http://ftp.gnu.org/gnu/findutils/findutils-4.6.0.tar.gz>

MD5 sum: 9936aa8009438ce185bea2694a997fc1

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, GCC, Gettext, Glibc, Grep, Make, Sed ja Texinfo

Testit riippuvat: DejaGNU, Diffutils ja Expect

Flex-2.6.3

Kuvaus: Paketti sisältää työkalun sellaisten ohjelmien generointiin, jotka tunnistavat kaavat teksteistä.

Arvioitu käännösaika: 0,4 SBU

Latauksen koko: 1,373 KB

Vaadittava levytila: 32 MB

Kotisivut: <http://flex.sourceforge.net>

Lataa: <https://github.com/westes/flex/releases/download/v2.6.3/flex-2.6.3.tar.gz>

MD5 sum: a5f65570cd9107ec8a8ec88f17b31bb1

Täytyy olla asennettu ennen: IPRoute2, Kbd ja Man-DB

Asennus riippuu: Bash, Binutils, Coreutils, GCC, Gettext, Glibc, Grep, M4, Make, Patch, Sed ja Texinfo

Testit riippuvat: Bison ja Gawk

Gawk-4.1.4

Kuvaus: Paketti sisältää ohjelmia tekstitiedostojen käsittelyyn.

Arvioitu käännösaika: 0,2 SBU + 0,3 SBU

Latauksen koko: 2,313 KB

Vaadittava levytila: 35 MB

Kotisivut: <http://www.gnu.org/software/gawk/>

Lataa: <http://ftp.gnu.org/gnu/gawk/gawk-4.1.4.tar.xz>

MD5 sum: 4e7dbc81163e60fd4f0b52496e7542c9

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, GCC, Gettext, Glibc, Grep, Make, MPFR, Patch, Readline, Sed ja Texinfo

Testit riippuvat: Diffutils

GCC-6.3.0

Kuvaus: Paketti sisältää GNU-projektin kääntäjien kokoelman, johon sisältyvät C- ja C++ -kääntäjät.

Arvioitu käännösaika: Osa 1: 8,4 SBU; Osa 2: 11 SBU + 82 (testien kanssa)

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, Diffutils, Findutils, Gawk, GCC, Gettext, Glibc, GMP, Grep, M4, Make, MPC, MPFR, Patch, Perl, Sed, Tar ja Texinfo

Testit riippuvat: DejaGNU ja Expect

Latauksen koko: 97,562 KB

Vaadittava levytila: Osa 1: 2,5 GB + Osa 2: 2,9 GB = 5,4 GB

Kotisivut: <http://gcc.gnu.org/>

Lataa: <http://ftp.gnu.org/gnu/gcc/gcc-6.3.0/gcc-6.3.0.tar.bz2>

MD5 sum: 677a7623c7ef6ab99881bc4e048debb6

GDBM-1.12

Kuvaus: Paketti sisältää tietokantahallinnan.

Arvioitu käännösaika: 0,1 SBU

Latauksen koko: 822 KB

Vaadittava levytila: 9 MB

Kotisivut: <http://www.gnu.org/software/gdbm/>

Lataa: <http://ftp.gnu.org/gnu/gdbm/gdbm-1.12.tar.gz>

MD5 sum: 9ce96ff4c99e74295ea19040931c8fb9

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, Diffutils, GCC, Grep, Make ja Sed

Testit riippuvat: -

Gettext-0.19.8.1

Kuvaus: Paketti sisältää työkaluja kansainvälistämiseen ja paikallistamiseen. Ne mahdollistavat ohjelmien kääntämisen NLS:ää käyttäen (Native Language Support) . Näitä ohjelmia käyttäjä voi käyttää omalla kielellään.

Arvioitu käännösaika: 0,9 SBU + 2,9 SBU

Latauksen koko: 7,041 KB

Vaadittava levytila: 164 MB

Kotisivut: <http://www.gnu.org/software/gettext/>

Lataa: <http://ftp.gnu.org/gnu/gettext/gettext-0.19.8.1.tar.xz>

MD5 sum: df3f5690eaa30fd228537b00cb7b7590

Täytyy olla asennettu ennen: Automake

Asennus riippuu: Bash, Binutils, Coreutils, Gawk, GCC, Glibc, Grep, Make, Sed ja Texinfo

Testit riippuvat: Diffutils, Perl ja Tcl

Glibc-2.25

Kuvaus: Paketti sisältää C-pääkirjaston. Kirjasto tarjoaa perusrutiinit muistin varaukseen, hakemistojen etsimiseen, tiedostojen avaamiseen ja sulkemiseen, tiedostojen lukemiseen ja muokkaukseen, merkkijonojen käsittelyyn, kuvioiden yhteensopivuuteen, aritmetiikkaa jne.

Arvioitu käännösaika: 4,1 SBU + 20 SBU

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, Diffutils, Gawk, GCC, Gettext, Grep, Gzip, Linux API Headers, Make, Perl, Sed ja Texinfo

Testit riippuvat: File

Latauksen koko: 13,549 KB

Vaadittava levytila: 753 MB

Kotisivut: <http://www.gnu.org/software/libc/>

Lataa: <http://ftp.gnu.org/gnu/glibc/glibc-2.25.tar.xz>

MD5 sum: 1496c3bf41adf9db0ebd0af01f202eed

GMP-6.1.2

Kuvaus: Paketti sisältää matematiikkakirjastoja ja hyödyllisiä toimintoja mielivaltaisen tarkkaa aritmetiikkaa (arbitrary precision arithmetic) varten.

Arvioitu käännösaika: 1,3 SBU

Latauksen koko: 1,901 KB

Vaadittava levytila: 59 MB

Kotisivut: <http://www.gnu.org/software/gmp/>

Lataa: <http://ftp.gnu.org/gnu/gmp/gmp-6.1.2.tar.xz>

MD5 sum: f58fa8001d60c4c77595fbbb62b63c1d

Täytyy olla asennettu ennen: MPFR ja GCC

Asennus riippuu: Bash, Binutils, Coreutils, Diffutils, Gawk, GCC, Glibc, Grep, M4, Make, Sed ja Texinfo

Testit riippuvat: -

Gperf-3.0.4

Kuvaus: Gperf generoi täydellisen tarkistusfunktion avainjoukosta.

Arvioitu käännösaika: alle 0,1 SBU

Latauksen koko: 961 KB

Vaadittava levytila: 5,4 MB

Kotisivut: <http://www.gnu.org/software/gperf/>

Lataa: <http://ftp.gnu.org/gnu/gperf/gperf-3.0.4.tar.gz>

MD5 sum: c1f1db32fb6598d6a93e6e88796a8632

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, GCC, Glibc ja Make

Testit riippuvat: Diffutils ja Expect

Grep-3.0

Kuvaus: Paketti sisältää ohjelmia tiedostoista etsimiseen.

Latauksen koko: 1,342 KB

Vaadittava levytila: 19 MB

Kotisivut: <http://www.gnu.org/software/grep/>

Lataa: <http://ftp.gnu.org/gnu/grep/grep-3.0.tar.xz>

MD5 sum: fa07c1616adeb9c3262be5177d10ad4a

Täytyy olla asennettu ennen: Automake

Asennus riippuu: Bash, Binutils, Coreutils, Gawk, GCC, Glibc, Grep, Make, Sed ja Texinfo

Testit riippuvat: Diffutils, Perl ja Tcl

Groff-1.22.3

Kuvaus: Paketti sisältää ohjelmia tekstien prosessointiin ja formatointiin.

Latauksen koko: 4,091 KB

Vaadittava levytila: 82 MB

Kotisivut: <http://www.gnu.org/software/groff/>

Lataa: <http://ftp.gnu.org/gnu/groff/groff-1.22.3.tar.gz>

MD5 sum: cc825fa64bc7306a885f2fb2268d3ec5

Täytyy olla asennettu ennen: Man-DB ja Perl

Asennus riippuu: Bash, Binutils, Bison, Coreutils, Gawk, GCC, Glibc, Grep, Make, Patch, Sed ja Texinfo

Testit riippuvat: Ei testejä saatavilla

GRUB-2.02 beta3

Kuvaus: Paketti sisältää käyttöjärjestelmän lataajan, Grand Unified Bootloader:n.

Latauksen koko: 5,890 KB

Vaadittava levytila: 142 MB

Kotisivut: <http://www.gnu.org/software/grub/>

Lataa: <http://alpha.gnu.org/gnu/grub/grub-2.02~beta3.tar.xz>

MD5 sum: ab399fc6f74a97d66ff77f04b743149c

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Bison, Coreutils, Diffutils, GCC, Gettext, Glibc, Grep, Make, Ncurses, Sed, Texinfo ja Xz

Testit riippuvat: -

Gzip-1.8

Kuvaus: Paketti sisältää ohjelmia tiedostojen pakkaamiseen ja purkamiseen.

Latauksen koko: 712 KB

Vaadittava levytila: 9 MB

Kotisivut: <http://www.gnu.org/software/gzip/>

Lataa: <http://ftp.gnu.org/gnu/gzip/gzip-1.8.tar.xz>

MD5 sum: f7caabb65cddc1a4165b398009bd05b9

Täytyy olla asennettu ennen: Man-DB

Asennus riippuu: Bash, Binutils, Coreutils, GCC, Glibc, Grep, Make, Sed ja Texinfo

Testit riippuvat: Diffutils ja Less

Iana-Etc-2.30

Kuvaus: Paketti tarjoaa dataa verkkopalveluille ja protokollille.

Latauksen koko: 201 KB

Vaadittava levytila: 2,3 MB

Kotisivut: <http://freecode.com/projects/iana-etc>

Lataa: <http://anduin.linuxfromscratch.org/LFS/iana-etc-2.30.tar.bz2>

MD5 sum: 3ba3afb1d1b261383d247f46cb135ee8

Täytyy olla asennettu ennen: Perl

Asennus riippuu: Coreutils, Gawk ja Make

Testit riippuvat: Ei testejä saatavilla

Inetutils-1.9.4

Kuvaus: Paketti on kokoelma yleisiä verkko-ohjelmia, kuten ftp, telnet, rsh jne.

Latauksen koko: 1,333 KB

Vaadittava levytila: 27 MB

Kotisivut: <http://www.gnu.org/software/inetutils/>

Lataa: <http://ftp.gnu.org/gnu/inetutils/inetutils-1.9.4.tar.xz>

MD5 sum: 87fef1fa3f603aef11c41dcc097af75e

Täytyy olla asennettu ennen: Tar

Asennus riippuu: Bash, Binutils, Coreutils, GCC, Glibc, Grep, Make, Ncurses, Patch, Sed, Texinfo ja Zlib

Testit riippuvat: Ei testejä saatavilla

Intltool-0.51.0

Kuvaus: Intltool on kansainvälistymistyökalu, jota käytetään erottamaan kielellisesti käännettävät osat lähdekoodista.

Latauksen koko: 159 KB

Vaadittava levytila: 1,5 MB

Kotisivut: <http://freedesktop.org/wiki/Software/intltool>

Lataa: <http://launchpad.net/intltool/trunk/0.51.0/+download/intltool-0.51.0.tar.gz>

MD5 sum: 12e517cac2b57a0121cda351570f1e63

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Gawk, Glibc, Make, Perl, Sed ja XML::Parser

Testit riippuvat: Perl

IPRoute2-4.9.0

Kuvaus: Paketti sisältää kokoelman työkaluja TCP/IP -verkotuksen hallintaan ja monitorointiin. Iproute2 on ollut standardityökalu Linuxeissa jo 2000-luvun alusta lähtien.

Latauksen koko: 599 KB

Vaadittava levytila: 11 MB

Kotisivut: <https://www.kernel.org/pub/linux/utils/net/iproute2/>

Lataa: <https://www.kernel.org/pub/linux/utils/net/iproute2/iproute2-4.9.0.tar.xz>

MD5 sum: 44a8371a4b2c40e48e4c9f98cbd41391

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Bison, Coreutils, Flex, GCC, Glibc, Make ja Linux API Headers

Testit riippuvat: Ei testejä saatavilla

Kbd-2.0.4

Kuvaus: Paketti sisältää työkaluja Linuxin konsolin hallintaan (konsoli, virtuaaliterminaalit, näppäimistöt jne.). Pääasiassa paketti lataa konsolifontit ja näppäimistökartat.

Latauksen koko: 1,008 KB

Vaadittava levytila: 29 MB

Kotisivut: <http://ftp.altlinux.org/pub/people/legion/kbd>

Lataa: <https://www.kernel.org/pub/linux/utils/kbd/kbd-2.0.4.tar.xz>

MD5 sum: c1635a5a83b63aca7f97a3eab39ebaa6

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Bison, Check, Coreutils, Flex, GCC, Gettext, Glibc, Gzip, Make, Patch ja Sed

Testit riippuvat: Ei testejä saatavilla

Kmod-23

Kuvaus: Paketti sisältää kirjastoja ja työkaluja kernel-moduulien lataamiseen.

Latauksen koko: 440 KB

Vaadittava levytila: 10,3 MB

Lataa: <https://www.kernel.org/pub/linux/utils/kernel/kmod/kmod-23.tar.xz>

MD5 sum: 3cf469f40ec2ed51f56ba45ea03793e7

Täytyy olla asennettu ennen: Eudev

Asennus riippuu: Bash, Binutils, Bison, Coreutils, Flex, GCC, Gettext, Glibc, Gzip, Make, Sed, Xz-Utils ja Zlib

Testit riippuvat: Ei testejä saatavilla

Less-481

Kuvaus: Paketti sisältää tekstitiedostojen näyttäjän.

Latauksen koko: 310 KB

Vaadittava levytila: 3,5 MB

Home page: <http://www.greenwoodsoftware.com/less/>

Download: <http://www.greenwoodsoftware.com/less/less-481.tar.gz>

MD5 sum: 50ef46065c65257141a7340123527767

Täytyy olla asennettu ennen: Gzip

Asennus riippuu: Bash, Binutils, Coreutils, Diffutils, GCC, Glibc, Grep, Make, Ncurses ja Sed
Testit riippuvat: Ei testejä saatavilla

Libcap-2.25

Kuvaus: Libcap noudattaa jo haudattua POSIX1003.1e -standardia. Linuxin kykyjako (capabilities) jakaa kaikkivoipaisen rootin oikeudet erillisiin tarkempiin oikeuksiin. Tällöin tietylle operaatiolle voidaan antaa rootin oikeudet ilman, että koko prosessi saa rootin oikeuksia. Käytännössä kykyjako asettaa rajoituksia root-oikeudet tarvitseville ohjelmille, jotta ne saavat oikeudet vain tiettyihin operaatioihin.

Latauksen koko: 64 KB

Vaadittava levytila: 1,3 MB

Kotisivut: <https://sites.google.com/site/fullycapable/>

Lataa: <https://www.kernel.org/pub/linux/libs/security/linux-privs/libcap2/libcap-2.25.tar.xz>

MD5 sum: 6666b839e5d46c2ad33fc8aa2ceb5f77

Täytyy olla asennettu ennen: -

Asennus riippuu: Attr, Bash, Binutils, Coreutils, GCC, Glibc, Perl, Make ja Sed

Testit riippuvat: Ei testejä saatavilla

Libpipeline-1.4.1

Kuvaus: Paketti sisältää kirjaston alaprosessien putkituksen käsittelyyn joustavalla ja kätevällä tavalla.

Latauksen koko: 787 KB

Vaadittava levytila: 7,9 MB

Kotisivut: <http://libpipeline.nongnu.org/>

Lataa: <http://download.savannah.gnu.org/releases/libpipeline/libpipeline-1.4.1.tar.gz>

MD5 sum: e54590ec68d6c1239f67b5b44e92022c

Täytyy olla asennettu ennen: Man-DB

Asennus riippuu: Bash, Binutils, Coreutils, Diffutils, Gawk, GCC, Glibc, Grep, Make, Sed ja Texinfo

Testit riippuvat: Check

Libtool-2.4.6

Kuvaus: Paketti sisältää geneerisen GNU-kirjastoa tukevan skriptin. Libtool piilottaa jaettujen kirjastojen käytön monimutkaisuuden johdonmukaisen ja siirrettävän rajapinnan taakse.

Latauksen koko: 951 KB

Vaadittava levytila: 43 MB

Kotisivut: <http://www.gnu.org/software/libtool/>

Lataa: <http://ftp.gnu.org/gnu/libtool/libtool-2.4.6.tar.xz>

MD5 sum: 1bfb9b923f2c1339b4d2ce1807064aa5

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, Diffutils, Gawk, GCC, Glibc, Grep, Make, Sed ja Texinfo

Testit riippuvat: Autoconf, Automake ja Findutils

Linux-4.9.9 (API Headers)

Kuvaus: Linux API Header:it paljastavat kernelin API:n (ohjelmointirajapinta) Glibc:n käyttöön.

