

APPENDIX I

Installation of Gentoo Linux on Soekris

1. Before start

This is an installation guide for Soekris using Gentoo Linux. Actually, there are a lot of different ways we can use such as PXE boot with NFS filesystem, embedded Linux installation and so on. But in our case, because we have to use university's DHCP servers and switches, we cannot use PXE boot for our Soekris. So, we decided to install embedded Linux on each Soekris, and we keep searching a better way to manage whole Soekris together conveniently. If you check some other web-pages (check 5. Web-pages related with OS installation of Soekris), you can see other better methods which are more suited to your environment. We made this installation guide by referring all other people's work.

2. Something you need

- A computer as a server: We use Gentoo Linux as an operating system of our server.
- Soekris : This should be assembled with all modules you need such as MiniPCIs, antennas and cables (see APPENDIX II)
- Power adapter of Soekris (sold separately. If you would like to use PoE instead of using power adapter, see APPENDIX III)
- One cross Ethernet cable or two direct Ethernet cables and one switch to connect between a computer and a Soekris by Ethernet
- One serial cable: (9holes, Female) ----- (9holes, Female). This is for displaying the terminal from Soekris. If you don't have any serial port on your computer, you can use a USB port, but you need a USB-SERIAL convert cable and this serial cable together.
- Internet connection for downloading Gentoo Linux installation files



3. How to Configure a Server with Gentoo Linux

3.1. Server System Specification

- Desktop computer from DELL
- Intel Pentium4 with Gentoo Linux OS (We use Gentoo Linux OS for a server and each Soekris. You may need to have background knowledge about Gentoo Linux. Please check the web-site; www.gentoo.org)
- Kernel version : 2.6.17.6
- Version of Gcc : i686-pc-linux-gnu-3.3.6 (currently default gcc version is 4.1.1. You may be able to use gcc-4.1.1 for whole process of below. If you would face any problem with gcc-4.1.1, we recommend you to install lower version of gcc like gcc-3.3.6.)

3.2. Soentoo

About

Soentoo is originally from <http://soentoo.sourceforge.net/>. The author (johannesbaue) of Soentoo stopped updating his code for the new version of each Gentoo installation file. So, based on the author's work we modified Soentoo for our own purpose with Soekris net4826. We will provide a modified Soentoo.

Step 1: Grabbing the files

You first need to download some files (besides the Soentoo files). This would be:

- Gentoo Stage3-Installation file (e.g. from [here](#))
- Gentoo Portage snapshot (e.g. from [here](#))
- A kernel (e.g. from [here](#))
- A kernel patch (which fits your kernel, from [here](#)) **Only needed for kernels <= 2.6.11**

In my example, I will use the following files:

- Gentoo Stage3-Installation: stage3-x86-2006.0.tar.bz2
- Gentoo Portage snapshot: portage-20061226.tar.bz2
- A current kernel: linux-2.6.18.tar.bz2
- A kernel patch: I don't use kernel patch because I used recent kernel version.

Make a dir like "/root/soentoo", and place all files you downloaded in there

```
tabak soentoo # pwd
/root/soentoo
tabak soentoo # ls
Soentoo-0.05.tar.bz2  linux-2.6.18.6.tar.bz2  portage-20061226.tar.bz2  stage3-x86-2006.0.tar.bz2
```

Extract all Soentoo files into the directory

```
tabak soentoo # tar xvj Soentoo-0.05.tar.bz2
```

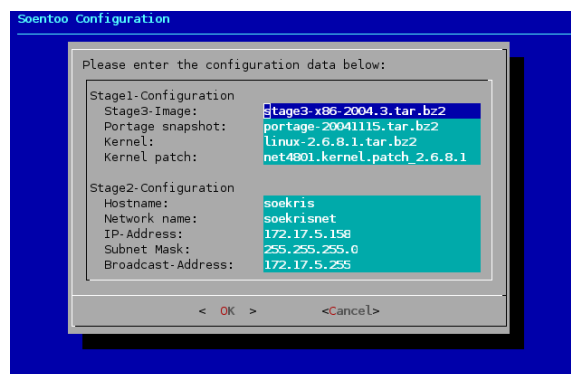
Step 2: Configuring the Soentoo Setup

This really is easy. Change to the "Soentoo-0.05" directory and type "./Configure_Soentoo":

```
tabak soentoo # cd Soentoo-0.05
```

```
tabak soentoo # ./Configure_Soentoo
```