HUOM! Linux-kerneliä päivitetään usein tietoturvaheavoittuvuuksien paikkaamiseksi. Kaikista viimeisintä saatavissa olevaa 4.9.x-versiota pitäisi käyttää, ellei [LFS:n errata](#)-sivusto muuta sano.

Kotisivut: <http://www.kernel.org/>

Lataa: <https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.9.9.tar.xz>

Latauksen koko: 91,025 KB

Vaadittava levytila: 767 MB

MD5 sum: de43a1a9e3a1ad97467c9a413bcdf896

Asennus riippuu: Bash, Bc, Binutils, Coreutils, Diffutils, Findutils, GCC, Glibc, Grep, Gzip, Kmod, Make, Ncurses, Perl ja Sed

Testit riippuvat: Ei testejä saatavilla

Täytyy olla asennettu ennen: OpenSSL

M4-1.4.18

Kuvaus: Paketti sisältää makroprosessorin.

Latauksen koko: 1,180 KB

Vaadittava levytila: 19 MB

Kotisivut: <http://www.gnu.org/software/m4/>

Lataa: <http://ftp.gnu.org/gnu/m4/m4-1.4.18.tar.xz>

MD5 sum: 730bb15d96fffe47e148d1e09235af82

Täytyy olla asennettu ennen: Autoconf ja Bison

Asennus riippuu: Bash, Binutils, Coreutils, GCC, Glibc, Grep, Make, Sed ja Texinfo

Testit riippuvat: Diffutils

Make-4.2.1

Kuvaus: Paketti sisältää ohjelman pakettien kääntämiseen.

Latauksen koko: 1,375 KB

Vaadittava levytila: 12,5 MB

Kotisivut: <http://www.gnu.org/software/make/>

Lataa: <http://ftp.gnu.org/gnu/make/make-4.2.1.tar.bz2>

MD5 sum: 15b012617e7c44c0ed482721629577ac

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, GCC, Gettext, Glibc, Grep, Make, Sed ja Texinfo

Testit riippuvat: Perl ja Procs-ng

Man-DB-2.7.6.1

Kuvaus: Paketti sisältää ohjelmia man-sivujen löytämiseen ja näyttämiseen.

Latauksen koko: 1,506 KB

Vaadittava levytila: 30 MB

Kotisivut: <http://www.nongnu.org/man-db/>

Lataa: <http://download.savannah.gnu.org/releases/man-db/man-db-2.7.6.1.tar.xz>

MD5 sum: 2948d49d0ed7265f60f83aa4a9ac9268

Asennus riippuu: Bash, Binutils, Bzip2, Coreutils, Flex, GCC, GDBM, Gettext, Glibc, Grep, Groff, Gzip, Less, Libpipeline, Make, Sed ja Xz
Testit riippuvat: Util-linux
Täytyy olla asennettu ennen: -

Man-Pages

Kuvaus: Paketti sisältää yli 2200 manuaalisivua.
Latauksen koko: 1,486 KB
Vaadittava levytila: 27 MB
Kotisivut: <http://www.kernel.org/doc/man-pages/>
Lataa: <https://www.kernel.org/pub/linux/docs/man-pages/man-pages-4.09.tar.xz>
MD5 sum: 91c721409bbf823d8f62bee3a1fe8ae3
Asennus riippuu: Bash, Coreutils ja Make
Testit riippuvat: -
Täytyy olla asennettu ennen: -

MPC-1.0.3

Kuvaus: Paketti sisältää kirjaston monimutkaisten numeroiden aritmetiikkaan mielivaltaisella korkealla tarkkuudella (Arbitrary-precision arithmetic) sekä tulosten oikean pyöristämisen. Se laajentaa IEEE-754 -liukulukustandardin periaatteita.
Latauksen koko: 655 KB
Vaadittava levytila: 17 MB
Kotisivut: <http://www.multiprecision.org/>
Lataa: <http://www.multiprecision.org/mpc/download/mpc-1.0.3.tar.gz>
MD5 sum: d6a1d5f8ddea3abd2cc3e98f58352d26
Täytyy olla asennettu ennen: GCC
Asennus riippuu: Bash, Binutils, Coreutils, Diffutils, Gawk, GCC, Glibc, Grep, GMP, Make, MPFR, Sed ja Texinfo
Testit riippuvat: -

MPFR-3.1.5

Kuvaus: MPFR = multiple precision floating-point. Sisältää toimintoja mielivaltaisen tarkkaa aritmetiikkaa (arbitrary-precision arithmetic) varten.
Latauksen koko: 1,101 KB
Vaadittava levytila: 45 MB
Kotisivut: <http://www.mpfr.org/>
Lataa: <http://www.mpfr.org/mpfr-3.1.5/mpfr-3.1.5.tar.xz>
MD5 sum: c4ac246cf9795a4491e7766002cd528f
Täytyy olla asennettu ennen: Gawk, GCC
Asennus riippuu: Bash, Binutils, Coreutils, Diffutils, Gawk, GCC, Glibc, Grep, GMP, Make, Sed ja Texinfo
Testit riippuvat: -

Nano-2.6.3

Kuvaus: Yksinkertainen tekstieditori.
Latauksen koko: 1,0 MB
Vaadittava levytila: 12,5 MB
Kotisivut: <https://www.nano-editor.org/>

Lataa: <https://www.nano-editor.org/dist/v2.6/nano-2.6.3.tar.xz>

MD5 sum: b82fe6f019f9bc8e069c5c021096e280

Ncurses-6.0

Kuvaus: Ncurses on ohjelmointikirjasto, joka sisältää API:n (ohjelmointirajapinnan), jolla ohjelmoija voi kirjoittaa tekstipohjaisia käyttöliittymiä. Sen avulla voi tehdä GUI:n (graafisen käyttöliittymän) kaltaisia ohjelmia, jotka pyörivät pääte-emulaattorilla.

Latauksen koko: 3,059 KB

Vaadittava levytila: 38 MB

Kotisivut: <http://www.gnu.org/software/ncurses/>

Lataa: <http://ftp.gnu.org/gnu/ncurses/ncurses-6.0.tar.gz>

MD5 sum: ee13d052e1ead260d7c28071f46eefb1

Täytyy olla asennettu ennen: Bash, GRUB, Inetutils, Less, Procps-ng, Psmisc, Readline, Texinfo, Util-linux ja Vim

Asennus riippuu: Bash, Binutils, Coreutils, Diffutils, Gawk, GCC, Glibc, Grep, Make, Patch ja Sed

Testit riippuvat: Ei testejä saatavilla

Patch-2.7.5

Kuvaus: Paketti sisältää ohjelman tiedostojen muokkaamiseen ja luomiseen lisäämällä patch-tiedoston.

Latauksen koko: 711 KB

Vaadittava levytila: 11 MB

Kotisivut: <http://savannah.gnu.org/projects/patch/>

Lataa: <http://ftp.gnu.org/gnu/patch/patch-2.7.5.tar.xz>

MD5 sum: e3da7940431633fb65a01b91d3b7a27a

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, GCC, Glibc, Grep, Make ja Sed

Testit riippuvat: Diffutils

Perl-5.24.1

Kuvaus: PERL = Practical Extraction ja Report Language. Perl on skriptimäinen ohjelmointikieli.

Latauksen koko: 13,759 KB

Vaadittava levytila: 249 MB

Kotisivut: <http://www.perl.org/>

Lataa: <http://www.cpan.org/src/5.0/perl-5.24.1.tar.bz2>

MD5 sum: 178ee0e8fa544dbc76d99cf041e2c9f0

Täytyy olla asennettu ennen: Autoconf

Asennus riippuu: Bash, Binutils, Coreutils, Gawk, GCC, GDBM, Glibc, Grep, Groff, Make, Sed ja Zlib

Testit riippuvat: Iana-Etc ja Procps-ng

Pkg-config-0.29.1

Kuvaus: Paketti sisältää työkalun sisälletyn polun ja/tai kirjaston polkujen antamiseen käännöstyökalulle konfiguroinnin ja make file -komennon aikana.

Latauksen koko: 1,967 KB

Vaadittava levytila: 28 MB

Kotisivut: <http://www.freedesktop.org/wiki/Software/pkg-config>

Lataa: <https://pkg-config.freedesktop.org/releases/pkg-config-0.29.1.tar.gz>

MD5 sum: f739a28cae4e0ca291f82d1d41ef107d

Täytyy olla asennettu ennen: Kmod

Asennus riippuu: Bash, Binutils, Coreutils, Gawk, GCC, Glibc, Grep, Make, Popt ja Sed

Testit riippuvat: -

Procps-ng-3.3.12

Kuvaus: Paketti sisältää ohjelmia prosessien valvomiseen.

Latauksen koko: 826 KB

Vaadittava levytila: 14 MB

Kotisivut: <http://sourceforge.net/projects/procps-ng>

Lataa: <http://sourceforge.net/projects/procps-ng/files/Production/procps-ng-3.3.12.tar.xz>

MD5 sum: 957e42e8b193490b2111252e4a2b443c

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, GCC, Glibc, Make ja Ncurses

Testit riippuvat: DejaGNU

Psmisc-22.21

Kuvaus: Paketti sisältää ohjelmia käynnissä olevien prosessien informaation näyttämiseen.

Latauksen koko: 447 KB

Vaadittava levytila: 4 MB

Kotisivut: <http://psmisc.sourceforge.net/>

Lataa: <http://downloads.sourceforge.net/project/psmisc/psmisc/psmisc-22.21.tar.gz>

MD5 sum: 935c0fd6eb208288262b385fa656f1bf

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, GCC, Gettext, Glibc, Grep, Make, Ncurses ja Sed

Testit riippuvat: Ei testejä saatavilla

Readline-7.0

Kuvaus: Paketti on joukko kirjastoja, jotka tarjoavat komentorivieditoinnin ja mahdollisuuden muistaa historian.

Latauksen koko: 2,842 KB

Vaadittava levytila: 15 MB

Kotisivut: <http://cnswww.cns.cwru.edu/php/chet/readline/rltop.html>

Lataa: <http://ftp.gnu.org/gnu/readline/readline-7.0.tar.gz>

MD5 sum: 205b03a87fc83dab653b628c59b9fc91

Täytyy olla asennettu ennen: Bash, Gawk

Asennus riippuu: Bash, Binutils, Coreutils, Gawk, GCC, Glibc, Grep, Make, Ncurses, Patch, Sed ja Texinfo

Testit riippuvat: Ei testejä saatavilla

Sed-4.4

Kuvaus: Sed = Stream editor. Paketti sisältää tehokkaan ohjelmoitavan vuonsuodattimen tekstivirran manipulointiin.

Latauksen koko: 1,154 KB

Vaadittava levytila: 16 MB

Kotisivut: <http://www.gnu.org/software/sed/>

Lataa: <http://ftp.gnu.org/gnu/sed/sed-4.4.tar.xz>

MD5 sum: e0c583d4c380059abd818cd540fe6938

Täytyy olla asennettu ennen: E2sprogs, File, Libtool ja Shadow

Asennus riippuu: Bash, Binutils, Coreutils, GCC, Gettext, Glibc, Grep, Make, Sed ja Texinfo

Testit riippuvat: Diffutils ja Gawk

Shadow-4.4

Kuvaus: Paketti sisältää ohjelmia salasanojen turvalliseen hallintaan.

Latauksen koko: 1,593 KB

Vaadittava levytila: 42 MB

Lataa: <https://github.com/shadow-maint/shadow/releases/download/4.4/shadow-4.4.tar.xz>

MD5 sum: c06f8c2571b44899e60662f9ad259dd6

Täytyy olla asennettu ennen: Coreutils

Asennus riippuu: Bash, Binutils, Coreutils, Diffutils, Findutils, Gawk, GCC, Gettext, Glibc, Grep, Make ja Sed

Testit riippuvat: Ei testejä saatavilla

Sysklogd-1.5.1

Kuvaus: Paketti sisältää ohjelmia järjestelmän viestien loggaukseen, esimerkiksi viestien, jotka kernel lähettää, kun jotain epätavallista tapahtuu.

Latauksen koko: 88 KB

Vaadittava levytila: 0.6 MB

Kotisivut: <http://www.infodrom.org/projects/sysklogd/>

Lataa: <http://www.infodrom.org/projects/sysklogd/download/sysklogd-1.5.1.tar.gz>

MD5 sum: c70599ab0d037fde724f7210c2c8d7f8

Asennus riippuu: Binutils, Coreutils, GCC, Glibc, Make ja Patch

Testit riippuvat: -

Täytyy olla asennettu ennen: -

Sysvinit-2.88dsf

Kuvaus: Paketti sisältää ohjelmia järjestelmän käynnistykseen, ajamisen ja sammutuksen hallintaan.

Latauksen koko: 108 KB

Vaadittava levytila: 1,1 MB

Kotisivut: <http://savannah.nongnu.org/projects/sysvinit>

Lataa: <http://download.savannah.gnu.org/releases/sysvinit/sysvinit-2.88dsf.tar.bz2>

MD5 sum: 6eda8a97b86e0a6f59dabbf25202aa6f

Asennus riippuu: Binutils, Coreutils, GCC, Glibc, Make ja Sed

Testit riippuvat: -

Täytyy olla asennettu ennen: -

Tar-1.29

Kuvaus: Paketti sisältää arkistointiohjelman.

Latauksen koko: 1,950 KB

Vaadittava levytila: 33 MB

Kotisivut: <http://www.gnu.org/software/tar/>

Lataa: <http://ftp.gnu.org/gnu/tar/tar-1.29.tar.xz>

MD5 sum: a1802fec550baaeecff6c381629653ef

Täytyy olla asennettu ennen: -

Asennus riippuu: Acl, Attr, Bash, Binutils, Bison, Coreutils, GCC, Gettext, Glibc, Grep, Inetutils, Make, Sed ja Texinfo

Testit riippuvat: Autoconf, Diffutils, Findutils, Gawk ja Gzip

Tcl-core-8.6.6

Kuvaus: Paketti sisältää tulkattavan ohjelmointikielen (Tool Command Language).

Latauksen koko: 5,731 KB

Vaadittava levytila: 40 MB

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, Diffutils, GCC, Glibc, Make ja Sed

Testit riippuvat: -

Kotisivut: <http://tcl.sourceforge.net/>

Lataa: <http://sourceforge.net/projects/tcl/files/Tcl/8.6.6/tcl-core8.6.6-src.tar.gz>

MD5 sum: 98ebf13bbd90257e006c219369dd5f67

Texinfo-6.3

Kuvaus: Paketti sisältää ohjelmia infosivujen lukemiseen, muokkaamiseen ja konvertointiin.

Latauksen koko: 4,364 KB

Vaadittava levytila: 99 MB

Kotisivut: <http://www.gnu.org/software/texinfo/>

Lataa: <http://ftp.gnu.org/gnu/texinfo/texinfo-6.3.tar.xz>

MD5 sum: 32baefe5c7080dfb512a4eac5ce67b2a

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, GCC, Gettext, Glibc, Grep, Make, Ncurses, Patch ja Sed

Testit riippuvat: -

Util-linux-2.29.1

Kuvaus: Paketti sisältää sekalaisia työkaluohjelmia.

Latauksen koko: 4,179 KB

Vaadittava levytila: 118 MB

Kotisivut: <http://freecode.com/projects/util-linux>

Lataa: <https://www.kernel.org/pub/linux/utils/util-linux/v2.29/util-linux-2.29.1.tar.xz>

MD5 sum: 0cbb6d16ab9c5736e5649ef1264bee6e

Täytyy olla asennettu ennen: -

Asennus riippuu: Bash, Binutils, Coreutils, Diffutils, Findutils, Gawk, GCC, Gettext, Glibc, Grep, Make, Ncurses, Sed, Eudev ja Zlib

Testit riippuvat: -

XML::Parser-2.44

Kuvaus: XML::Parser -moduuli on Perl-käyttäjäliittymä James Clarkin XML-jäsentäjään, Expatiin.

Latauksen koko: 232 KB

Vaadittava levytila: 2 MB

Kotisivut: <https://github.com/chorny/XML-Parser>

Lataa: <http://cpan.metacpan.org/authors/id/T/TO/TODDR/XML-Parser-2.44.tar.gz>

MD5 sum: af4813fe3952362451201ced6fbce379

Täytyy olla asennettu ennen: Intltool

Asennus riippuu: Bash, Binutils, Coreutils, Expat, GCC, Glibc, Make ja Perl

Testit riippuvat: Perl

Xz-5.2.3

Kuvaus: Paketti sisältää ohjelmia tiedostojen pakkaamiseen ja purkamiseen.

Latauksen koko: 1009 KB

Vaadittava levytila: 16 MB

Kotisivut: <http://tukaani.org/xz>

Lataa: <http://tukaani.org/xz/xz-5.2.3.tar.xz>

MD5 sum: 60fb79cab777e3f71ca43d298adacbd5

Täytyy olla asennettu ennen: Intltool

Asennus riippuu: Bash, Binutils, Coreutils, Expat, GCC, Glibc, Make ja Perl

Testit riippuvat: Perl

Zlib-1.2.11

Kuvaus: Paketti sisältää joidenkin ohjelmien käyttämiä pakkaus- ja purkamisrutiineja.

Latauksen koko: 457 KB

Vaadittava levytila: 4,5 MB

Kotisivut: <http://www.zlib.net/>

Lataa: <http://zlib.net/zlib-1.2.11.tar.xz>

MD5 sum: 85adef240c5f370b308da8c938951a68

Täytyy olla asennettu ennen: File, Kmod, Perl ja Util-linux

Asennus riippuu: Bash, Binutils, Coreutils, GCC, Glibc, Make ja Sed

Testit riippuvat: -

8 LIITTEET

Liite 1 - version-check.sh -skripti

```
cat > version-check.sh << "EOF"
#!/bin/bash
# Simple script to list version numbers of critical development tools
export LC_ALL=C
bash --version | head -n1 | cut -d" " -f2-4
MYSH=$(readlink -f /bin/sh)
echo "/bin/sh -> $MYSH"
echo $MYSH | grep -q bash || echo "ERROR: /bin/sh does not point to
bash"
unset MYSH

echo -n "Binutils: "; ld --version | head -n1 | cut -d" " -f3-
bison --version | head -n1

if [ -h /usr/bin/yacc ]; then
    echo "/usr/bin/yacc -> `readlink -f /usr/bin/yacc`";
elif [ -x /usr/bin/yacc ]; then
    echo yacc is `/usr/bin/yacc --version | head -n1`
else
    echo "yacc not found"
fi

bzip2 --version 2>&1 < /dev/null | head -n1 | cut -d" " -f1,6-
echo -n "Coreutils: "; chown --version | head -n1 | cut -d")" -f2
diff --version | head -n1
find --version | head -n1
gawk --version | head -n1

if [ -h /usr/bin/awk ]; then
    echo "/usr/bin/awk -> `readlink -f /usr/bin/awk`";
elif [ -x /usr/bin/awk ]; then
    echo awk is `/usr/bin/awk --version | head -n1`
else
    echo "awk not found"
fi
```

```
gcc --version | head -n1
g++ --version | head -n1
ldd --version | head -n1 | cut -d" " -f2- # glibc version
grep --version | head -n1
gzip --version | head -n1
cat /proc/version
m4 --version | head -n1
make --version | head -n1
patch --version | head -n1
echo Perl `perl -V:version`
sed --version | head -n1
tar --version | head -n1
makeinfo --version | head -n1
xz --version | head -n1

echo 'int main(){}' > dummy.c && g++ -o dummy dummy.c
if [ -x dummy ]
    then echo "g++ compilation OK";
    else echo "g++ compilation failed"; fi
rm -f dummy.c dummy
EOF

bash version-check.sh
```

Liite 2 - library-check.sh -skripti

```
cat > library-check.sh << "EOF"
#!/bin/bash
for lib in lib{gmp,mpfr,mpc}.la; do
    echo $lib: $(if find /usr/lib* -name $lib |
                    grep -q $lib;then :;else echo not;fi) found
done
unset lib
EOF

bash library-check.sh
```

Liite 3 - wget-list

<http://download.savannah.gnu.org/releases/acl/acl-2.2.52.src.tar.gz>
<http://download.savannah.gnu.org/releases/attr/attr-2.4.47.src.tar.gz>
<http://ftp.gnu.org/gnu/autoconf/autoconf-2.69.tar.xz>
<http://ftp.gnu.org/gnu/automake/automake-1.15.tar.xz>
<http://ftp.gnu.org/gnu/bash/bash-4.4.tar.gz>
<http://alpha.gnu.org/gnu/bc/bc-1.06.95.tar.bz2>
<http://ftp.gnu.org/gnu/binutils/binutils-2.27.tar.bz2>
<http://ftp.gnu.org/gnu/bison/bison-3.0.4.tar.xz>
<http://www.bzip.org/1.0.6/bzip2-1.0.6.tar.gz>
<https://github.com/libcheck/check/releases/download/0.11.0/check-0.11.0.tar.gz>
<http://ftp.gnu.org/gnu/coreutils/coreutils-8.26.tar.xz>
<http://dbus.freedesktop.org/releases/dbus/dbus-1.10.14.tar.gz>
<http://ftp.gnu.org/gnu/dejagnu/dejagnu-1.6.tar.gz>
<http://ftp.gnu.org/gnu/diffutils/diffutils-3.5.tar.xz>
<http://dev.gentoo.org/~blueness/eudev/eudev-3.2.1.tar.gz>
<http://downloads.sourceforge.net/project/e2fsprogs/e2fsprogs/v1.43.4/e2fsprogs-1.43.4.tar.gz>
<http://prdownloads.sourceforge.net/expat/expat-2.2.0.tar.bz2>
<http://prdownloads.sourceforge.net/expect/expect5.45.tar.gz>
<ftp://ftp.astron.com/pub/file/file-5.30.tar.gz>
<http://ftp.gnu.org/gnu/findutils/findutils-4.6.0.tar.gz>
<https://github.com/westes/flex/releases/download/v2.6.3/flex-2.6.3.tar.gz>
<http://ftp.gnu.org/gnu/gawk/gawk-4.1.4.tar.xz>
<http://ftp.gnu.org/gnu/gcc/gcc-6.3.0/gcc-6.3.0.tar.bz2>
<http://ftp.gnu.org/gnu/gdbm/gdbm-1.12.tar.gz>
<http://ftp.gnu.org/gnu/gettext/gettext-0.19.8.1.tar.xz>
<http://ftp.gnu.org/gnu/glibc/glibc-2.25.tar.xz>
<http://ftp.gnu.org/gnu/gmp/gmp-6.1.2.tar.xz>
<http://ftp.gnu.org/gnu/gperf/gperf-3.0.4.tar.gz>
<http://ftp.gnu.org/gnu/grep/grep-3.0.tar.xz>
<http://ftp.gnu.org/gnu/groff/groff-1.22.3.tar.gz>
<http://alpha.gnu.org/gnu/grub/grub-2.02~beta3.tar.xz>
<http://ftp.gnu.org/gnu/gzip/gzip-1.8.tar.xz>
<http://anduin.linuxfromscratch.org/LFS/iana-etc-2.30.tar.bz2>
<http://ftp.gnu.org/gnu/inetutils/inetutils-1.9.4.tar.xz>
<http://launchpad.net/intltool/trunk/0.51.0/+download/intltool-0.51.0.tar.gz>
<https://www.kernel.org/pub/linux/utils/net/iproute2/iproute2-4.9.0.tar.xz>
<https://www.kernel.org/pub/linux/utils/kbd/kbd-2.0.4.tar.xz>
<https://www.kernel.org/pub/linux/utils/kernel/kmod/kmod-23.tar.xz>
<http://www.greenwoodsoftware.com/less/less-481.tar.gz>
<http://www.linuxfromscratch.org/lfs/downloads/8.0/lfs-bootscripts-20150222.tar.bz2>
<https://www.kernel.org/pub/linux/libs/security/linux-privs/libcap2/libcap-2.25.tar.xz>
<http://download.savannah.gnu.org/releases/libpipeline/libpipeline-1.4.1.tar.gz>
<http://ftp.gnu.org/gnu/libtool/libtool-2.4.6.tar.xz>
<https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.9.9.tar.xz>
<http://ftp.gnu.org/gnu/m4/m4-1.4.18.tar.xz>
<http://ftp.gnu.org/gnu/make/make-4.2.1.tar.bz2>
<http://download.savannah.gnu.org/releases/man-db/man-db-2.7.6.1.tar.xz>
<https://www.kernel.org/pub/linux/docs/man-pages/man-pages-4.09.tar.xz>
<http://www.multiprecision.org/mpc/download/mpc-1.0.3.tar.gz>
<http://www.mpfr.org/mpfr-3.1.5/mpfr-3.1.5.tar.xz>

<http://ftp.gnu.org/gnu/ncurses/ncurses-6.0.tar.gz>
<http://ftp.gnu.org/gnu/patch/patch-2.7.5.tar.xz>
<http://www.cpan.org/src/5.0/perl-5.24.1.tar.bz2>
<https://pkg-config.freedesktop.org/releases/pkg-config-0.29.1.tar.gz>
<http://sourceforge.net/projects/procps-ng/files/Production/procps-ng-3.3.12.tar.xz>
<http://downloads.sourceforge.net/project/psmisc/psmisc/psmisc-22.21.tar.gz>
<http://ftp.gnu.org/gnu/readline/readline-7.0.tar.gz>
<http://ftp.gnu.org/gnu/sed/sed-4.4.tar.xz>
<https://github.com/shadow-maint/shadow/releases/download/4.4/shadow-4.4.tar.xz>
<http://www.infodrom.org/projects/sysklogd/download/sysklogd-1.5.1.tar.gz>
<http://andu.in.linuxfromscratch.org/sources/other/systemd/systemd-232.tar.xz>
<http://download.savannah.gnu.org/releases/sysvinit/sysvinit-2.88dsf.tar.bz2>
<http://ftp.gnu.org/gnu/tar/tar-1.29.tar.xz>
<http://sourceforge.net/projects/tcl/files/Tcl/8.6.6/tcl-core8.6.6-src.tar.gz>
<http://ftp.gnu.org/gnu/texinfo/texinfo-6.3.tar.xz>
<http://www.iana.org/time-zones/repository/releases/tzdata2016j.tar.gz>
<http://andu.in.linuxfromscratch.org/LFS/udev-lfs-20140408.tar.bz2>
<https://www.kernel.org/pub/linux/utils/util-linux/v2.29/util-linux-2.29.1.tar.xz>
<ftp://ftp.vim.org/pub/vim/unix/vim-8.0.069.tar.bz2>
<http://cpan.metacpan.org/authors/id/T/TO/TODDR/XML-Parser-2.44.tar.gz>
<http://tukaani.org/xz/xz-5.2.3.tar.xz>
<http://zlib.net/zlib-1.2.11.tar.xz>
http://www.linuxfromscratch.org/patches/lfs/8.0/bash-4.4-upstream_fixes-1.patch
http://www.linuxfromscratch.org/patches/lfs/8.0/bc-1.06.95-memory_leak-1.patch
http://www.linuxfromscratch.org/patches/lfs/8.0/bzip2-1.0.6-install_docs-1.patch
<http://www.linuxfromscratch.org/patches/lfs/8.0/coreutils-8.26-i18n-1.patch>
<http://www.linuxfromscratch.org/patches/lfs/8.0/glibc-2.25-fhs-1.patch>
<http://www.linuxfromscratch.org/patches/lfs/8.0/kbd-2.0.4-backspace-1.patch>
<http://www.linuxfromscratch.org/patches/lfs/8.0/sysvinit-2.88dsf-consolidated-1.patch>

Liite 4 - md5sums

```
a61415312426e9c2212bd7dc7929abda
84f58dec00b60f2dc8fd1c9709291cc7
50f97f4159805e374639a73e2636f22e
9a1ddb0e053474d9d1105cfe39b0c48d
148888a7c95ac23705559b6f477dfe25
5126a721b73f97d715bb72c13c889035
2869c9bf3e60ee97c74ac2a6bf4e9d68
c342201de104cc9ce0a21e0ad10d4021
00b516f4704d4a7cb50ald97e6e8e15b
9b90522b31f5628c2e0f55dda348e558
d5aa2072f662d4118b9f4c63b94601a6
3f7b013ce8f641cd4c897acda0ef3467
1fdc2eb0d592c4f89d82d24dfdf02f0b
569354697ff1cfc9a9de3781361015fa
0bd1c74f357f6e9ae2ab6fa6229b9aea
2f47841c829facb346eb6e3fab5212e2
44e1a4f4c877e9ddc5a542dfa7ecc92b
f35eaf05489ae566eafc4d26bb1dcd90
9936aa8009438ce185bea2694a997fc1
a5f65570cd9107ec8a8ec88f17b31bb1
4e7dbc81163e60fd4f0b52496e7542c9
677a7623c7ef6ab99881bc4e048debb6
9ce96ff4c99e74295ea19040931c8fb9
df3f5690eaa30fd228537b00cb7b7590
1496c3bf41adf9db0ebd0af01f202eed
f58fa8001d60c4c77595fbbb62b63c1d
c1f1db32fb6598d6a93e6e88796a8632
fa07c1616adeb9c3262be5177d10ad4a
cc825fa64bc7306a885f2fb2268d3ec5
ab399fc6f74a97d66ff77f04b743149c
f7caabb65cddc1a4165b398009bd05b9
3ba3afb1d1b261383d247f46cb135ee8
87fef1fa3f603aef11c41dcc097af75e
12e517cac2b57a0121cda351570f1e63
44a8371a4b2c40e48e4c9f98cbd41391
c1635a5a83b63aca7f97a3eab39ebaa6
3cf469f40ec2ed51f56ba45ea03793e7
50ef46065c65257141a7340123527767
6666b839e5d46c2ad33fc8aa2ceb5f77
e54590ec68d6c1239f67b5b44e92022c
1bfb9b923f2c1339b4d2ce1807064aa5
de43a1a9e3alad97467c9a413bcd8f896
730bb15d96fffe47e148d1e09235af82
15b012617e7c44c0ed482721629577ac
2948d49d0ed7265f60f83aa4a9ac9268
91c721409bbf823d8f62bee3a1fe8ae3
d6a1d5f8ddea3abd2cc3e98f58352d26
c4ac246cf9795a4491e7766002cd528f
ee13d052e1ead260d7c28071f46eefb1
e3da7940431633fb65a01b91d3b7a27a
178ee0e8fa544dbc76d99cf041e2c9f0
f739a28cae4e0ca291f82d1d41ef107d
957e42e8b193490b2111252e4a2b443c
935c0fd6eb208288262b385fa656f1bf
205b03a87fc83dab653b628c59b9fc91
e0c583d4c380059abd818cd540fe6938
c06f8c2571b44899e60662f9ad259dd6
281604885d5a87f0359244f6f84037cb
acl-2.2.52.src.tar.gz
attr-2.4.47.src.tar.gz
autoconf-2.69.tar.xz
automake-1.15.tar.xz
bash-4.4.tar.gz
bc-1.06.95.tar.bz2
binutils-2.27.tar.bz2
bison-3.0.4.tar.xz
bzip2-1.0.6.tar.gz
check-0.11.0.tar.gz
coreutils-8.26.tar.xz
dbus-1.10.14.tar.gz
dejagnu-1.6.tar.gz
diffutils-3.5.tar.xz
e2fsprogs-1.43.4.tar.gz
expat-2.2.0.tar.bz2
expect5.45.tar.gz
file-5.30.tar.gz
findutils-4.6.0.tar.gz
flex-2.6.3.tar.gz
gawk-4.1.4.tar.xz
gcc-6.3.0.tar.bz2
gdbm-1.12.tar.gz
gettext-0.19.8.1.tar.xz
glibc-2.25.tar.xz
gmp-6.1.2.tar.xz
gperf-3.0.4.tar.gz
grep-3.0.tar.xz
groff-1.22.3.tar.gz
grub-2.02~beta3.tar.xz
gzip-1.8.tar.xz
iana-etc-2.30.tar.bz2
inetutils-1.9.4.tar.xz
intltool-0.51.0.tar.gz
iproute2-4.9.0.tar.xz
kbd-2.0.4.tar.xz
kmod-23.tar.xz
less-481.tar.gz
libcap-2.25.tar.xz
libpipeline-1.4.1.tar.gz
libtool-2.4.6.tar.xz
linux-4.9.9.tar.xz
m4-1.4.18.tar.xz
make-4.2.1.tar.bz2
man-db-2.7.6.1.tar.xz
man-pages-4.09.tar.xz
mpc-1.0.3.tar.gz
mpfr-3.1.5.tar.xz
ncurses-6.0.tar.gz
patch-2.7.5.tar.xz
perl-5.24.1.tar.bz2
pkg-config-0.29.1.tar.gz
procps-ng-3.3.12.tar.xz
psmisc-22.21.tar.gz
readline-7.0.tar.gz
sed-4.4.tar.xz
shadow-4.4.tar.xz
systemd-232.tar.xz
```

a1802fec550baaeecff6c381629653ef	tar-1.29.tar.xz
98ebf13bbd90257e006c219369dd5f67	tcl-core8.6.6-src.tar.gz
32baefe5c7080dfb512a4eac5ce67b2a	texinfo-6.3.tar.xz
db361d005ac8b30a2d18c5ca38d3e8ab	tzdata2016j.tar.gz
0cbb6d16ab9c5736e5649ef1264bee6e	util-linux-2.29.1.tar.xz
457543a7754b0d3c1c0aa4d4c3bb4070	vim-8.0.069.tar.bz2
af4813fe3952362451201ced6fbce379	XML-Parser-2.44.tar.gz
60fb79cab777e3f71ca43d298adacbd5	xz-5.2.3.tar.xz
85adef240c5f370b308da8c938951a68	zlib-1.2.11.tar.xz
e3d5bf23a4e5628680893d46e6ff286e	bash-4.4-upstream_fixes-1.patch
877e81fba316fe487ec23501059d54b8	bc-1.06.95-memory_leak-1.patch
6a5ac7e89b791aae556de0f745916f7f	bzip2-1.0.6-install_docs-1.patch
e1f87b10b23d66344e5e99e7fabfa7a2	coreutils-8.26-il8n-1.patch
9a5997c3452909b1769918c759eff8a2	glibc-2.25-fhs-1.patch
f75cca16a38da6caa7d52151f7136895	kbd-2.0.4-backspace-1.patch

Liite 5 - LFS:n hakemistorakenne

\$LFS = /mnt/lfs

/sources = Pakettien ja korjausten lähdetiedostot

/dev = Laitetiedostot (CD-ROM, USB jne.)