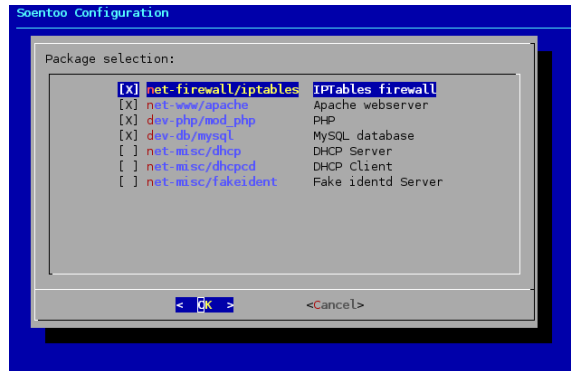
You'll need to have "dialog" installed for this to work (which you are very likely to already have). Alternatively, you can edit the "Stage1.conf" and "Stage2.conf" configuration files, which are pretty self-explanatory. There will be three configuration screens which I will explain in detail:



First dialog (version of each files may be different)

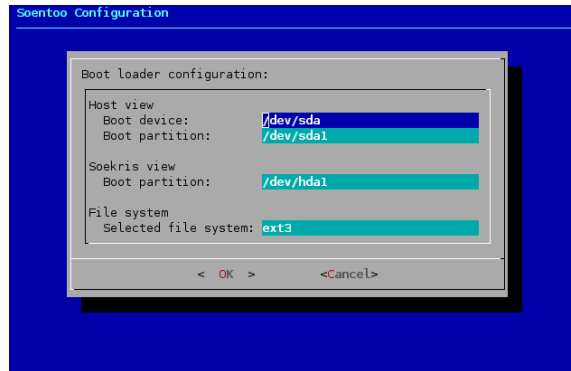
| Item | Description |
|------------------|---|
| Stage3-Image | Enter the filename of the Gentoo Stage3 image you downloaded. |
| Portage snapshot | Enter the filename of the Gentoo portage snapshot you downloaded. |
| Kernel | Enter the filename of the Linux kernel source tree you downloaded. |
| Kernel patch | If you'd like to patch the kernel you downloaded, enter the filename to the patch here. If not, you can leave this field empty. Patching with the appropriate Soekris Kernel patch is highly recommended, though. |
| Hostname | The hostname of your Soekris box, i.e. how your box will be called on the network. (You can change this later) |
| Network name | The domain name of your Soekris box, i.e. how the network is called on which |

| | |
|-------------------|---|
| | your Soekris box is connected to. (You can change this later) |
| IP Address | The IP address of eth0 (e.g. 192.168.1.123). (You can change this later) |
| Subnet mask | The subnet mask of eth0 (e.g. 255.255.255.0). (You can change this later) |
| Broadcast address | The broadcast address of eth0 (e.g. 192.168.1.255). (You can change this later) |



Second dialog

The second dialog lets you select what packages you want to install. You will need to watch the overall size of the distribution yourself as it may well exceed 128MB when you select everything. In my case, I select iptables, dhcp and dhcpd.



Third dialog

| Item | Description |
|-----------------------------|--|
| Host view boot device | In our case, just type “/dev/had” |
| Host view boot partition | In our case, just type “/dev/hda1” |
| Soekris view boot partition | This is the partition which your Soekris will boot from from the Soekris's perspective. In our case, just type “/dev/hda1” |

| | |
|-------------|--|
| File system | I use ext3 here. You'll of course have to take care of the correct formatting of the partition yourself. In our case, use "ext2" |
|-------------|--|

Also note that in the third step of the configuration nothing is written to any bootsector or whatever. So during configuration you can't break anything on your hddisk. This is just a nice interface to configure /etc/lilo.conf and /etc/fstab in your Image directory. However, as soon as you run lilo, you should be sure these settings are correct!

Step 3: Bootstrapping Soentoo

This is almost as easy as configuration: Just type in your Soentoo-directory

```
tabak soentoo-0.05 # ./Create_Image
```

And it will magically install itself! The setup has very little verbosity. If you would like to know more and watch the installation process, you can switch to another terminal and follow the files "Soekris_Stage1_Debug.log". You can do so by typing

```
tabak soentoo-0.05 # tail -f Soekris_Stage1_Debug.log
```

As soon as the Stage1-installation is finished, it will automatically switch to Stage2 (e.g. chrooting to the created environment, emerging packets and so on). You can then follow the installation progress by typing

```
tabak soentoo-0.05 # tail -f Image/soekris/Soekris_Stage2_Debug.log
```