/sys = Tiedot laitteista, ajureista ja tietyistä kernelin ominaisuuksista

/boot = Sisältää kaiken, mitä tarvitaan järjestelmän käynnistämiseksi

/opt = Varattu add-on -ohjelmapakettien asentamiseen.

/home = Käyttäjän kotihakemisto; sisältää käyttäjäkohtaiset konfiguraatietiedostot

/tools = Pakettien käännöskohde

/tools/lib

/proc = Prosessi- ja järjestelmätiedot

/bin = Sisältää komentoja, joita sekä admin että käyttäjän voivat käyttää, mutta joita tarvitaan, kun muita tiedostojärjestelmiä ei ole liitetty

/sbin = Työkalut, joita käytetään järjestelmänhallinnassa; binäärit, joita tarvitaan käynnistyksessä, palauttamisessa ja järjestelmän korjauksessa

/srv = Sisältää paikallisen palvelinohjelmiston (esimerkiksi HTTP- tai FTP-palvelimen) tarjoamaa sivustokohtaista dataa

/run = Run-time data; järjestelmäinfo viimeisimmästä käynnistyksestä lähtien; tyhjenee aina käynnistyksen alussa; mm. PID-tiedostot (Process identifier)

/etc = Konfiguraatietiedostot, joilla kontrolloidaan ohjelmien toimintaa; ei lainkaan suoritettavia binääritiedostoja

/lib = Sisältää jaetut kirjaston kuvat (library image), joita tarvitaan järjestelmän käynnistämiseen ja komentojen ajamiseen root-tiedostojärjestelmässä; binäärit /bin ja /sbin -hakemistoissa tarvitsevat näitä

/mnt = On olemassa, jotta admin voi tarvittaessa liittää tilapäisen tiedostojärjestelmän

/media = Sisältää alahakemistot, joita käytetään liitospisteinä ulkoisille media-laitteille, kuten korpuille ja CD-ROMeille.

/media/floppy

/media/cdrom

/var = Muuttuvat tiedostot, lokitiedostot yms.

Liite 6 - Joitakin Bash:n ympäristömuuttujia

`BASHOPTS` = Kaksoispisteillä erotettu lista käytössä olevia shellin asetuksia.

`EDITOR` = Oletusarvoinen tekstieditori.

`HOME` = Kirjautuneen käyttäjän kotihakemisto.

`IFS` = Internal-field separator; Lista merkkejä, jotka erottavat kenttiä. Oletuksena välilyönti, tab ja enter.

`MAIL` = Mikäli `MAIL` muuttujan arvo on mail-tiedoston nimi ja `MAILPATH` muuttujaa ei ole asetettu, bash ilmoittaa käyttäjälle sähköpostin saapumisesta asetettuun tiedostoon.

`MAILCHECK` = Ilmoittaa sekunteina, kuinka usein [shellin](#) pitäisi tarkistaa sähköposti `MAILPATH` tai `MAIL` -muuttujissa määritellyissä tiedostoissa.

`MAILPATH` = Kaksoispisteillä erotettu lista tiedostonimiä, jotka shell tietyin välein tarkistaa uuden sähköpostin varalta.

`PATH` = Kaksoispisteillä erotettu lista hakemistoja, joista shell etsii komentoja.

`PS1` = Ensisijainen kehote (prompt), oletuksena \$-merkki.

`PS2` = Toissijainen kehote, oletuksena >-merkki.

`PWD` = Nykyinen työhakemisto. Asetetaan `cd`-komennolla.

`USER` = Kertoo sen hetkisen kirjautuneen käyttäjän käyttäjätunnuksen.

Liite 7 - /etc/rc.d/init.d/rc

```
#!/bin/bash

#####
#
# Begin rc
#
# Description : Main Run Level Control Script
#
# Authors      : Gerard Beekmans - gerard@linuxfromscratch.org
#               : DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
#####

. /lib/lsb/init-functions

print_error_msg()
{
    log_failure_msg
    # $i is set when called
    MSG="FAILURE:\n\nYou should not be reading this error message.\n\n"
    MSG="${MSG}It means that an unforeseen error took place in\n"
    MSG="${MSG}${i},\n"
    MSG="${MSG}which exited with a return value of ${error_value}.\n"

    MSG="${MSG}If you're able to track this error down to a bug in one
of\n"
    MSG="${MSG}the files provided by the ${DISTRO_MINI} book,\n"
    MSG="${MSG}please be so kind to inform us at ${DISTRO_CONTACT}.\n"
    log_failure_msg "${MSG}"

    log_info_msg "Press Enter to continue..."
    wait_for_user
}

check_script_status()
{
```

```

# $i is set when called
if [ ! -f ${i} ]; then
    log_warning_msg "${i} is not a valid symlink."
    continue
fi

if [ ! -x ${i} ]; then
    log_warning_msg "${i} is not executable, skipping."
    continue
fi
}

run()
{
    if [ -z $interactive ]; then
        ${1} ${2}
        return $?
    fi

    while true; do
        read -p "Run ${1} ${2} (Yes/no/continue)? " -n 1 runit
        echo

        case ${runit} in
            c | C)
                interactive=""
                ${i} ${2}
                ret=${?}
                break;
                ;;

            n | N)
                return 0
                ;;

            y | Y)
                ${i} ${2}
                ret=${?}
                break
                ;;

            esac

```

```

done

return $ret
}

# Read any local settings/overrides
[ -r /etc/sysconfig/rc.site ] && source /etc/sysconfig/rc.site

DISTRO=${DISTRO:-"Linux From Scratch"}
DISTRO_CONTACT=${DISTRO_CONTACT:-"lfs-dev@linuxfromscratch.org (Regis-
tration required)"}
DISTRO_MINI=${DISTRO_MINI:-"LFS"}
IPROMPT=${IPROMPT:-"no"}

# These 3 signals will not cause our script to exit
trap "" INT QUIT TSTP

[ "${1}" != "" ] && runlevel=${1}

if [ "${runlevel}" == "" ]; then
    echo "Usage: ${0} <runlevel>" >&2
    exit 1
fi

previous=${PREVLEVEL}
[ "${previous}" == "" ] && previous=N

if [ ! -d /etc/rc.d/rc${runlevel}.d ]; then
    log_info_msg "/etc/rc.d/rc${runlevel}.d does not exist.\n"
    exit 1
fi

if [ "$runlevel" == "6" -o "$runlevel" == "0" ]; then IPROMPT="no"; fi

# Note: In ${LOGLEVEL:-7}, it is ':' 'dash' '7', not minus 7
if [ "$runlevel" == "S" ]; then
    [ -r /etc/sysconfig/console ] && source /etc/sysconfig/console
    dmesg -n "${LOGLEVEL:-7}"
fi

if [ "${IPROMPT}" == "yes" -a "$runlevel" == "S" ]; then

```

```

# The total length of the distro welcome string, without escape
codes
wlen=${wlen:-$(echo "Welcome to ${DISTRO}" | wc -c )}
welcome_message=${welcome_message:-"Welcome to ${INFO}${DIS-
TRO}${NORMAL}"}

# The total length of the interactive string, without escape codes
ilen=${ilen:-$(echo "Press 'I' to enter interactive startup" | wc -c
)}
i_message=${i_message:-"Press '${FAILURE}I${NORMAL}' to enter inter-
active startup"}

# dcol and icol are spaces before the message to center the message
# on screen. itime is the amount of wait time for the user to press
a key
wcol=$(( ( ${COLUMNS} - ${wlen} ) / 2 ))
icol=$(( ( ${COLUMNS} - ${ilen} ) / 2 ))
itime=${itime:-"3"}

echo -e "\n\n"
echo -e "\\033[${wcol}G${welcome_message}
echo -e "\\033[${icol}G${i_message}${NORMAL}
echo ""
read -t "${itime}" -n 1 interactive 2>&1 > /dev/null
fi

# Make lower case
[ "${interactive}" == "I" ] && interactive="i"
[ "${interactive}" != "i" ] && interactive=""

# Read the state file if it exists from runlevel S
[ -r /var/run/interactive ] && source /var/run/interactive

# Attempt to stop all services started by the previous runlevel,
# and killed in this runlevel
if [ "${previous}" != "N" ]; then
    for i in $(ls -v /etc/rc.d/rc${runlevel}.d/K* 2> /dev/null)
    do
        check_script_status

```

```

suffix=${i#/etc/rc.d/rc$runlevel.d/K[0-9][0-9]}
prev_start=/etc/rc.d/rc$previous.d/S[0-9][0-9]$suffix
sysinit_start=/etc/rc.d/rcS.d/S[0-9][0-9]$suffix

if [ "${runlevel}" != "0" -a "${runlevel}" != "6" ]; then
    if [ ! -f ${prev_start} -a ! -f ${sysinit_start} ]; then
        MSG="WARNING:\n\n${i} can't be "
        MSG="${MSG}executed because it was not "
        MSG="${MSG}not started in the previous "
        MSG="${MSG}runlevel (${previous})."
        log_warning_msg "$MSG"
        continue
    fi
fi

run ${i} stop
error_value=${?}

if [ "${error_value}" != "0" ]; then print_error_msg;
fi
done
fi

if [ "${previous}" == "N" ]; then export IN_BOOT=1; fi

if [ "$runlevel" == "6" -a -n "${FASTBOOT}" ]; then
    touch /fastboot
fi

# Start all functions in this runlevel
for i in $( ls -v /etc/rc.d/rc${runlevel}.d/S* 2> /dev/null )
do
    if [ "${previous}" != "N" ]; then
        suffix=${i#/etc/rc.d/rc$runlevel.d/S[0-9][0-9]}
        stop=/etc/rc.d/rc$runlevel.d/K[0-9][0-9]$suffix
        prev_start=/etc/rc.d/rc$previous.d/S[0-9][0-9]$suffix

        [ -f ${prev_start} -a ! -f ${stop} ] && continue
    fi

```

```

check_script_status

case ${runlevel} in
    0|6)
        run ${i} stop
        ;;
    *)
        run ${i} start
        ;;
esac

error_value=${?}

if [ "${error_value}" != "0" ]; then print_error_msg; fi
done

# Store interactive variable on switch from runlevel S and remove if
not
if [ "${runlevel}" == "S" -a "${interactive}" == "i" ]; then
    echo "interactive=\"i\"" > /var/run/interactive
else
    rm -f /var/run/interactive 2> /dev/null
fi

# Copy the boot log on initial boot only
if [ "${previous}" == "N" -a "${runlevel}" != "S" ]; then
    cat $BOOTLOG >> /var/log/boot.log

    # Mark the end of boot
    echo "-----" >> /var/log/boot.log

    # Remove the temporary file
    rm -f $BOOTLOG 2> /dev/null
fi

# End rc

```

Liite 8 - /lib/lsb/init-functions

```
#!/bin/sh

#####

#
#
# Begin /lib/lsb/init-funtions
#
# Description : Run Level Control Functions
#
# Authors      : Gerard Beekmans - gerard@linuxfromscratch.org
#               : DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
# Notes        : With code based on Matthias Benkmann's simpleinit-msb
#               http://winterdrache.de/linux/newboot/index.html
#
#               The file should be located in /lib/lsb
#
#####

## Environmental setup
# Setup default values for environment
umask 022
export PATH="/bin:/usr/bin:/sbin:/usr/sbin"

## Set color commands, used via echo
# Please consult `man console_codes` for more information
# under the "ECMA-48 Set Graphics Rendition" section
#
# Warning: when switching from a 8bit to a 9bit font,
# the linux console will reinterpret the bold (1;) to
# the top 256 glyphs of the 9bit font. This does
# not affect framebuffer consoles

NORMAL="\033[0;39m"      # Standard console grey
SUCCESS="\033[1;32m"      # Success is green
WARNING="\033[1;33m"      # Warnings are yellow
```

```

FAILURE="\033[1;31m"      # Failures are red
INFO="\033[1;36m"         # Information is light cyan
BRACKET="\033[1;34m"      # Brackets are blue

# Use a colored prefix
BMPREFIX=""
SUCCESS_PREFIX="\${SUCCESS} * \${NORMAL}"
FAILURE_PREFIX="\${FAILURE}*****\${NORMAL}"
WARNING_PREFIX="\${WARNING} *** \${NORMAL}"
SKIP_PREFIX="\${INFO} S \${NORMAL}"

SUCCESS_SUFFIX="\${BRACKET}[\${SUCCESS} OK \${BRACKET}]\${NORMAL}"
FAILURE_SUFFIX="\${BRACKET}[\${FAILURE} FAIL \${BRACKET}]\${NORMAL}"
WARNING_SUFFIX="\${BRACKET}[\${WARNING} WARN \${BRACKET}]\${NORMAL}"
SKIP_SUFFIX="\${BRACKET}[\${INFO} SKIP \${BRACKET}]\${NORMAL}"

BOOTLOG=/run/bootlog
KILLDELAY=3

# Set any user specified environment variables e.g. HEA
LESS
[ -r /etc/sysconfig/rc.site ] && . /etc/sysconfig/rc.site

## Screen Dimensions
# Find current screen size
if [ -z "\${COLUMNS}" ]; then
    COLUMNS=$(stty size)
    COLUMNS=${COLUMNS##* }
fi

# When using remote connections, such as a serial port, stty size re-
turns 0
if [ "\${COLUMNS}" = "0" ]; then
    COLUMNS=80
fi

## Measurements for positioning result messages
COL=$((\${COLUMNS} - 8))
WCOL=$((\${COL} - 2))

## Set Cursor Position Commands, used via echo

```

```

SET_COL="\033[${COL}G"      # at the $COL char
SET_WCOL="\033[${WCOL}G"    # at the $WCOL char
CURS_UP="\033[1A\033[0G"    # Up one line, at the 0'th char
CURS_ZERO="\033[0G

#####
#####
# start_daemon()
#
# Usage: start_daemon [-f] [-n nicelevel] [-p pidfile] pathname
# [args...]          #
#
# Purpose: This runs the specified program as a daemon
#
#
#
# Inputs: -f: (force) run the program even if it is already running.
#
#          -n nicelevel: specify a nice level. See 'man nice(1)'.
#
#          -p pidfile: use the specified file to determine PIDs.
#
#          pathname: the complete path to the specified program
#
#          args: additional arguments passed to the program (pathname)
#
#
# Return values (as defined by LSB exit codes):
#
#          0 - program is running or service is OK
#
#          1 - generic or unspecified error
#
#          2 - invalid or excessive argument(s)
#
#          5 - program is not installed
#
#####

```

```
#####
start_daemon()
{
    local force=""
    local nice="0"
    local pidfile=""
    local pidlist=""
    local retval=""

    # Process arguments
    while true
    do
        case "${1}" in

            -f)
                force="1"
                shift 1
                ;;

            -n)
                nice="${2}"
                shift 2
                ;;

            -p)
                pidfile="${2}"
                shift 2
                ;;

            -*)
                return 2
                ;;

            *)
                program="${1}"
                break
                ;;

        esac
    done

    # Check for a valid program
```

```

if [ ! -e "${program}" ]; then return 5; fi

# Execute
if [ -z "${force}" ]; then
    if [ -z "${pidfile}" ]; then
        # Determine the pid by discovery
        pidlist=`pidofproc "${1}"`
        retval="${?}"
    else
        # The PID file contains the needed PIDs
        # Note that by LSB requirement, the path must be given to
pidofproc,
        # however, it is not used by the current implementation or
standard.
        pidlist=`pidofproc -p "${pidfile}" "${1}"`
        retval="${?}"
    fi

    # Return a value ONLY
    # It is the init script's (or distribution's functions) respon-
sibility
    # to log messages!
    case "${retval}" in

        0)
            # Program is already running correctly, this is a
            # successful start.
            return 0
            ;;

        1)
            # Program is not running, but an invalid pid file ex-
ists
            # remove the pid file and continue
            rm -f "${pidfile}"
            ;;

        3)
            # Program is not running and no pidfile exists
            # do nothing here, let start_daemon continue.
            ;;
    esac

```

```

        *)
            # Others as returned by status values shall not be in-
interpreted
            # and returned as an unspecified error.
            return 1
            ;;
        esac
    fi

    # Do the start!
    nice -n "${nice}" "${@}"
}

#####
#####
# killproc()
#
# Usage: killproc [-p pidfile] pathname [signal]
#
#
#
# Purpose: Send control signals to running processes
#
#
#
# Inputs: -p pidfile, uses the specified pidfile
#
#         pathname, pathname to the specified program
#
#
#         signal, send this signal to pathname
#
#
#
# Return values (as defined by LSB exit codes):
#
#         0 - program (pathname) has stopped/is already stopped or a
#
#         running program has been sent specified signal and stopped
#

```

```

#             successfully
#
#             1 - generic or unspecified error
#
#             2 - invalid or excessive argument(s)
#
#             5 - program is not installed
#
#             7 - program is not running and a signal was supplied
#
#####
#####
killproc()
{
    local pidfile
    local program
    local prefix
    local progname
    local signal="-TERM"
    local fallback="-KILL"
    local nosig
    local pidlist
    local retval
    local pid
    local delay="30"
    local piddead
    local dtime

    # Process arguments
    while true; do
        case "${1}" in
            -p)
                pidfile="${2}"
                shift 2
                ;;

            *)
                program="${1}"
                if [ -n "${2}" ]; then
                    signal="${2}"
                    fallback=""
                fi
            ;;
        esac
    done
}

```

```

        else
            nosig=1
        fi

        # Error on additional arguments
        if [ -n "${3}" ]; then
            return 2
        else
            break
        fi
    ;;
esac
done

# Check for a valid program
if [ ! -e "${program}" ]; then return 5; fi

# Check for a valid signal
check_signal "${signal}"
if [ "${?}" -ne "0" ]; then return 2; fi

# Get a list of pids
if [ -z "${pidfile}" ]; then
    # determine the pid by discovery
    pidlist=`pidofproc "${1}"`
    retval="${?}"
else
    # The PID file contains the needed PIDs
    # Note that by LSB requirement, the path must be given to
pidofproc,
    # however, it is not used by the current implementation or
standard.
    pidlist=`pidofproc -p "${pidfile}" "${1}"`
    retval="${?}"
fi

# Return a value ONLY
# It is the init script's (or distribution's functions) responsi-
bility
# to log messages!

```

```

case "${retval}" in

    0)
        # Program is running correctly
        # Do nothing here, let killproc continue.
        ;;

    1)
        # Program is not running, but an invalid pid file exists
        # Remove the pid file.
        rm -f "${pidfile}"

        # This is only a success if no signal was passed.
        if [ -n "${nosig}" ]; then
            return 0
        else
            return 7
        fi
        ;;

    3)
        # Program is not running and no pidfile exists
        # This is only a success if no signal was passed.
        if [ -n "${nosig}" ]; then
            return 0
        else
            return 7
        fi
        ;;

    *)
        # Others as returned by status values shall not be interpreted
        # and returned as an unspecified error.
        return 1
        ;;

esac

# Perform different actions for exit signals and control signals
check_sig_type "${signal}"

```

```

        if [ "${?}" -eq "0" ]; then # Signal is used to terminate the pro-
gram

        # Account for empty pidlist (pid file still exists and no
        # signal was given)
        if [ "${pidlist}" != "" ]; then

            # Kill the list of pids
            for pid in ${pidlist}; do

                kill -0 "${pid}" 2> /dev/null

                if [ "${?}" -ne "0" ]; then
                    # Process is dead, continue to next and assume all
is well

                    continue
                else
                    kill "${signal}" "${pid}" 2> /dev/null

                    # Wait up to ${delay}/10 seconds to for "${pid}" to
                    # terminate in 10ths of a second

                    while [ "${delay}" -ne "0" ]; do
                        kill -0 "${pid}" 2> /dev/null || piddead="1"
                        if [ "${piddead}" = "1" ]; then break; fi
                        sleep 0.1
                        delay=$(( ${delay} - 1 ))
                    done

                    # If a fallback is set, and program is still run-
ning, then

                    # use the fallback
                    if [ -n "${fallback}" -a "${piddead}" != "1" ];
then
                        kill "${fallback}" "${pid}" 2> /dev/null
                        sleep 1
                        # Check again, and fail if still running
                        kill -0 "${pid}" 2> /dev/null && return 1
                    fi
                fi
            done
        done

```

```

fi

# Check for and remove stale PID files.
if [ -z "${pidfile}" ]; then
    # Find the basename of $program
    prefix=`echo "${program}" | sed 's/[^/]*$//'`
    proname=`echo "${program}" | sed "s@${prefix}@@"`

    if [ -e "/var/run/${proname}.pid" ]; then
        rm -f "/var/run/${proname}.pid" 2> /dev/null
    fi
else
    if [ -e "${pidfile}" ]; then rm -f "${pidfile}" 2>
/dev/null; fi
fi

# For signals that do not expect a program to exit, simply
# let kill do its job, and evaluate kill's return for value

else # check_sig_type - signal is not used to terminate program
    for pid in ${pidlist}; do
        kill "${signal}" "${pid}"
        if [ "${?}" -ne "0" ]; then return 1; fi
    done
fi
}

#####
#####
# pidofproc()
#
# Usage: pidofproc [-p pidfile] pathname
#
#
# Purpose: This function returns one or more pid(s) for a particular
daemon    #
#
#
# Inputs: -p pidfile, use the specified pidfile instead of pidof

```

```

#
#      pathname, path to the specified program
#
#
#
# Return values (as defined by LSB status codes):
#
#      0 - Success (PIDs to stdout)
#
#      1 - Program is dead, PID file still exists (remaining PIDs out-
put)      #
#      3 - Program is not running (no output)
#

#####
#####
pidofproc()
{
    local pidfile
    local program
    local prefix
    local proname
    local pidlist
    local lpids
    local exitstatus="0"

    # Process arguments
    while true; do
        case "${1}" in

            -p)
                pidfile="${2}"
                shift 2
                ;;

            *)
                program="${1}"
                if [ -n "${2}" ]; then
                    # Too many arguments
                    # Since this is status, return unknown
                    return 4

```

```

        else
            break
        fi
    ;;
esac
done

# If a PID file is not specified, try and find one.
if [ -z "${pidfile}" ]; then
    # Get the program's basename
    prefix=`echo "${program}" | sed 's/[^/]*$//'`

    if [ -z "${prefix}" ]; then
        proname="${program}"
    else
        proname=`echo "${program}" | sed "s@${prefix}@@"`
    fi

    # If a PID file exists with that name, assume that is it.
    if [ -e "/var/run/${proname}.pid" ]; then
        pidfile="/var/run/${proname}.pid"
    fi
fi

# If a PID file is set and exists, use it.
if [ -n "${pidfile}" -a -e "${pidfile}" ]; then

    # Use the value in the first line of the pidfile
    pidlist=`/bin/head -n1 "${pidfile}"`
    # This can optionally be written as 'sed 1q' to replace 'head -
n1'

    # should LFS move /bin/head to /usr/bin/head
else
    # Use pidof
    pidlist=`pidof "${program}"`
fi

# Figure out if all listed PIDs are running.
for pid in ${pidlist}; do
    kill -0 ${pid} 2> /dev/null

```

```

        if [ "${?}" -eq "0" ]; then
            lpids="${lpids}${pid} "
        else
            exitstatus="1"
        fi
    done

    if [ -z "${lpids}" -a ! -f "${pidfile}" ]; then
        return 3
    else
        echo "${lpids}"
        return "${exitstatus}"
    fi
}

#####
#####
# statusproc()
#
# Usage: statusproc [-p pidfile] pathname
#
#
#
# Purpose: This function prints the status of a particular daemon to
stdout    #
#
#
# Inputs: -p pidfile, use the specified pidfile instead of pidof
#
#         pathname, path to the specified program
#
#
#
# Return values:
#
#         0 - Status printed
#
#         1 - Input error. The daemon to check was not specified.
#
#####
#####

```

```

statusproc()
{
    local pidfile
    local pidlist

    if [ "${#}" = "0" ]; then
        echo "Usage: statusproc [-p pidfile] {program}"
        exit 1
    fi

    # Process arguments
    while true; do
        case "${1}" in

            -p)
                pidfile="${2}"
                shift 2
                ;;

            *)
                if [ -n "${2}" ]; then
                    echo "Too many arguments"
                    return 1
                else
                    break
                fi
                ;;
        esac
    done

    if [ -n "${pidfile}" ]; then
        pidlist=`pidofproc -p "${pidfile}" $@`
    else
        pidlist=`pidofproc $@`
    fi

    # Trim trailing blanks
    pidlist=`echo "${pidlist}" | sed -r 's/ +$//'\`

    base="${1##*/}"

```

```

if [ -n "${pidlist}" ]; then
    /bin/echo -e "${INFO}${base} is running with Process" \
        "ID(s) ${pidlist}.${NORMAL}"
else
    if [ -n "${base}" -a -e "/var/run/${base}.pid" ]; then
        /bin/echo -e "${WARNING}${1} is not running but" \
            "/var/run/${base}.pid exists.${NORMAL}"
    else
        if [ -n "${pidfile}" -a -e "${pidfile}" ]; then
            /bin/echo -e "${WARNING}${1} is not running" \
                "but ${pidfile} exists.${NORMAL}"
        else
            /bin/echo -e "${INFO}${1} is not running.${NORMAL}"
        fi
    fi
fi
}

#####
#####
# timespec()
#
#
#
# Purpose: An internal utility function to format a timestamp
#
#         a boot log file.  Sets the STAMP variable.
#
#
#
# Return value: Not used
#
#####
#####
timespec()
{
    STAMP="$(echo `date +%b %d %T %:z` `hostname`)"
    return 0
}

#####

```

```
#####
# log_success_msg()
#
# Usage: log_success_msg ["message"]
#
#
#
# Purpose: Print a successful status message to the screen and
#
#         a boot log file.
#
#
#
#
# Inputs: $@ - Message
#
#
#
# Return values: Not used
#
#####
#####
log_success_msg()
{
    /bin/echo -n -e "${BMPREFIX}${@}"
    /bin/echo -e "${CURS_ZERO}${SUCCESS_PREFIX}${SET_COL}${SUCCESS_SUFFIX}"

    # Strip non-printable characters from log file
    logmessage=`echo "${@}" | sed 's/\\033[^\a-zA-Z]*.//g'`

    timespec
    /bin/echo -e "${STAMP} ${logmessage} OK" >> ${BOOTLOG}

    return 0
}

log_success_msg2()
{
    /bin/echo -n -e "${BMPREFIX}${@}"

```

```

/bin/echo -e "${CURS_ZERO}${SUCCESS_PREFIX}${SET_COL}${SUCCESS_SUFFIX}"

echo " OK" >> ${BOOTLOG}

return 0
}

#####
#####
# log_failure_msg()
#
# Usage: log_failure_msg ["message"]
#
#
#
# Purpose: Print a failure status message to the screen and
#
#           a boot log file.
#
#
#
# Inputs: $@ - Message
#
#
#
# Return values: Not used
#
#####
#####
log_failure_msg()
{
    /bin/echo -n -e "${BMPREFIX}${@}"
    /bin/echo -e "${CURS_ZERO}${FAILURE_PREFIX}${SET_COL}${FAILURE_SUFFIX}"

    # Strip non-printable characters from log file

    timespec
    logmessage=`echo "${@}" | sed 's/\\033[^\a-zA-Z]*.//g`
    /bin/echo -e "${STAMP} ${logmessage} FAIL" >> ${BOOTLOG}

```

```

        return 0
    }

log_failure_msg2()
{
    /bin/echo -n -e "${BMPREFIX}${@}"
    /bin/echo -e "${CURS_ZERO}${FAILURE_PREFIX}${SET_COL}${FAILURE_SUFFIX}"

    echo "FAIL" >> ${BOOTLOG}

    return 0
}

#####
#####
# log_warning_msg()
#
# Usage: log_warning_msg ["message"]
#
#
#
# Purpose: Print a warning status message to the screen and
#
#           a boot log file.
#
#
#
# Return values: Not used
#
#####
#####
log_warning_msg()
{
    /bin/echo -n -e "${BMPREFIX}${@}"
    /bin/echo -e "${CURS_ZERO}${WARNING_PREFIX}${SET_COL}${WARNING_SUFFIX}"

    # Strip non-printable characters from log file
    logmessage=`echo "${@}" | sed 's/\\033[^a-zA-Z]*.//g'`

```

```

timespec
/bin/echo -e "${STAMP} ${logmessage} WARN" >> ${BOOTLOG}

return 0
}

log_skip_msg()
{
    /bin/echo -n -e "${BMPREFIX}${@}"
    /bin/echo -e "${CURS_ZERO}${SKIP_PREFIX}${SET_COL}${SKIP_SUFFIX}"

    # Strip non-printable characters from log file
    logmessage=`echo "${@}" | sed 's/\\033[^\a-zA-Z]*.//g'`
    /bin/echo "SKIP" >> ${BOOTLOG}

    return 0
}

#####
#####
# log_info_msg()
#
# Usage: log_info_msg message
#
#
# Purpose: Print an information message to the screen and
#
#          a boot log file. Does not print a trailing newline character.
#
#
# Return values: Not used
#
#####
#####
log_info_msg()
{
    /bin/echo -n -e "${BMPREFIX}${@}"

    # Strip non-printable characters from log file

```

```

    logmessage=`echo "${@}" | sed 's/\\033[^\a-zA-Z]*\./g'`
    timespec
    /bin/echo -n -e "${STAMP} ${logmessage}" >> ${BOOTLOG}

    return 0
}

log_info_msg2()
{
    /bin/echo -n -e "${@}"

    # Strip non-printable characters from log file
    logmessage=`echo "${@}" | sed 's/\\033[^\a-zA-Z]*\./g'`
    /bin/echo -n -e "${logmessage}" >> ${BOOTLOG}

    return 0
}

#####
#####
# evaluate_retval()
#
# Usage: Evaluate a return value and print success or failyure as appropriate #
#
#
# Purpose: Convenience function to terminate an info message
#
#
#
# Return values: Not used
#
#####
#####
evaluate_retval()
{
    local error_value="${?}"

    if [ ${error_value} = 0 ]; then
        log_success_msg2
    else

```

```

        log_failure_msg2
    fi
}

#####
#####
# check_signal()
#
# Usage: check_signal [ -{signal} | {signal} ]
#
#
#
# Purpose: Check for a valid signal. This is not defined by any LSB
draft,    #
#         however, it is required to check the signals to determine if
the      #

#         signals chosen are invalid arguments to the other functions.
#
#
#
# Inputs: Accepts a single string value in the form or -{signal} or
{signal}  #
#
#
# Return values:
#
#         0 - Success (signal is valid
#
#         1 - Signal is not valid
#
#####
#####
check_signal()
{
    local valsig

    # Add error handling for invalid signals
    valsig="-ALRM -HUP -INT -KILL -PIPE -POLL -PROF -TERM -USR1 -USR2"
    valsig="${valsig} -VTALRM -STKFLT -PWR -WINCH -CHLD -URG -TSTP -
TTIN"

```

```

    valsig="${valsig} -TTOU -STOP -CONT -ABRT -FPE -ILL -QUIT -SEGV -
TRAP"

    valsig="${valsig} -SYS -EMT -BUS -XCPU -XFSZ -0 -1 -2 -3 -4 -5 -6 -
8 -9"

    valsig="${valsig} -11 -13 -14 -15"

    echo "${valsig}" | grep -- " ${1} " > /dev/null

    if [ "${?}" -eq "0" ]; then
        return 0
    else
        return 1
    fi
}

#####
#####
# check_sig_type()
#
# Usage: check_signal [ -{signal} | {signal} ]
#
#
#
# Purpose: Check if signal is a program termination signal or a control
signal #
#
# This is not defined by any LSB draft, however, it is re-
quired to #
#
# check the signals to determine if they are intended to end a
#
#
# program or simply to control it.
#
#
#
# Inputs: Accepts a single string value in the form or -{signal} or
{signal} #
#
#
# Return values:
#
#
# 0 - Signal is used for program termination
#