The whole process should somewhat look like

```
joe [/home/joe/Soekris]: ./Create_Image
Extracting Stage3-Image : 1:09 ms
Extracting Portage snapshot: 0:44 ms
Extracting kernel : 0:45 ms
Complete Stage1 installation took 2:39 ms.

Now leaving Stage1 and entering Stage2 of installation...
...Stage2 of installation started.

Starting kernel compilation: 5:19 ms
Emerging sys-boot/lilo : 0:54 ms
Emerging iptables : 0:34 ms
Emerging dev-db/mysql : 15:15 ms
Emerging dev-php/mod_php : 29:19 ms
Emerging net-www/apache : 5:55 ms
Complete Stage2 installation took 57:20 ms.

Stage 2 installation completed. The image is now bootable and could be
stripped.

joe [/home/joe/Soekris]: []
```

Step 4: Customizing Soentoo

When it all went fine, you have a working Soentoo in the "Image/" path now. You can chroot to the environment by issuing like

```
tabak soentoo-0.05 # cd Image
tabak Image # chroot.
tabak / #
```

From there, you can emerge packets you'd like to additionally have or remove files which you don't want to have on there.

Step 5: Stripping Soentoo

In order to make Soentoo really tight and small, there are stripping scripts included. If you are installing on a hard disk and don't need to care about the space, you can skip this step.

Do the stripping only after you've done everything else! You will not be able to emerge anything anymore after this point, as portage itself will be removed. You might want to backup the whole "Image/" tree in case you want to change something and don't want to re-emerge all packets. As soon as you're ready for stripping, chroot to your new environment and type:

```
tabak / # cd /soekris
tabak soekris # ./Strip_Image
```

Please make sure you're really in the right environment! This is very important. Although there are a number of safety features installed, I do not want you to strip down your "real" Gentoo installation. This would be quite unfortunate... So please ensure you're really chrooted to your Soentoo-tree (see step 4).

The whole process should somewhat look like

```
joe [/soekris]: ./Strip_Image
To strip an image, you need to be 'chrooted' on it.
Are you absolutely sure this is the correct environment?
You may ruin your system otherwise!
Are you sure (y/n) y
Giving you 10 more seconds to think it over...
S...4...3...2...1...
We're going in...
Unmerging sys-apps/slocate           : 0:01 ms
Unmerging app-editors/nano           : 0:11 ms
Unmerging app-shells/sash            : 0:17 ms
Unmerging dev-lang/perl              : 0:21 ms
Unmerging dev-util/yacc              : 0:07 ms
Unmerging sys-apps/groff             : 0:19 ms
Unmerging sys-apps/man               : 0:18 ms
Unmerging sys-devel/autotools        : 0:19 ms
Unmerging sys-devel/automake         : 0:20 ms
Unmerging sys-devel/binutils         : 0:22 ms
Unmerging sys-devel/bison            : 0:20 ms
Unmerging sys-devel/gcc              : 0:19 ms
Unmerging sys-devel/gcc-config       : 0:08 ms
Unmerging sys-devel/gettext          : 0:20 ms
Unmerging sys-devel/libperl          : 0:07 ms
Unmerging dev-lang/python            : 0:24 ms
Complete unmerge took                : 4:15 ms
Removal of remaining directories     : 0:19 ms
Complete stripping took              : 4:34 ms
joe [/soekris]: []
```

Exit from chroot

```
tabak soekris # exit
```

Finally, our Soekris file system is ready now.

4. Copying The System to The Target

Step 1: Server Setup

Now that our base system is ready it is time to configure your server to boot the Soekris over your local network. (NOTE: This is all done outside your chroot environment!!)

Firstly you need to adjust your dhcp settings.

I use dnsmasq as I find it easier for a basic network setup all I had to do was add to the bottom of my /etc/dnsmasq.conf file.

```
dhcp-boot=/pxelinux.0,SERVER_NAME,SERVER_IP
```

Next we emerge a tftp server..

```
tabak ~ # emerge tftp.hpa
```

Once it is installed I adjusted my /etc/conf.d/in.tftpd to read:

```
# /etc/init.d/in.tftpd
# Path to server files from
INTFTPD_PATH="/tftpboot"
INTFTPD_USER="nobody"
# For more options, see tftpd
INTFTPD_OPTS="-u ${INTFTPD_USER} -l -vvvvv -p -c -s ${INTFTPD_PATH}"
```

Next create a directory called tftpboot:

```
tabak ~ # mkdir /tftpboot
```

Then start tftp

```
tabak ~ # /etc/init.d/in.tftpd start
```

Now there are plenty of different ways to get your soekris board to boot off the network but to make life easy I used the LEAF distribution from <http://leaf.sourceforge.net>. On their website find the newest version of pxelinux.tgz file.