```

```

#      1 - Signal is used for program control
#
#####
#####
check_sig_type()
{
    local valsig

    # The list of termination signals (limited to generally used items)
    valsig="-ALRM -INT -KILL -TERM -PWR -STOP -ABRT -QUIT -2 -3 -6 -9 -
14 -15"

    echo "${valsig}" | grep -- " ${1} " > /dev/null

    if [ "${?}" -eq "0" ]; then
        return 0
    else
        return 1
    fi
}

#####
#####
# wait_for_user()
#
#
#
# Purpose: Wait for the user to respond if not a headless system
#
#
#
#####
#####
wait_for_user()
{
    # Wait for the user by default
    [ "${HEADLESS=0}" = "0" ] && read ENTER
    return 0
}

#####

```

```
#####
# is_true()
#
#
#
# Purpose: Utility to test if a variable is true | yes | 1
#
#
#
#####
#####
is_true()
{
    [ "$1" = "1" ] || [ "$1" = "yes" ] || [ "$1" = "true" ] || [ "$1" =
"y" ] ||
    [ "$1" = "t" ]
}

# End /lib/lsb/init-functions
```

Liite 9 - /etc/rc.d/init.d/mountvirtfs

```
#!/bin/sh

#####
#
# Begin mountvirtfs
#
# Description : Mount proc, sysfs, and run
#
# Authors      : Gerard Beekmans - gerard@linuxfromscratch.org
#                DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
#####

### BEGIN INIT INFO
# Provides:          mountvirtfs
# Required-Start:
# Should-Start:
# Required-Stop:
# Should-Stop:
# Default-Start:     S
# Default-Stop:
# Short-Description: Mounts /sys and /proc virtual (kernel) filesystems.
#
#                   Mounts /run (tmpfs) and /dev (devtmpfs).
# Description:       Mounts /sys and /proc virtual (kernel) filesystems.
#
#                   Mounts /run (tmpfs) and /dev (devtmpfs).
# X-LFS-Provided-By: LFS
### END INIT INFO

. /lib/lsb/init-functions

case "${1}" in
    start)
        # Make sure /run is available before logging any messages
        if ! mountpoint /run >/dev/null; then
```

```

        mount /run || failed=1
    fi

    mkdir -p /run/lock /run/shm
    chmod 1777 /run/shm /run/lock

    log_info_msg "Mounting virtual file systems: ${INFO}/run"

    if ! mountpoint /proc >/dev/null; then
        log_info_msg2 " ${INFO}/proc"
        mount -o nosuid,noexec,nodev /proc || failed=1
    fi

    if ! mountpoint /sys >/dev/null; then
        log_info_msg2 " ${INFO}/sys"
        mount -o nosuid,noexec,nodev /sys || failed=1
    fi

    if ! mountpoint /dev >/dev/null; then
        log_info_msg2 " ${INFO}/dev"
        mount -o mode=0755,nosuid /dev || failed=1
    fi

    ln -sf /run/shm /dev/shm

    (exit ${failed})
    evaluate_retval
    exit $failed
;;

*)
    echo "Usage: ${0} {start}"
    exit 1
;;

esac

# End mountvirtfs

```

Liite 10 - /etc/rc.d/init.d/modules

```
#!/bin/sh

#####

#

# Begin modules

#

# Description : Module auto-loading script

#

# Authors      : Zack Winkles
#                DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
#####

### BEGIN INIT INFO
# Provides:          modules
# Required-Start:    mountvirtfs sysctl
# Should-Start:
# Required-Stop:
# Should-Stop:
# Default-Start:     S
# Default-Stop:
# Short-Description: Loads required modules.
# Description:       Loads modules listed in /etc/sysconfig/modules.
# X-LFS-Provided-By: LFS
### END INIT INFO

# Assure that the kernel has module support.
[ -e /proc/modules ] || exit 0

. /lib/lsb/init-functions

case "${1}" in
    start)
        # Exit if there's no modules file or there are no
        # valid entries
        [ -r /etc/sysconfig/modules ] || exit 0
```

```

egrep -qv '^(|$|#)' /etc/sysconfig/modules || exit 0

log_info_msg "Loading modules:"

# Only try to load modules if the user has actually given us
# some modules to load.

while read module args; do

    # Ignore comments and blank lines.
    case "$module" in
        ""|"#"*) continue ;;
    esac

    # Attempt to load the module, passing any arguments provided.
    modprobe ${module} ${args} >/dev/null

    # Print the module name if successful, otherwise take note.
    if [ $? -eq 0 ]; then
        log_info_msg2 " ${module}"
    else
        failedmod="${failedmod} ${module}"
    fi
done < /etc/sysconfig/modules

# Print a message about successfully loaded modules on the cor-
rect line.
log_success_msg2

# Print a failure message with a list of any modules that
# may have failed to load.
if [ -n "${failedmod}" ]; then
    log_failure_msg "Failed to load modules:${failedmod}"
    exit 1
fi

;;

*)
echo "Usage: ${0} {start}"
exit 1

;;

```

```
esac
```

```
exit 0
```

```
# End modules
```

Liite 11 - /etc/rc.d/init.d/udev

```
#!/bin/sh

#####
#
# Begin udev
#
# Description : Udev cold-plugging script
#
# Authors      : Zack Winkles, Alexander E. Patrakov
#                DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
#####

### BEGIN INIT INFO
# Provides:          udev $time
# Required-Start:
# Should-Start:      modules
# Required-Stop:
# Should-Stop:
# Default-Start:     S
# Default-Stop:
# Short-Description: Populates /dev with device nodes.
# Description:        Mounts a tmpfs on /dev and starts the udevd
daemon.
#
#                Device nodes are created as defined by udev.
# X-LFS-Provided-By:  LFS
### END INIT INFO

. /lib/lsb/init-functions

case "${1}" in
    start)
        log_info_msg "Populating /dev with device nodes... "
        if ! grep -q '[[[:space:]]sysfs' /proc/mounts; then
            log_failure_msg2
            msg="FAILURE:\n\nUnable to create "
```

```

        msg="${msg}devices without a SysFS filesystem\n\n"
        msg="${msg}After you press Enter, this system "
        msg="${msg}will be halted and powered off.\n\n"
        log_info_msg "$msg"
        log_info_msg "Press Enter to continue..."
        wait_for_user
        /etc/rc.d/init.d/halt stop
    fi

    # Start the udev daemon to continually watch for, and act on,
    # uevents
    /sbin/udev --daemon

    # Now traverse /sys in order to "coldplug" devices that have
    # already been discovered
    /sbin/udevadm trigger --action=add      --type=subsystems
    /sbin/udevadm trigger --action=add      --type=devices
    /sbin/udevadm trigger --action=change --type=devices

    # Now wait for udevd to process the uevents we triggered
    if ! is_true "$OMIT_UDEV_SETTLE"; then
        /sbin/udevadm settle
    fi

    # If any LVM based partitions are on the system, ensure they
    # are activated so they can be used.
    if [ -x /sbin/vgchange ]; then /sbin/vgchange -a y >/dev/null; fi

    log_success_msg2
    ;;

*)
    echo "Usage ${0} {start}"
    exit 1
    ;;
esac

exit 0

# End udev

```

Liite 12 - /etc/rc.d/init.d/swap

```
#!/bin/sh

#####
#
# Begin swap
#
# Description : Swap Control Script
#
# Authors      : Gerard Beekmans - gerard@linuxfromscratch.org
#               DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
#####

### BEGIN INIT INFO
# Provides:          swap
# Required-Start:    udev
# Should-Start:      modules
# Required-Stop:     localnet
# Should-Stop:
# Default-Start:     S
# Default-Stop:      0 6
# Short-Description: Mounts and unmounts swap partitions.
# Description:       Mounts and unmounts swap partitions defined in
#                   /etc/fstab.
# X-LFS-Provided-By: LFS
### END INIT INFO

. /lib/lsb/init-functions

case "${1}" in
    start)
        log_info_msg "Activating all swap files/partitions..."
        swapon -a
        evaluate_retval
        ;;
```

```
stop)
    log_info_msg "Deactivating all swap files/partitions..."
    swapoff -a
    evaluate_retval
    ;;

restart)
    ${0} stop
    sleep 1
    ${0} start
    ;;

status)
    log_success_msg "Retrieving swap status."
    swapon -s
    ;;

*)
    echo "Usage: ${0} {start|stop|restart|status}"
    exit 1
    ;;
esac

exit 0

# End swap
```

Liite 13 - /etc/rc.d/init.d/setclock

```
#!/bin/sh

#####

#

# Begin setclock

#

# Description : Setting Linux Clock

#

# Authors      : Gerard Beekmans - gerard@linuxfromscratch.org
#                DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
#####

### BEGIN INIT INFO
# Provides:
# Required-Start:
# Should-Start:      modules
# Required-Stop:
# Should-Stop:       $syslog
# Default-Start:     S
# Default-Stop:
# Short-Description: Stores and restores time from the hardware clock
# Description:        On boot, system time is obtained from hwclock.
The
#
#                hardware clock can also be set on shutdown.
# X-LFS-Provided-By:  LFS BLFS
### END INIT INFO

. /lib/lsb/init-functions

[ -r /etc/sysconfig/clock ] && . /etc/sysconfig/clock

case "${UTC}" in
    yes|true|1)
        CLOCKPARAMS="${CLOCKPARAMS} --utc"
```

```

;;

no|false|0)
    CLOCKPARAMS="${CLOCKPARAMS} --localtime"
;;

esac

case ${1} in
    start)
        hwclock --hctosys ${CLOCKPARAMS} >/dev/null
        ;;

    stop)
        log_info_msg "Setting hardware clock..."
        hwclock --systohc ${CLOCKPARAMS} >/dev/null
        evaluate_retval
        ;;

    *)
        echo "Usage: ${0} {start|stop}"
        exit 1
        ;;

esac

exit 0

```

Liite 14 - /etc/rc.d/init.d/checkfs

```
#!/bin/sh

#####
#
# Begin checkfs
#
# Description : File System Check
#
# Authors      : Gerard Beekmans - gerard@linuxfromscratch.org
#                A. Luebke - luebke@users.sourceforge.net
#                DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
# Based on checkfs script from LFS-3.1 and earlier.
#
# From man fsck
# 0    - No errors
# 1    - File system errors corrected
# 2    - System should be rebooted
# 4    - File system errors left uncorrected
# 8    - Operational error
# 16   - Usage or syntax error
# 32   - Fsck canceled by user request
# 128  - Shared library error
#
#####
##

### BEGIN INIT INFO
# Provides:          checkfs
# Required-Start:    udev swap $time
# Should-Start:
# Required-Stop:
# Should-Stop:
# Default-Start:     S
# Default-Stop:
# Short-Description: Checks local filesystems before mounting.
# Description:       Checks local filesystems before mounting.
```

```

# X-LFS-Provided-By:    LFS
### END INIT INFO

. /lib/lsb/init-functions

case "${1}" in
    start)
        if [ -f /fastboot ]; then
            msg="/fastboot found, will omit "
            msg="${msg} file system checks as requested.\n"
            log_info_msg "${msg}"
            exit 0
        fi

        log_info_msg "Mounting root file system in read-only mode... "
        mount -n -o remount,ro / >/dev/null

        if [ ${?} != 0 ]; then
            log_failure_msg2
            msg="\n\nCannot check root "
            msg="${msg}filesystem because it could not be mounted "
            msg="${msg}in read-only mode.\n\n"
            msg="${msg}After you press Enter, this system will be "
            msg="${msg}halted and powered off.\n\n"
            log_failure_msg "${msg}"

            log_info_msg "Press Enter to continue..."
            wait_for_user
            /etc/rc.d/init.d/halt stop
        else
            log_success_msg2
        fi

        if [ -f /forcefsck ]; then
            msg="\n/forcefsck found, forcing file"
            msg="${msg} system checks as requested."
            log_success_msg "${msg}"
            options="-f"
        else
            options=""
        fi
    fi

```

```

log_info_msg "Checking file systems..."
# Note: -a option used to be -p; but this fails e.g. on
fsck.minix
if is_true "$VERBOSE_FSCK"; then
    fsck ${options} -a -A -C -T
else
    fsck ${options} -a -A -C -T >/dev/null
fi

error_value=${?}

if [ "${error_value}" = 0 ]; then
    log_success_msg2
fi

if [ "${error_value}" = 1 ]; then
    msg="\nWARNING:\n\nFile system errors "
    msg="${msg}were found and have been corrected.\n"
    msg="${msg}You may want to double-check that "
    msg="${msg}everything was fixed properly."
    log_warning_msg "$msg"
fi

if [ "${error_value}" = 2 -o "${error_value}" = 3 ]; then
    msg="\nWARNING:\n\nFile system errors "
    msg="${msg}were found and have been been "
    msg="${msg}corrected, but the nature of the "
    msg="${msg}errors require this system to be rebooted.\n\n"
    msg="${msg}After you press enter, "
    msg="${msg}this system will be rebooted\n\n"
    log_failure_msg "$msg"

    log_info_msg "Press Enter to continue..."
    wait_for_user
    reboot -f
fi

if [ "${error_value}" -gt 3 -a "${error_value}" -lt 16 ]; then
    msg="\nFAILURE:\n\nFile system errors "
    msg="${msg}were encountered that could not be "

```

```

        msg="${msg}fixed automatically.  This system "
        msg="${msg}cannot continue to boot and will "
        msg="${msg}therefore be halted until those "
        msg="${msg}errors are fixed manually by a "
        msg="${msg}System Administrator.\n\n"
        msg="${msg}After you press Enter, this system will be "
        msg="${msg}halted and powered off.\n\n"
        log_failure_msg "$msg"

        log_info_msg "Press Enter to continue..."
        wait_for_user
        /etc/rc.d/init.d/halt stop
    fi

    if [ "${error_value}" -ge 16 ]; then
        msg="\nFAILURE:\n\nUnexpected Failure "
        msg="${msg}running fsck.  Exited with error "
        msg="${msg} code: ${error_value}."
        log_failure_msg $msg
        exit ${error_value}
    fi

    exit 0

;;
*)
    echo "Usage: ${0} {start}"
    exit 1

;;
esac

# End checkfs

```

Liite 15 - /etc/rc.d/init.d/mountfs

```
#!/bin/sh

#####

#

# Begin mountfs

#

# Description : File System Mount Script

#

# Authors      : Gerard Beekmans - gerard@linuxfromscratch.org
#                DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
#####

### BEGIN INIT INFO
# Provides:          $local_fs
# Required-Start:    udev checkfs
# Should-Start:
# Required-Stop:     swap
# Should-Stop:
# Default-Start:     S
# Default-Stop:      0 6
# Short-Description: Mounts/unmounts local filesystems defined in
/etc/fstab.
# Description:       Remounts root filesystem read/write and mounts
all
#                    remaining local filesystems defined in
/etc/fstab on
#                    start.  Remounts root filesystem read-only and
unmounts
#                    remaining filesystems on stop.
# X-LFS-Provided-By: LFS
### END INIT INFO

. /lib/lsb/init-functions

case "${1}" in
```

```

start)

    log_info_msg "Remounting root file system in read-write mode..."
    mount -o remount,rw / >/dev/null
    evaluate_retval

    # Remove fsck-related file system watermarks.
    rm -f /fastboot /forcefsck

    # Make sure /dev/pts exists
    mkdir -p /dev/pts

    # This will mount all filesystems that do not have _netdev in
    # their option list. _netdev denotes a network filesystem.

    log_info_msg "Mounting remaining file systems..."
    mount -a -O no_netdev >/dev/null
    evaluate_retval
    exit $failed
;;

stop)

    # Don't unmount virtual file systems like /run
    log_info_msg "Unmounting all other currently mounted file sys-
tems..."
    umount -a -d -r -t notmpfs,nosysfs,nodevtmpfs,noproc,nodevpts
>/dev/null
    evaluate_retval

    # Make sure / is mounted read only (umount bug)
    mount -o remount,ro /

    # Make all LVM volume groups unavailable, if appropriate
    # This fails if swap or / are on an LVM partition
    #if [ -x /sbin/vgchange ]; then /sbin/vgchange -an > /dev/null;
fi

;;

*)

    echo "Usage: ${0} {start|stop}"
    exit 1