Then unpack it into /tftpboot:

```
tabak ~ # tar xvfz pxelinux.tgz -C /tftpboot
```

Next copy the net4501 file to default

```
tabak ~ # cp /tftpboot/pxelinux.cfg/net4501 /tftpboot/pxelinux.cfg/default
```

Next we have to configure the default file

```
tabak pexlinux.cfg # emacs /tftpboot/pxelinux.cfg/default
```

All you should need to change is the IP address to the IP address of your server. You may have different baud rate settings so adjust them as needed.

My default file shows below

```
serial 0 19200
console 0
timeout 0
ipappend 1
append console=ttyS0,19200 reboot=bios
default pxe/linux rw initrd=pxe/initrd.lrp init=/linuxrc root=/dev/ram0 boot=/tftp/192.168.0.1
LRP=etc.inittabs,hdsupp,wget,smbmount
```

Next we need to be able to see the console output of the soekris, you can use minicom.

Step 2: First steps with the Soekris

You need hook up a null-modem cable to the serial console and to the serial port of your PC. Before plugging in the power cord into the Soekris box, start a terminal application on the PC (I use [minicom](#) for that purpose). Configure it so that its baudrate is at 19200 Baud, 8 Databits, no parity and 1 Stopbit (19200, 8, N, 1).

Step 3: Copying the System To the Target

When its booting hit Ctl-P and it will drop you into a command line, and type

```
> boot f0
```

This will tell the soekris to boot off the network adaptor

It will then try and load pxelinux, if it says file not found then there is something wrong with your setup.

Once LEAF has booted and you had a login, just type 'root' to login.

Run fdisk to make a linux partition

```
pxe# fdisk /dev/hda
```

Make whole /dev/hda1 linux partition

```
Command (m for help): p
Disk /dev/hda: 128 MB, 128057344 bytes
8 heads, 32 sectors/track, 977 cylinders
Units = cylinders of 256 * 512 = 131072 bytes
   Device Boot   Start     End  Blocks  Id System
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
P
Partition number (1-4): 1
First cylinder (1-977, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-977, default 977):
Using default value 977

Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
pxe#
```

We now need to prepare the harddrive so type:

```
> mkfs.ext2 /dev/hda1
```

This will then create the filesystem on your embedded flash disk.

The mount it with:

```
> mkdir /mnt/hdd
> mount /dev/hda1 /mnt/hdd
```

Step 4: Copy filesystem image to Soekris

For soekris (destination), you need to change '/target/dir'. If you are in target directory, then '.' would be enough.

```
pxe# nc -l -p 2342 | tar -C /target/dir -xz -
```

For server (source), you need to replace target_box by its ip address.

```
tabak Image # tar -cz * | nc target_box 2342
```

Then, you can see that net LED starting blinking. After about 5~10min if net LED doesn't blink at all, data transfer is done.

```
pxe# chroot
pxe# lilo
```

Then, you can see message

```
Added Soentoo *
pxe# exit
exit
pxe# reboot
```

Then reboot the Soekris. You should see how the RAM counts up to the amount you have and then, after the correct boot device was selected see the LILO prompt. You won't have to do anything, it will start booting itself. As soon as you start seeing kernel messages, you've done it!

```
comBIOS ver. 1.24 20040312 Copyright (C) 2000-2004 Soekris Engineering.
net4801
0128 Mbyte Memory CPU Geode 266 Mhz
Pri Mas SanDisk SDCFB-128 LBA 980-8-32 125 Mbyte
PXE-M00: BootManage UEFI, PXE-2.0 (build 082)
Slot Vend Dev ClassRev Cmd Stat CL LT HT Base1 Base2 Int
-----
0:00:0 1078 0001 06000000 0107 0280 00 00 00 00000000 00000000 00
0:06:0 100B 0020 02000000 0107 0290 00 3F 00 0000E101 A0000000 10
0:07:0 100B 0020 02000000 0107 0290 00 3F 00 0000E201 A0001000 10
0:08:0 100B 0020 02000000 0107 0290 00 3F 00 0000E301 A0002000 10
0:18:2 100B 0502 01018001 0005 0280 00 00 00 00000000 00000000 00
0:19:0 0E11 A0F8 0C031008 0117 0280 08 38 00 A0003000 00000000 11
1 Seconds to automatic boot. Press Ctrl-P for entering Monitor.
LILO 22.6 boot:
Loading Soentoo.....
BIOS data check successful
[]
```