```

```
;;  
esac  
  
# End mountfs
```

Liite 16 - /etc/rc.d/init.d/udev_retry

```
#!/bin/sh

#####
#
# Begin udev_retry
#
# Description : Udev cold-plugging script (retry)
#
# Authors      : Alexander E. Patrakov
#                DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#                Bryan Kadzban -
#
# Version      : LFS 7.0
#
#####

### BEGIN INIT INFO
# Provides:      udev_retry
# Required-Start: udev
# Should-Start:  $local_fs
# Required-Stop:
# Should-Stop:
# Default-Start: S
# Default-Stop:
# Short-Description: Replays failed uevents and creates additional
#                    devices.
# Description:      Replays any failed uevents that were skipped due
#                    to
#                    slow hardware initialization, and creates those
#                    needed
#                    device nodes
# X-LFS-Provided-By: LFS
### END INIT INFO

. /lib/lsb/init-functions

case "${1}" in
    start)
```

```

log_info_msg "Retrying failed uevents, if any..."

# As of udev-186, the --run option is no longer valid
#rundir=$(/sbin/udevadm info --run)
rundir=/run/udev
# From Debian: "copy the rules generated before / was mounted
# read-write":

for file in ${rundir}/tmp-rules--*; do
    dest=${file##*tmp-rules--}
    [ "$dest" = '*' ] && break
    cat $file >> /etc/udev/rules.d/$dest
    rm -f $file
done

# Re-trigger the uevents that may have failed,
# in hope they will succeed now
/bin/sed -e 's/#.*$//' /etc/sysconfig/udev_retry | /bin/grep -v
'^$' | \
while read line ; do
    for subsystem in $line ; do
        /sbin/udevadm trigger --subsystem-match=$subsystem --ac-
tion=add
    done
done

# Now wait for udevd to process the uevents we triggered
if ! is_true "$OMIT_UDEV_RETRY_SETTLE"; then
    /sbin/udevadm settle
fi

evaluate_retval
;;

*)
echo "Usage ${0} {start}"
exit 1
;;
esac

exit 0

```

```
# End udev_retry
```

Liite 17 - /etc/rc.d/init.d/cleanfs

```
#!/bin/sh

#####
#
# Begin cleanfs
#
# Description : Clean file system
#
# Authors      : Gerard Beekmans - gerard@linuxfromscratch.org
#                DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
#####

### BEGIN INIT INFO
# Provides:          cleanfs
# Required-Start:    $local_fs
# Should-Start:
# Required-Stop:
# Should-Stop:
# Default-Start:     S
# Default-Stop:
# Short-Description: Cleans temporary directories early in the boot
process.
# Description:       Cleans temporary directories /var/run,
/var/lock, and
#                    optionally, /tmp. cleanfs also creates
/var/run/utmp
#                    and any files defined in /etc/sysconfig/create-
files.
# X-LFS-Provided-By: LFS
### END INIT INFO

. /lib/lsb/init-functions

# Function to create files/directory on boot.
create_files()
```

```

{
    # Input to file descriptor 9 and output to stdin (redirection)
    exec 9>&0 < /etc/sysconfig/createfiles

    while read name type perm usr grp dtype maj min junk
    do
        # Ignore comments and blank lines.
        case "${name}" in
            ""|\#*) continue ;;
        esac

        # Ignore existing files.
        if [ ! -e "${name}" ]; then
            # Create stuff based on its type.
            case "${type}" in
                dir)
                    mkdir "${name}"
                    ;;
                file)
                    :> "${name}"
                    ;;
                dev)
                    case "${dtype}" in
                        char)
                            mknod "${name}" c ${maj} ${min}
                            ;;
                        block)
                            mknod "${name}" b ${maj} ${min}
                            ;;
                        pipe)
                            mknod "${name}" p
                            ;;
                        *)
                            log_warning_msg "\nUnknown device type: ${dtype}"
                            ;;
                    esac
                    ;;
                *)
                    log_warning_msg "\nUnknown type: ${type}"
                    continue
                    ;;
            esac
        fi
    done
}

```

```

        esac

        # Set up the permissions, too.
        chown ${usr}:${grp} "${name}"
        chmod ${perm} "${name}"
    fi
done

# Close file descriptor 9 (end redirection)
exec 0>&9 9>&-
return 0
}

case "${1}" in
    start)
        log_info_msg "Cleaning file systems:"

        if [ "${SKIPTMPCLEAN}" = "" ]; then
            log_info_msg2 " /tmp"
            cd /tmp &&
            find . -xdev -mindepth 1 ! -name lost+found -delete ||
failed=1
        fi

        > /var/run/utmp

        if grep -q '^utmp:' /etc/group ; then
            chmod 664 /var/run/utmp
            chgrp utmp /var/run/utmp
        fi

        (exit ${failed})
        evaluate_retval

        if egrep -qv '^(#|$)' /etc/sysconfig/createfiles 2>/dev/null;
then
            log_info_msg "Creating files and directories... "
            create_files      # Always returns 0
            evaluate_retval
        fi

```

```
        exit $failed
    ;;
*)
    echo "Usage: ${0} {start}"
    exit 1
    ;;
esac

# End cleanfs
```

Liite 18 - /etc/rc.d/init.d/console

```
#!/bin/sh

#####
#
# Begin console
#
# Description : Sets keymap and screen font
#
# Authors      : Gerard Beekmans - gerard@linuxfromscratch.org
#                Alexander E. Patrakov
#                DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
#####

### BEGIN INIT INFO
# Provides:          console
# Required-Start:
# Should-Start:      $local_fs
# Required-Stop:
# Should-Stop:
# Default-Start:     S
# Default-Stop:
# Short-Description: Sets up a localised console.
# Description:       Sets up fonts and language settings for the user's
#                    local as defined by /etc/sysconfig/console.
# X-LFS-Provided-By: LFS
### END INIT INFO

. /lib/lsb/init-functions

# Native English speakers probably don't have /etc/sysconfig/console at
all
[ -r /etc/sysconfig/console ] && . /etc/sysconfig/console

is_true()
```

```

{
    [ "$1" = "1" ] || [ "$1" = "yes" ] || [ "$1" = "true" ]
}

failed=0

case "${1}" in
    start)
        # See if we need to do anything
        if [ -z "${KEYMAP}" ] && [ -z "${KEYMAP_CORRECTIONS}" ]
&&
            [ -z "${FONT}" ] && [ -z "${LEGACY_CHARSET}" ]
&&
            ! is_true "${UNICODE}"; then
                exit 0
            fi

        # There should be no bogus failures below this line!
        log_info_msg "Setting up Linux console..."

        # Figure out if a framebuffer console is used
        [ -d /sys/class/graphics/fb0 ] && use_fb=1 || use_fb=0

        # Figure out the command to set the console into the
        # desired mode
        is_true "${UNICODE}" &&
            MODE_COMMAND="echo -en '\033%G' && kbd_mode -u" ||
            MODE_COMMAND="echo -en '\033%@033(K' && kbd_mode -a"

        # On framebuffer consoles, font has to be set for each vt in
        # UTF-8 mode. This doesn't hurt in non-UTF-8 mode also.

        ! is_true "${use_fb}" || [ -z "${FONT}" ] ||
            MODE_COMMAND="${MODE_COMMAND} && setfont ${FONT}"

        # Apply that command to all consoles mentioned in
        # /etc/inittab. Important: in the UTF-8 mode this should
        # happen before setfont, otherwise a kernel bug will
        # show up and the unicode map of the font will not be
        # used.

```

```

for TTY in `grep '^[^#].*respawn:/sbin/agetty' /etc/inittab |
    grep -o '\bttty[[:digit:]]*\b'`
do
    openvt -f -w -c ${TTY#tty} -- \
        /bin/sh -c "${MODE_COMMAND}" || failed=1
done

# Set the font (if not already set above) and the keymap
[ "${use_fb}" == "1" ] || [ -z "${FONT}" ] || setfont $FONT ||
failed=1

[ -z "${KEYMAP}" ] ||
    loadkeys ${KEYMAP} >/dev/null 2>&1 ||
    failed=1

[ -z "${KEYMAP_CORRECTIONS}" ] ||
    loadkeys ${KEYMAP_CORRECTIONS} >/dev/null 2>&1 ||
    failed=1

# Convert the keymap from $LEGACY_CHARSET to UTF-8
[ -z "${LEGACY_CHARSET}" ] ||
    dumpkeys -c "${LEGACY_CHARSET}" | loadkeys -u >/dev/null 2>&1 ||
    failed=1

# If any of the commands above failed, the trap at the
# top would set $failed to 1
( exit $failed )
evaluate_retval

exit $failed
;;

*)
    echo "Usage:  ${0} {start}"
    exit 1
    ;;
esac

# End console

```

Liite 19 - /etc/rc.d/init.d/localnet

```
#!/bin/sh

#####

#

# Begin localnet

#

# Description : Loopback device

#

# Authors      : Gerard Beekmans - gerard@linuxfromscratch.org
#                DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
#####

### BEGIN INIT INFO
# Provides:          localnet
# Required-Start:    $local_fs
# Should-Start:
# Required-Stop:
# Should-Stop:
# Default-Start:     S
# Default-Stop:      0 6
# Short-Description: Starts the local network.
# Description:       Sets the hostname of the machine and starts the
#                    loopback interface.
# X-LFS-Provided-By: LFS
### END INIT INFO

. /lib/lsb/init-functions
[ -r /etc/sysconfig/network ] && . /etc/sysconfig/network
[ -r /etc/hostname ] && HOSTNAME=`cat /etc/hostname`

case "${1}" in
    start)
        log_info_msg "Bringing up the loopback interface..."
        ip addr add 127.0.0.1/8 label lo dev lo
        ip link set lo up
    ;;
    stop)
        ip link set lo down
    ;;
    *)
        echo "Usage: localnet {start|stop}"
        exit 1
    ;;
esac
```

```

        evaluate_retval

        log_info_msg "Setting hostname to ${HOSTNAME}..."
        hostname ${HOSTNAME}
        evaluate_retval
        ;;

stop)
    log_info_msg "Bringing down the loopback interface..."
    ip link set lo down
    evaluate_retval
    ;;

restart)
    ${0} stop
    sleep 1
    ${0} start
    ;;

status)
    echo "Hostname is: $(hostname)"
    ip link show lo
    ;;

*)
    echo "Usage: ${0} {start|stop|restart|status}"
    exit 1
    ;;
esac

exit 0

# End localnet

```

Liite 20 - /etc/rc.d/init.d/sysctl

```
#!/bin/sh

#####

#

# Begin sysctl
#
# Description : File uses /etc/sysctl.conf to set kernel runtime
#               parameters
#
# Authors      : Nathan Coulson (nathan@linuxfromscratch.org)
#               Matthew Burgess (matthew@linuxfromscratch.org)
#               DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
#####

### BEGIN INIT INFO
# Provides:          sysctl
# Required-Start:    mountvirtfs
# Should-Start:
# Required-Stop:
# Should-Stop:
# Default-Start:     S
# Default-Stop:
# Short-Description: Makes changes to the proc filesystem
# Description:       Makes changes to the proc filesystem as defined
in
#
#                   /etc/sysctl.conf.  See 'man sysctl(8)'.
# X-LFS-Provided-By:  LFS
### END INIT INFO

. /lib/lsb/init-functions

case "${1}" in
    start)
        if [ -f "/etc/sysctl.conf" ]; then
            log_info_msg "Setting kernel runtime parameters..."

```

```
        sysctl -q -p
        evaluate_retval
    fi
    ;;

    status)
        sysctl -a
        ;;

    *)
        echo "Usage: ${0} {start|status}"
        exit 1
        ;;
esac

exit 0

# End sysctl
```

Liite 21 - /etc/rc.d/init.d/sysklogd

```
#!/bin/sh

#####

#

# Begin sysklogd

#

# Description : Sysklogd loader

#

# Authors      : Gerard Beekmans - gerard@linuxfromscratch.org
#                DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
#####

### BEGIN INIT INFO
# Provides:          $syslog
# Required-Start:    localnet
# Should-Start:
# Required-Stop:     $local_fs sendsignals
# Should-Stop:
# Default-Start:     3 4 5
# Default-Stop:      0 1 2 6
# Short-Description: Starts kernel and system log daemons.
# Description:       Starts kernel and system log daemons.
#                    /etc/fstab.
# X-LFS-Provided-By: LFS
### END INIT INFO

# Note: sysklogd is not started in runlevel 2 due to possible
# remote logging configurations

. /lib/lsb/init-functions

case "${1}" in
    start)
        log_info_msg "Starting system log daemon..."
        parms=${SYSKLOGD_PARAMS-'-m 0'}

```

```

start_daemon /sbin/syslogd $parms
evaluate_retval

log_info_msg "Starting kernel log daemon..."
start_daemon /sbin/klogd
evaluate_retval
;;

stop)
log_info_msg "Stopping kernel log daemon..."
killproc /sbin/klogd
evaluate_retval

log_info_msg "Stopping system log daemon..."
killproc /sbin/syslogd
evaluate_retval
;;

reload)
log_info_msg "Reloading system log daemon config file..."
pid=`pidofproc syslogd`
kill -HUP "${pid}"
evaluate_retval
;;

restart)
${0} stop
sleep 1
${0} start
;;

status)
statusproc /sbin/syslogd
statusproc klogd
;;

*)
echo "Usage: ${0} {start|stop|reload|restart|status}"
exit 1
;;

esac

```

```
exit 0
```

```
# End syslogd
```

Liite 22 - /etc/rc.d/init.d/network

```
#!/bin/sh

#####
#
# Begin network
#
# Description : Network Control Script
#
# Authors      : Gerard Beekmans - gerard@linuxfromscratch.org
#                Nathan Coulson - nathan@linuxfromscratch.org
#                Kevin P. Fleming - kpfleming@linuxfromscratch.org
#                DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
#####

### BEGIN INIT INFO
# Provides:          $network
# Required-Start:    $local_fs swap localnet
# Should-Start:      $syslog
# Required-Stop:     $local_fs swap localnet
# Should-Stop:       $syslog
# Default-Start:     3 4 5
# Default-Stop:      0 1 2 6
# Short-Description: Starts and configures network interfaces.
# Description:       Starts and configures network interfaces.
# X-LFS-Provided-By: LFS
### END INIT INFO

case "${1}" in
    start)
        # Start all network interfaces
        for file in /etc/sysconfig/ifconfig.*
        do
            interface=${file##*/ifconfig.}

            # Skip if $file is * (because nothing was found)
```

```

        if [ "${interface}" = "*" ]
        then
            continue
        fi

        /sbin/ifup ${interface}
    done
;;

stop)
    # Reverse list
    net_files=""
    for file in /etc/sysconfig/ifconfig.*
    do
        net_files="${file} ${net_files}"
    done

    # Stop all network interfaces
    for file in ${net_files}
    do
        interface=${file##*/ifconfig.}

        # Skip if $file is * (because nothing was found)
        if [ "${interface}" = "*" ]
        then
            continue
        fi

        /sbin/ifdown ${interface}
    done
;;

restart)
    ${0} stop
    sleep 1
    ${0} start
;;

*)
    echo "Usage: ${0} {start|stop|restart}"
    exit 1

```

```
;;  
esac  
  
exit 0  
  
# End network
```

Liite 23 - /etc/rc.d/init.d/sendsignals

```
#!/bin/sh

#####
#
# Begin sendsignals
#
# Description : Sendsignals Script
#
# Authors      : Gerard Beekmans - gerard@linuxfromscratch.org
#               DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
#####

### BEGIN INIT INFO
# Provides:          sendsignals
# Required-Start:
# Should-Start:
# Required-Stop:     $local_fs swap localnet
# Should-Stop:
# Default-Start:
# Default-Stop:      0 6
# Short-Description: Attempts to kill remaining processes.
# Description:       Attempts to kill remaining processes.
# X-LFS-Provided-By:  LFS
### END INIT INFO

. /lib/lsb/init-functions

case "${1}" in
    stop)
        log_info_msg "Sending all processes the TERM signal..."
        killall15 -15
        error_value=${?}

        sleep ${KILLDELAY}
    ;;
esac
```

```

        if [ "${error_value}" = 0 -o "${error_value}" = 2 ]; then
            log_success_msg
        else
            log_failure_msg
        fi

        log_info_msg "Sending all processes the KILL signal..."
        killall5 -9
        error_value=${?}

        sleep ${KILLDELAY}

        if [ "${error_value}" = 0 -o "${error_value}" = 2 ]; then
            log_success_msg
        else
            log_failure_msg
        fi
        ;;

    *)
        echo "Usage: ${0} {stop}"
        exit 1
        ;;

esac

exit 0

# End sendsignals

```

Liite 24 - /etc/rc.d/init.d/reboot

```
#!/bin/sh

#####

#

# Begin reboot

#

# Description : Reboot Scripts

#

# Authors      : Gerard Beekmans - gerard@linuxfromscratch.org
#                DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
#####

### BEGIN INIT INFO
# Provides:          reboot
# Required-Start:
# Should-Start:
# Required-Stop:
# Should-Stop:
# Default-Start:     6
# Default-Stop:
# Short-Description: Reboots the system.
# Description:       Reboots the System.
# X-LFS-Provided-By:  LFS
### END INIT INFO

. /lib/lsb/init-functions

case "${1}" in
    stop)
        log_info_msg "Restarting system..."
        reboot -d -f -i
        ;;

    *)
        echo "Usage: ${0} {stop}"

```

```
        exit 1
    ;;

esac

# End reboot
```

Liite 25 - /etc/rc.d/init.d/halt

```
#!/bin/sh

#####
#
# Begin halt
#
# Description : Halt Script
#
# Authors      : Gerard Beekmans - gerard@linuxfromscratch.org
#                DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
#####

### BEGIN INIT INFO
# Provides:          halt
# Required-Start:
# Should-Start:
# Required-Stop:
# Should-Stop:
# Default-Start:     0
# Default-Stop:
# Short-Description: Halts the system.
# Description:       Halts the System.
# X-LFS-Provided-By:  LFS
### END INIT INFO

case "${1}" in
    stop)
        halt -d -f -i -p
        ;;

    *)
        echo "Usage: {stop}"
        exit 1
        ;;
esac
```

```
# End halt
```

Liite 26 - /etc/rc.d/init.d/template

```
#!/bin/sh

#####

#

# Begin scriptname
#
# Description :
#
# Authors      :
#
# Version      : LFS x.x
#
# Notes        :
#
#####

### BEGIN INIT INFO
# Provides:          template
# Required-Start:
# Should-Start:
# Required-Stop:
# Should-Stop:
# Default-Start:
# Default-Stop:
# Short-Description:
# Description:
# X-LFS-Provided-By:
### END INIT INFO

. /lib/lsb/init-functions

case "${1}" in
    start)
        log_info_msg "Starting..."
        start_daemon fully_qualified_path
        ;;

    stop)
        log_info_msg "Stopping..."