When booting off a serial console, the whole thing should somewhat look like

```
* Activating (possible) swap... [ ok ]
* Remounting root filesystem read-only (if necessary)... [ ok ]
* Checking root filesystem.../dev/hda1: clean, 12252/31360 files, 111980/125424 blocks
[ ok ]
* Remounting root filesystem read/write... [ ok ]
* Setting hostname to soekris... [ ok ]
* Calculating module dependencies... [ ok ]
* Using /etc/modules.autoload.d/kernel-2.6 as config:
* Loading module errorld_control... [ ok ]
* Autoloaded 1 module(s)
* Checking all filesystems... [ ok ]
* Mounting local filesystems... [ ok ]
* Mounting USB device filesystem (usbfs)... [ ok ]
* Activating (possibly) more swap... [ ok ]
* Caching service dependencies... * Setting system clock to hardware clock [UTC]... [ ok ]
* Configuring kernel parameters... [ ok ]
* Updating environment... [ ok ]
* Cleaning /var/lock, /var/run... [ ok ]
* Cleaning /tmp directory... [ ok ]
* Setting user font... [ ok ]
* Loading key mappings... [ ok ]
* Bringing lo up... [ ok ]
* Initializing random number generator... [ ok ]
INIT: Entering runlevel: 3
* Bringing etho up (192.17.5.158)... [ ok ]
* Starting mysqld... [ ok ]
* Mounting network filesystems... [ ok ]
* Starting sshd... [ ok ]
* Starting speech2... [ ok ]
* Starting local... [ ok ]
This is soekris.soekrisnet (Linux 1586 2.6.8.1) 06:36:21
soekris login: []
CTRL-A 2 for help | 19200 Bn1 | NDR | Minicom 2.00.0 | VT102 | Online 00:01
```

When you then ssh to your box, this will most likely look like

```
acer_joe [-]: ssh root@172.17.5.158
Password:
Last login: Sat Jan 5 06:38:26 1980 from joe.alex.stw.uni-erlangen.de
soekris [-]: ls
total 3.0K
-rw-r--r-- 1 root root 1.3K 05.01.1980 06:28:40 .bash_history
-rw-r--r-- 1 root root 29 20.10.2004 15:05:10 .inputrc
-rw-r--r-- 1 root root 0 18.07.2004 02:47:19 .keep
soekris [-]: echo 'Hello world, Soentoo!'
Hello world, Soentoo!
soekris [-]: uptime
06:38:56 up 4 min, 1 user, load average: 0.11, 0.30, 0.15
soekris [-]: cat /proc/cpuinfo
processor       : 0
vendor_id     : Geode by NSC
cpu family    : 5
model         : 9
model name    : Unknown
stepping      : 1
cpu MHz       : 266.771
fdtvg_bug     : no
hlt_bug       : no
f00f_bug      : no
coma_bug      : no
fpu           : yes
fpu_exception : yes
cpuid level   : 2
wp            : yes
flags         : fpu tsc mtr cx8 cmov mmx cxmmx
bogomips     : 524.28
soekris [-]: df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hda1       119M  106M   7.0M  94% /
none            62M   0    62M   0% /dev/shm
soekris [-]:
```

Step 5: General information about the setup

The box is by default configured so that **eth0** has the IP you selected during configuration. **sshd** is also automatically started, so you can immediately start ssh-ing to your box. There's by default no user configured, just "root" (who may also log in at the serial console). The default password for the "root" user is "soekris".

You can write some commands which should be started during booting time at the end of `‘/etc/profile’`. Add `‘mount / -o remount,rw’` under `‘/etc/profile’` to prevent that filesystem is mounted by read-only. You can edit text by NANO editor. The size of VI or EMAC is too big to install for Soekris.

Step 6: Installation of New Software

Currently we keep working on how to install new software easily. Because we installed only basic software and removed all parts for installation or upgrading software for the minimized filesystem of Soekris, we cannot directly install or upgrade software on Soekris. However, we probably have to install some other software for various experiments later. For applications, currently we are using NFS (Network File System) for all Soekris to share a compiled application on a server. Using the filesystem inside `‘/Image’` before stripping, we can install or compile a new application by `‘chroot’`. Then, we can move the binary files to one specific directory which is connected to other Soekris by NFS. When we compile and install a new application, if we can modify `‘./configure’` file or `‘Makefile’`, the created binary file can be placed in the specific directory automatically.

5. Web-pages related with OS installation of Soekris

- <http://www.soekris.com/support.htm>
- <http://soentoo.sourceforge.net/>
- <http://forums.gentoo.org/viewtopic-t-511777-highlight-soekris+4826.html>
- <http://www.ultradesic.com/index.php?section=21>
- <http://wiki.antlinux.com/pmwiki.php?n=AntLinux.LinuxOnSoekris#LinuxOnSoekrisLinks>
- http://andersdrenge.dk/?page_id=41