```

```
killproc fully_qualified_path
;;

restart)
    ${0} stop
    sleep 1
    ${0} start
    ;;

*)
    echo "Usage: ${0} {start|stop|restart}"
    exit 1
    ;;
esac

exit 0

# End scriptname
```

Liite 27 - /etc/sysconfig/modules

```
#####  
#  
# Begin /etc/sysconfig/modules  
#  
# Description : Module auto-loading configuration  
#  
# Authors      :  
#  
# Version      : 00.00  
#  
# Notes        : The syntax of this file is as follows:  
#                 <module> [<arg1> <arg2> ...]  
#  
# Each module should be on its own line, and any options that you want  
# passed to the module should follow it. The line delimiter is ei-  
# ther  
# a space or a tab.  
#####  
#  
  
# End /etc/sysconfig/modules
```

Liite 28 - /etc/sysconfig/createfiles

```
#####  
#  
# Begin /etc/sysconfig/createfiles  
#  
# Description : Createfiles script config file  
#  
# Authors      :  
#  
# Version      : 00.00  
#  
# Notes        : The syntax of this file is as follows:  
#                 if type is equal to "file" or "dir"  
#                 <filename> <type> <permissions> <user> <group>  
#                 if type is equal to "dev"  
#                 <filename> <type> <permissions> <user> <group> <devtype>  
#                 <major> <minor>  
#  
#                 <filename> is the name of the file which is to be created  
#                 <type> is either file, dir, or dev.  
#                 file creates a new file  
#                 dir creates a new directory  
#                 dev creates a new device  
#                 <devtype> is either block, char or pipe  
#                 block creates a block device  
#                 char creates a character device  
#                 pipe creates a pipe, this will ignore the <major> and  
#                 <minor> fields  
#                 <major> and <minor> are the major and minor numbers used for  
#                 the device.  
#####  
#  
  
# End /etc/sysconfig/createfiles
```

Liite 29 - /etc/sysconfig/udev-retry

```
#####  
#  
# Begin /etc/sysconfig/udev_retry  
#  
# Description : udev_retry script configuration  
#  
# Authors      :  
#  
# Version      : 00.00  
#  
# Notes        : Each subsystem that may need to be re-triggered after  
mountfs  
#                runs should be listed in this file.  Probable subsys-  
tems to be  
#                listed here are rtc (due to /var/lib/hwclock/adjtime)  
and sound  
#                (due to both /var/lib/alsa/asound.state and  
/usr/sbin/alsactl).  
#                Entries are whitespace-separated.  
#####  
#  
  
rtc  
  
# End /etc/sysconfig/udev_retry
```

Liite 30 - /sbin/ifup

```
#!/bin/sh

#####
#
# Begin /sbin/ifup
#
# Description : Interface Up
#
# Authors      : Nathan Coulson - nathan@linuxfromscratch.org
#                Kevin P. Fleming - kpffleming@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.7
#
# Notes        : The IFCONFIG variable is passed to the SERVICE script
#                in the /lib/services directory, to indicate what file
the
#                service should source to get interface specifications.
#
#####
#

up()
{
    if ip link show $1 > /dev/null 2>&1; then
        link_status=`ip link show $1`

        if [ -n "${link_status}" ]; then
            if ! echo "${link_status}" | grep -q UP; then
                ip link set $1 up
            fi
        fi

    else
        log_failure_msg "\nInterface ${IFACE} doesn't exist."
        exit 1
    fi
}

RELEASE="7.7"
```

```

USAGE="Usage: $0 [ -hV ] [--help] [--version] interface"
VERSTR="LFS ifup, version ${RELEASE}"

while [ $# -gt 0 ]; do
    case "$1" in
        --help | -h)      help="y"; break ;;

        --version | -V)   echo "${VERSTR}"; exit 0 ;;

        -*)               echo "ifup: ${1}: invalid option" >&2
                           echo "${USAGE}" >& 2
                           exit 2 ;;

        *)               break ;;
    esac
done

if [ -n "$help" ]; then
    echo "${VERSTR}"
    echo "${USAGE}"
    echo
    cat << HERE_EOF
ifup is used to bring up a network interface.  The interface
parameter, e.g. eth0 or eth0:2, must match the trailing part of the
interface specifications file, e.g. /etc/sysconfig/ifconfig.eth0:2.

HERE_EOF
    exit 0
fi

file=/etc/sysconfig/ifconfig.${1}

# Skip backup files
[ "${file}" = "${file%""~""}" ] || exit 0

. /lib/lsb/init-functions

log_info_msg "Bringing up the ${1} interface... "

if [ ! -r "${file}" ]; then

```

```

    log_failure_msg2 "${file} is missing or cannot be accessed."
    exit 1
fi

. $file

if [ "$IFACE" = "" ]; then
    log_failure_msg2 "${file} does not define an interface [IFACE]."
    exit 1
fi

# Do not process this service if started by boot, and ONBOOT
# is not set to yes
if [ "${IN_BOOT}" = "1" -a "${ONBOOT}" != "yes" ]; then
    log_skip_msg
    exit 0
fi

for S in ${SERVICE}; do
    if [ ! -x "/lib/services/${S}" ]; then
        MSG="\nUnable to process ${file}. Either "
        MSG="${MSG}the SERVICE '${S}' was not present "
        MSG="${MSG}or cannot be executed."
        log_failure_msg "$MSG"
        exit 1
    fi
done

if [ "${SERVICE}" = "wpa" ]; then log_success_msg; fi

# Create/configure the interface
for S in ${SERVICE}; do
    IFCONFIG=${file} /lib/services/${S} ${IFACE} up
done

# Bring up the interface and any components
for I in $IFACE $INTERFACE_COMPONENTS; do up $I; done

# Set MTU if requested. Check if MTU has a "good" value.
if test -n "${MTU}"; then
    if [[ ${MTU} =~ ^[0-9]+$ ]] && [[ $MTU -ge 68 ]]; then

```

```
        for I in $IFACE $INTERFACE_COMPONENTS; do
            ip link set dev $I mtu $MTU;
        done
    else
        log_info_msg2 "Invalid MTU $MTU"
    fi
fi

# Set the route default gateway if requested
if [ -n "${GATEWAY}" ]; then
    if ip route | grep -q default; then
        log_skip_msg "\n      Gateway already setup; skipping."
    else
        log_info_msg "Setting up default gateway..."
        ip route add default via ${GATEWAY} dev ${IFACE}
        evaluate_retval
    fi
fi

# End /sbin/ifup
```

Liite 31 - /sbin/ifdown

```
#!/bin/bash
#####
#
# Begin /sbin/ifdown
#
# Description : Interface Down
#
# Authors      : Nathan Coulson - nathan@linuxfromscratch.org
#               Kevin P. Fleming - kpffleming@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
# Notes        : the IFCONFIG variable is passed to the scripts found
#               in the /lib/services directory, to indicate what file
the
#               service should source to get interface specifications.
#
#####
#

RELEASE="7.0"

USAGE="Usage: $0 [ -hV ] [--help] [--version] interface"
VERSTR="LFS ifdown, version ${RELEASE}"

while [ $# -gt 0 ]; do
    case "$1" in
        --help | -h)      help="y"; break ;;

        --version | -V)    echo "${VERSTR}"; exit 0 ;;

        -*)                echo "ifup: ${1}: invalid option" >&2
                           echo "${USAGE}" >& 2
                           exit 2 ;;

        *)                 break ;;
    esac
done
```

```

if [ -n "$help" ]; then
    echo "${VERSTR}"
    echo "${USAGE}"
    echo
    cat << HERE_EOF
ifdown is used to bring down a network interface.  The interface
parameter, e.g. eth0 or eth0:2, must match the trailing part of the
interface specifications file, e.g. /etc/sysconfig/ifconfig.eth0:2.

HERE_EOF
    exit 0
fi

file=/etc/sysconfig/ifconfig.${1}

# Skip backup files
[ "${file}" = "${file%""~""}" ] || exit 0

. /lib/lsb/init-functions

if [ ! -r "${file}" ]; then
    log_warning_msg "${file} is missing or cannot be accessed."
    exit 1
fi

. ${file}

if [ "$IFACE" = "" ]; then
    log_failure_msg "${file} does not define an interface [IFACE]."
    exit 1
fi

# We only need to first service to bring down the interface
S=`echo ${SERVICE} | cut -f1 -d" "`

if ip link show ${IFACE} > /dev/null 2>&1; then
    if [ -n "${S}" -a -x "/lib/services/${S}" ]; then
        IFCONFIG=${file} /lib/services/${S} ${IFACE} down
    else
        MSG="Unable to process ${file}.  Either "

```

```

        MSG="${MSG}the SERVICE variable was not set "
        MSG="${MSG}or the specified service cannot be executed."
        log_failure_msg "$MSG"
        exit 1
    fi
else
    log_warning_msg "Interface ${1} doesn't exist."
fi

# Leave the interface up if there are additional interfaces in the de-
vice
link_status=`ip link show ${IFACE} 2>/dev/null`

if [ -n "${link_status}" ]; then
    if [ "$(echo "${link_status}" | grep UP)" != "" ]; then
        if [ "$(ip addr show ${IFACE} | grep 'inet ')" == "" ]; then
            log_info_msg "Bringing down the ${IFACE} interface..."
            ip link set ${IFACE} down
            evaluate_retval
        fi
    fi
fi

# End /sbin/ifdown

```

Liite 32 - /lib/services/ipv4-static

```
#!/bin/sh

#####
#
# Begin /lib/services/ipv4-static
#
# Description : IPV4 Static Boot Script
#
# Authors      : Nathan Coulson - nathan@linuxfromscratch.org
#               Kevin P. Fleming - kpflaming@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
#####

. /lib/lsb/init-functions
. ${IFCONFIG}

if [ -z "${IP}" ]; then
    log_failure_msg "\nIP variable missing from ${IFCONFIG}, cannot con-
    tinue."
    exit 1
fi

if [ -z "${PREFIX}" -a -z "${PEER}" ]; then
    log_warning_msg "\nPREFIX variable missing from ${IFCONFIG}, assum-
    ing 24."
    PREFIX=24
    args="${args} ${IP}/${PREFIX}"
fi

elif [ -n "${PREFIX}" -a -n "${PEER}" ]; then
    log_failure_msg "\nPREFIX and PEER both specified in ${IFCONFIG},
    cannot continue."
    exit 1
fi

elif [ -n "${PREFIX}" ]; then
    args="${args} ${IP}/${PREFIX}"
fi
```

```

elif [ -n "${PEER}" ]; then
    args="${args} ${IP} peer ${PEER}"
fi

if [ -n "${LABEL}" ]; then
    args="${args} label ${LABEL}"
fi

if [ -n "${BROADCAST}" ]; then
    args="${args} broadcast ${BROADCAST}"
fi

case "${2}" in
    up)
        if [ "$(ip addr show ${1} 2>/dev/null | grep ${IP}/)" = "" ];
        then

            # Cosmetic output
            if ! $(echo ${SERVICE} | grep -q " "); then
                log_info_msg2 "\n" # Terminate the previous message
            fi

            log_info_msg "Adding IPv4 address ${IP} to the ${1} inter-
face..."
            ip addr add ${args} dev ${1}
            evaluate_retval
        else
            log_warning_msg "Cannot add IPv4 address ${IP} to ${1}. Al-
ready present."
        fi
        ;;

    down)
        if [ "$(ip addr show ${1} 2>/dev/null | grep ${IP}/)" != "" ];
        then
            log_info_msg "Removing IPv4 address ${IP} from the ${1} inter-
face..."
            ip addr del ${args} dev ${1}
            evaluate_retval
        fi
    esac

```

```
        if [ -n "${GATEWAY}" ]; then
            # Only remove the gateway if there are no remaining ipv4 ad-
addresses
            if [ "$(ip addr show ${1} 2>/dev/null | grep 'inet ')" != ""
]; then
                log_info_msg "Removing default gateway..."
                ip route del default
                evaluate_retval
            fi
        fi
    ;;

*)
    echo "Usage: ${0} [interface] {up|down}"
    exit 1
    ;;
esac

# End /lib/services/ipv4-static
```

Liite 33 - /lib/services/ipv4-static-route

```
#!/bin/sh

#####
#
# Begin /lib/services/ipv4-static-route
#
# Description : IPV4 Static Route Script
#
# Authors      : Kevin P. Fleming - kpflaming@linuxfromscratch.org
#               DJ Lucas - dj@linuxfromscratch.org
# Update       : Bruce Dubbs - bdubbs@linuxfromscratch.org
#
# Version      : LFS 7.0
#
#####

. /lib/lsb/init-functions
. ${IFCONFIG}

case "${TYPE}" in
    (" " | "network")
        need_ip=1
        need_gateway=1
        ;;

    ("default")
        need_gateway=1
        args="${args} default"
        desc="default"
        ;;

    ("host")
        need_ip=1
        ;;

    ("unreachable")
        need_ip=1
        args="${args} unreachable"
        desc="unreachable "
```

```

;;

(*)
    log_failure_msg "Unknown route type (${TYPE}) in ${IFCONFIG},
cannot continue."
    exit 1
;;
esac

if [ -n "${GATEWAY}" ]; then
    MSG="The GATEWAY variable cannot be set in ${IFCONFIG} for static
routes.\n"
    log_failure_msg "$MSG Use STATIC_GATEWAY only, cannot continue"
    exit 1
fi

if [ -n "${need_ip}" ]; then
    if [ -z "${IP}" ]; then
        log_failure_msg "IP variable missing from ${IFCONFIG}, cannot
continue."
        exit 1
    fi

    if [ -z "${PREFIX}" ]; then
        log_failure_msg "PREFIX variable missing from ${IFCONFIG}, cannot
continue."
        exit 1
    fi

    args="${args} ${IP}/${PREFIX}"
    desc="${desc}${IP}/${PREFIX}"
fi

if [ -n "${need_gateway}" ]; then
    if [ -z "${STATIC_GATEWAY}" ]; then
        log_failure_msg "STATIC_GATEWAY variable missing from ${IFCON-
FIG}, cannot continue."
        exit 1
    fi
    args="${args} via ${STATIC_GATEWAY}"
fi

```

```

if [ -n "${SOURCE}" ]; then
    args="${args} src ${SOURCE}"
fi

case "${2}" in
    up)
        log_info_msg "Adding '${desc}' route to the ${1} interface..."
        ip route add ${args} dev ${1}
        evaluate_retval
        ;;

    down)
        log_info_msg "Removing '${desc}' route from the ${1} inter-
face..."
        ip route del ${args} dev ${1}
        evaluate_retval
        ;;

    *)
        echo "Usage: ${0} [interface] {up|down}"
        exit 1
        ;;
esac

# End /lib/services/ipv4-static-route

```